DASSAULT SYSTEMES

SIMULIA

ABAQUS 6.14

THEORY GUIDE



3DEXPERIENCE

Abaqus

Theory Guide

Legal Notices

CAUTION: This documentation is intended for qualified users who will exercise sound engineering judgment and expertise in the use of the Abaqus Software. The Abaqus Software is inherently complex, and the examples and procedures in this documentation are not intended to be exhaustive or to apply to any particular situation. Users are cautioned to satisfy themselves as to the accuracy and results of their analyses.

Dassault Systèmes and its subsidiaries, including Dassault Systèmes Simulia Corp., shall not be responsible for the accuracy or usefulness of any analysis performed using the Abaqus Software or the procedures, examples, or explanations in this documentation. Dassault Systèmes and its subsidiaries shall not be responsible for the consequences of any errors or omissions that may appear in this documentation.

The Abaqus Software is available only under license from Dassault Systèmes or its subsidiary and may be used or reproduced only in accordance with the terms of such license. This documentation is subject to the terms and conditions of either the software license agreement signed by the parties, or, absent such an agreement, the then current software license agreement to which the documentation relates.

This documentation and the software described in this documentation are subject to change without prior notice.

No part of this documentation may be reproduced or distributed in any form without prior written permission of Dassault Systèmes or its subsidiary.

The Abaqus Software is a product of Dassault Systèmes Simulia Corp., Providence, RI, USA.

© Dassault Systèmes, 2014

Abaqus, the 3DS logo, SIMULIA, CATIA, and Unified FEA are trademarks or registered trademarks of Dassault Systèmes or its subsidiaries in the United States and/or other countries.

Other company, product, and service names may be trademarks or service marks of their respective owners. For additional information concerning trademarks, copyrights, and licenses, see the Legal Notices in the Abaqus 6.14 Installation and Licensing Guide.

Preface

This section lists various resources that are available for help with using Abaqus Unified FEA software.

Support

Both technical software support (for problems with creating a model or performing an analysis) and systems support (for installation, licensing, and hardware-related problems) for Abaqus are offered through a global network of support offices, as well as through our online support system. Regional contact information is accessible from **SIMULIA→Locations** at www.3ds.com/simulia. The online support system is accessible by selecting the **SUBMIT A REQUEST** link at www.3ds.com/support-training (http://www.3ds.com/support-training/).

Online support

Dassault Systèmes provides a knowledge base of questions and answers, solutions to questions that we have answered, and guidelines on how to use Abaqus, Engineering Process Composer, Isight, Tosca, fe-safe, and other SIMULIA products. The knowledge base is available from www.3ds.com/support-training (http://www.3ds.com/support-training/).

By using the online support system, you can also submit new requests for support. All support/service requests are tracked. If you contact us by means outside the system to discuss an existing support problem and you know the support request number, please mention it so that we can query the knowledge base to see what the latest action has been.

Training

All support offices offer regularly scheduled public training classes. The courses are offered in a traditional classroom form and via the Web. We also provide training seminars at customer sites. All training classes and seminars include workshops to provide as much practical experience with Abaqus as possible. For a schedule and descriptions of available classes, see the **Training Classes** link at www.3ds.com/simulia or call your support office.

Feedback

We welcome any suggestions for improvements to Abaqus software, the support program, or documentation. We will ensure that any enhancement requests you make are considered for future releases. If you wish to make a suggestion about the service or products, refer to www.3ds.com/simulia. Complaints should be made by contacting your support office or by visiting **SIMULIA→Quality Assurance** at www.3ds.com/simulia.

Contents

1. Introduction and Basic Equations

Introduction

Introduction: general 1.1.1

Notation

Notation 1.2.1

Finite rotations

Rotation variables 1.3.1

Deformation, strain, and strain rates

Deformation 1.4.1
Strain measures 1.4.2
Rate of deformation and strain increment 1.4.3
The additive strain rate decomposition 1.4.4

Equilibrium, stress, and state storage

Equilibrium and virtual work 1.5.1
Stress measures 1.5.2
Stress rates 1.5.3
State storage 1.5.4
Energy balance 1.5.5

2. Procedures

Overview

Procedures: overview and basic equations 2.1.1

Nonlinear solution methods

Nonlinear solution methods in Abaqus/Standard 2.2.1
Quasi-Newton solution technique 2.2.2
Direct cyclic algorithm 2.2.3

Buckling and postbuckling

Eigenvalue buckling prediction 2.3.1
Modified Riks algorithm 2.3.2

Nonlinear dynamics

Implicit dynamic analysis 2.4.1
Intermittent contact/impact 2.4.2
Subspace dynamics 2.4.3

Equivalent rigid body dynamic motion 2.4.4
Explicit dynamic analysis 2.4.5

Modal dynamics

Eigenvalue extraction 2.5.1
Variables associated with the natural modes of a model 2.5.2
Linear dynamic analysis using modal superposition 2.5.3
Damping options for modal dynamics 2.5.4
Modal dynamic analysis 2.5.5
Response spectrum analysis 2.5.6
Steady-state linear dynamic analysis 2.5.7
Random response analysis 2.5.8
Base motions in modal-based procedures 2.5.9

Complex harmonic oscillations

Direct steady-state dynamic analysis 2.6.1
Subspace-based steady-state dynamic analysis 2.6.2

Steady-state transport analysis

Steady-state transport analysis 2.7.1

Analysis of porous media

Effective stress principle for porous media 2.8.1
Discretized equilibrium statement for a porous medium 2.8.2
Constitutive behavior in a porous medium 2.8.3
Continuity statement for the wetting liquid phase in a porous medium 2.8.4
Solution strategy for coupled diffusion/deformation 2.8.5

Coupled fluid-solid analysis

Coupled acoustic-structural medium analysis 2.9.1

Piezoelectric analysis

Piezoelectric analysis 2.10.1

Heat transfer

Uncoupled heat transfer analysis 2.11.1
Shell heat conduction 2.11.2
Convection/diffusion 2.11.3
Cavity radiation 2.11.4
View factor calculation 2.11.5

Coupled thermal-electrical analysis

Coupled thermal-electrical analysis 2.12.1

Mass diffusion

Mass diffusion analysis 2.13.1

Substructuring

Substructuring and substructure analysis 2.14.1

Submodeling

Submodeling analysis 2.15.1

Fracture mechanics

J-integral evaluation 2.16.1
Stress intensity factor extraction 2.16.2
T-stress extraction 2.16.3
Prediction of the direction of crack propagation 2.16.4

Stress linearization

Stress linearization 2.17.1

Design sensitivity analysis

Design sensitivity analysis 2.18.1

3. Elements

Overview

Element library: overview 3.1.1

Continuum elements

Solid element overview 3.2.1
Solid element formulation 3.2.2
Hybrid incompressible solid element formulation 3.2.3
Solid isoparametric quadrilaterals and hexahedra 3.2.4
Continuum elements with incompatible modes 3.2.5
Triangular, tetrahedral, and wedge elements 3.2.6
Generalized plane strain elements 3.2.7
Axisymmetric elements 3.2.8
Axisymmetric elements allowing nonlinear bending 3.2.9

Infinite elements

Solid infinite elements 3.3.1
Acoustic infinite elements 3.3.2

Membrane and truss elements

Membrane elements 3.4.1
Truss elements 3.4.2
Axisymmetric membranes 3.4.3

Beam elements

Beam element overview 3.5.1

Beam element formulation 3.5.2
Euler-Bernoulli beam elements 3.5.3
Hybrid beam elements 3.5.4
Mass and inertia for Timoshenko beams 3.5.5
Meshed beam cross-sections 3.5.6

Shell elements

Shell element overview 3.6.1
Axisymmetric shell elements 3.6.2
Shear flexible small-strain shell elements 3.6.3
Triangular facet shell elements 3.6.4
Finite-strain shell element formulation 3.6.5
Small-strain shell elements in Abaqus/Explicit 3.6.6
Axisymmetric shell element allowing asymmetric loading 3.6.7
Transverse shear stiffness in composite shells and offsets from the midsurface 3.6.8
Rotary inertia for 5 degree of freedom shell elements 3.6.9

Rebar

Rebar modeling in two dimensions 3.7.1
Rebar modeling in three dimensions 3.7.2
Rebar modeling in shell, membrane, and surface elements 3.7.3

Hydrostatic fluid calculations

Hydrostatic fluid calculations 3.8.1

Special-purpose elements

Elbow elements 3.9.1
Frame elements with lumped plasticity 3.9.2
Buckling strut response for frame elements 3.9.3
Tube support elements 3.9.4
Line spring elements 3.9.5
Flexible joint element 3.9.6
Rotary inertia element 3.9.7
Distributing coupling elements 3.9.8

4. Mechanical Constitutive Theories

Overview

Mechanical constitutive models 4.1.1

Plasticity overview

Plasticity models: general discussion 4.2.1
Integration of plasticity models 4.2.2

Metal plasticity

Metal plasticity models 4.3.1
Isotropic elasto-plasticity 4.3.2
Stress potentials for anisotropic metal plasticity 4.3.3
Rate-dependent metal plasticity (creep) 4.3.4
Models for metals subjected to cyclic loading 4.3.5
Porous metal plasticity 4.3.6
Cast iron plasticity 4.3.7
ORNL constitutive theory 4.3.8
Deformation plasticity 4.3.9
Heat generation caused by plastic straining 4.3.10

Plasticity for non-metals

Porous elasticity 4.4.1
Models for granular or polymer behavior 4.4.2
Critical state models 4.4.3
Drucker-Prager/Cap model for geological materials 4.4.4
Mohr-Coulomb model 4.4.5
Models for crushable foams 4.4.6

Other inelastic models

An inelastic constitutive model for concrete 4.5.1
Damaged plasticity model for concrete and other quasi-brittle materials 4.5.2
A cracking model for concrete and other brittle materials 4.5.3
Constitutive model for jointed materials 4.5.4

Large-strain elasticity

Hyperelastic material behavior 4.6.1
Fitting of hyperelastic and hyperfoam constants 4.6.2
Anisotropic hyperelastic material behavior 4.6.3

Mullins effect and permanent set

Mullins effect 4.7.1
Permanent set 4.7.2

Viscoelasticity

Viscoelasticity 4.8.1
Finite-strain viscoelasticity 4.8.2
Frequency domain viscoelasticity 4.8.3

Hysteresis

Hysteresis 4.9.1

5. Interface Modeling

Contact modeling

Small-sliding interaction between bodies 5.1.1
Finite-sliding interaction between deformable bodies 5.1.2
Finite-sliding interaction between a deformable and a rigid body 5.1.3

Surface interactions

Contact pressure definition 5.2.1
Pressure and fluid flow in pore pressure contact 5.2.2
Coulomb friction 5.2.3
Thermal interface definition 5.2.4
Heat generation caused by frictional sliding 5.2.5
Heat generation caused by electrical current 5.2.6
Surface-based acoustic-structural medium interaction 5.2.7

6. Loading and Constraints

Dynamic loading

Centrifugal, Coriolis, and rotary acceleration forces 6.1.1
Baseline correction of accelerograms 6.1.2

Abaqus/Aqua loading

Drag, inertia, and buoyancy loading 6.2.1
Airy wave theory 6.2.2
Stokes wave theory 6.2.3

Incident wave loading

Loading due to an incident dilatational wave field 6.3.1

Pressure penetration loading

Pressure penetration loading with surface-based contact 6.4.1

Load stiffness

Pressure load stiffness 6.5.1
Load stiffness for beam elements 6.5.2
Pressure loadings on elbow elements 6.5.3

Multi-point constraints

Sliding constraint 6.6.1
Shell to solid constraint 6.6.2
Revolute joint 6.6.3
Universal joint 6.6.4

Local velocity constraint 6.6.5
Kinematic coupling 6.6.6

7. References

References 7.1.1

1. Introduction and Basic Equations

Introduction 1.1
Notation 1.2
Finite rotations 1.3
Deformation, strain, and strain rates 1.4
Equilibrium, stress, and state storage 1.5

1.1 Introduction

- “Introduction: general,” Section 1.1.1

1.1.1 INTRODUCTION: GENERAL

The Abaqus finite element system includes:

- Abaqus/Standard, a general-purpose finite element program;
- Abaqus/Explicit, an explicit dynamics finite element program;
- Abaqus/CAE, an interactive environment used to create finite element models, submit Abaqus analyses, monitor and diagnose jobs, and evaluate results; and
- Abaqus/Viewer, a subset of Abaqus/CAE that contains only the postprocessing capabilities of the Visualization module.

Several add-on options that are available only if your license includes them can be used to further extend the capabilities of Abaqus/Standard and Abaqus/Explicit.

- The Abaqus/Aqua option works with Abaqus/Standard and Abaqus/Explicit. Abaqus/Aqua contains optional features that are specifically designed for the analysis of beam-like structures installed underwater and subject to loading by water currents and wave action.
- The Abaqus/Design option enables you to perform design sensitivity analysis (DSA). The Abaqus/Design option works with Abaqus/Standard.
- Abaqus/AMS is an optional eigensolver that works within Abaqus/Standard providing very fast solution of large symmetric eigenvalue problems.
- The Abaqus co-simulation technique provides two applications, available as separate add-on capabilities, for coupling between Abaqus and third-party analysis programs.
- Abaqus/Foundation is an optional subset of Abaqus/Standard that provides more cost-efficient access to the linear static and dynamic analysis functionality in Abaqus/Standard.

This guide describes the theories used in Abaqus. Many sections in this guide apply to both Abaqus/Standard and Abaqus/Explicit. Certain sections obviously apply only to either Abaqus/Standard or Abaqus/Explicit; for example, all sections in the chapter on procedures apply to Abaqus/Standard, except the section discussing the explicit dynamic integration procedure, which applies to Abaqus/Explicit. If it is not obvious to which program a section applies, it is clearly indicated.

The objective of this guide is to define the theories used in Abaqus that are generally not available in the standard textbooks on mechanics, structures, and finite elements but are well known to the engineer who uses Abaqus. The guide is intended as a reference document that defines what is available in the code. Nevertheless, it is written in such a way that it can also be used as a tutorial document by a reader who needs to obtain some background in an unfamiliar area. The material is presented in a way that should make it accessible to any user with an engineering background. Some of the theories may be relatively unfamiliar to such a user; for example, few engineering curricula provide extensive background in plasticity, shell theory, finite deformations of solids, or the analysis of porous media. Yet Abaqus contains capabilities for all of these models and many others. The guide is far from comprehensive in its coverage of such topics: in this sense it is only a reference volume. The user is strongly encouraged to pursue topics of interest through texts and papers. Chapter 7, “References,” at the end of this guide lists references that should provide a starting point

for obtaining such information. (Abaqus does not supply copies of papers that have appeared in publications other than those of Abaqus. EPRI reports can be obtained from Research Reports Center (RRC), Box 50490, Palo Alto, CA 94303.)

Chapter 1, “Introduction and Basic Equations,” discusses the notation used in the guide, some basic concepts of kinematics and mechanics—such as rotations, stress, and equilibrium—as well as the basic equations of nonlinear finite element analysis. Chapter 2, “Procedures,” describes the various analysis procedures (nonlinear static stress analysis, dynamics, eigenvalue extraction, etc.) that are available in Abaqus. Chapter 3, “Elements,” describes the element formulations. Chapter 4, “Mechanical Constitutive Theories,” describes the mechanical constitutive theories.

Chapter 5, “Interface Modeling,” discusses the most important aspects of the contact/interaction formulation in Abaqus/Standard. Chapter 6, “Loading and Constraints,” describes the formulation of some of the more complicated load types and multi-point constraints.

1.2 Notation

- “Notation,” Section 1.2.1

1.2.1 NOTATION

Products: Abaqus/Standard Abaqus/Explicit

Notation is often a serious obstacle that prevents an engineer from using advanced textbooks; for example, general curvilinear tensor analysis and functional analysis are both necessary in some of the theories used in Abaqus, but the unfamiliar notations commonly used in these areas often discourage the user from pursuing their study. The notation used in most of this guide (direct matrix notation) may be unfamiliar to some readers; but it is not difficult or time consuming to gain enough familiarity with the notation for it to be useful, and it is definitely worthwhile. This notation is commonly used in the modern engineering literature—it is a shorthand version of the familiar matrix notation used in many older engineering textbooks. The notation is appealing—once it is understood—because it allows the equations to be developed concisely, and the physical ideas can be perceived without the distraction of the complexities that arise from the choice of the particular basis system that will eventually be used to express the same concepts in component form. Because the notation has become so standard in the literature, the user who wishes or needs to read textbooks and papers that are related to the use of Abaqus will find that familiarity with this notation is desirable.

Both direct matrix notation and component form notation are used in the guide. Both notations are described in this section. Direct matrix notation is used whenever possible. However, vectors, matrices, and the higher-order tensors used in the theories must eventually be written in component form to store them as a set of numbers on the computer. Thus, both ways of writing these quantities will be needed in the guide.

Basic quantities

The quantities needed to formulate the theory are scalars, vectors, second-order tensors (matrices), and—occasionally—fourth-order tensors (for example, the stress-strain transformation for linear elasticity). In direct matrix notation these are written as:

a scalar value	a
a vector	$\mathbf{a}$ or $\lfloor a \rfloor$
with the transpose	$\mathbf{a}^T$ or $\{a\}$
a second-order tensor or matrix	$\mathbf{a}$ or $[a]$
with the transpose	$\mathbf{a}^T$ or $[a]^T$
and	
a fourth-order tensor	$\mathbf{A}$

Vectors and second-order tensors (matrices) are written in the same way: they are distinguished by the context. In direct matrix notation there is generally no need to indicate that a vector must be transposed. The context determines whether a vector is to be used as a "column" vector $\mathbf{a}$ or as a "row" vector $\mathbf{a}^T$. In this case the transpose superscript is only used to improve the readability of an expression.

On the other hand, for second-order nonsymmetric tensors the addition of a transpose superscript will change the meaning of an expression.

This representation of vectors and tensors is very general and convenient for developing the theory so that the equations can be understood easily in terms of their physical meaning. However, in actual computations we have to work with individual numbers, so vectors and tensors must be expressed in terms of their components. These components are associated with an axis system that defines a set of base vectors at each point in space. The simplest axis system is rectangular Cartesian, because the base vectors are orthogonal unit vectors in the same direction at all points. Unfortunately, we need more generality than this because we will be dealing with shells and beams, where stress, strain, etc. are most conveniently described in terms of directions on the surface of the shell (or associated with the axis of the beam), and these usually change as we move around on the surface. To retain this necessary generality and express vectors and matrices in component form, we introduce a general set of base vectors, $\mathbf{e}_\alpha$, $\alpha = 1, 2, 3$, which are not necessarily orthogonal or of unit length but are sufficient to define the components of a vector (for this purpose they must not be parallel or have zero length). A vector $\mathbf{a}$ can then be written

$$\mathbf{a} = a^1\mathbf{e}_1 + a^2\mathbf{e}_2 + a^3\mathbf{e}_3,$$

where the numbers a^1, a^2, and a^3 are the components of $\mathbf{a}$ associated with $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$.

In actual cases the $\mathbf{e}_\alpha$ are chosen for convenience (for example, see "Conventions," Section 1.2.2 of the Abaqus Analysis User's Guide, for a description of how base vectors are chosen for surface elements in Abaqus), and then the a^α are obtained.

To save writing, we adopt the usual summation convention that a repeated index is summed—in this case over the range 1 to 3—so that the above equation is written

$$\mathbf{a} = a^\alpha\mathbf{e}_\alpha.$$

Likewise, the component form of a matrix will be

$$\mathbf{a} = \mathbf{e}_\alpha a^{\alpha\beta}\mathbf{e}_\beta = a^{\alpha\beta}\mathbf{e}_\alpha\mathbf{e}_\beta,$$

or, written out,

$$\begin{aligned}\mathbf{a} =& \mathbf{e}_1 a^{11}\mathbf{e}_1 + \mathbf{e}_1 a^{12}\mathbf{e}_2 + \mathbf{e}_1 a^{13}\mathbf{e}_3 \\ & \mathbf{e}_2 a^{21}\mathbf{e}_1 + \mathbf{e}_2 a^{22}\mathbf{e}_2 + \mathbf{e}_2 a^{23}\mathbf{e}_3 \\ & \mathbf{e}_3 a^{31}\mathbf{e}_1 + \mathbf{e}_3 a^{32}\mathbf{e}_2 + \mathbf{e}_3 a^{33}\mathbf{e}_3.\end{aligned}$$

Similarly, a fourth-order tensor can be written in component form as

$$\mathbf{A} = A^{\alpha\beta\gamma\delta}\mathbf{e}_\alpha\mathbf{e}_\beta\mathbf{e}_\gamma\mathbf{e}_\delta.$$

While we will need such completely general base vectors for describing the stresses and strains on shells and beams, in many cases it is convenient to use rectangular Cartesian components so that the $\mathbf{e}_\alpha$ are orthogonal unit vectors. To distinguish this particular case, we will use Latin indices instead of Greek

indices. Thus, $\mathbf{e}_\alpha$ are a set of general base vectors; while $\mathbf{e}_i$ are rectangular Cartesian base vectors; and a^α is the component of the vector $\mathbf{a}$ along a general base vector, while a^i, $i = 1, 2, 3$, is the component of $\mathbf{a}$ along the ith Cartesian direction.

Vector and tensor concepts and their representation are discussed in many textbooks—see Flugge (1972), for example.

Basic operations

The usual matrix and vector operators are indicated in this guide as follows:

Dot product of two vectors:

$$a = \mathbf{b} \cdot \mathbf{c}$$

(The dot symbol defines this operation completely, regardless of whether $\mathbf{b}$ or $\mathbf{c}$ is transposed—i.e., $\mathbf{b} \cdot \mathbf{c} = \mathbf{b}^T \cdot \mathbf{c}$.)

Cross product of two vectors:

$$\mathbf{a} = \mathbf{b} \times \mathbf{c}$$

Matrix multiplication:

$$\mathbf{a} = \mathbf{b} \cdot \mathbf{c}$$

(It is implicitly assumed that $\mathbf{b}$ and $\mathbf{c}$ are dimensioned correctly, as needed for the operation to make sense; in addition, if $\mathbf{b}$ is a nonsymmetric tensor, $\mathbf{b}^T \cdot \mathbf{c} \neq \mathbf{b} \cdot \mathbf{c}$.)

Scalar product of two matrices:

$$a = \mathbf{b} : \mathbf{c}$$

This operation means that corresponding conjugate components of the two matrices are multiplied as pairs and the products summed. Thus, for instance, if $\mathbf{b}$ is the stress matrix, $\boldsymbol{\sigma}$, and $\mathbf{c}$ the conjugate rate of strain matrix, $d\boldsymbol{\varepsilon}$, then $\boldsymbol{\sigma} : d\boldsymbol{\varepsilon}$ would give the rate of internal work per volume, dW^I.

It is also necessary to define the dyadic product of two vectors:

$$\mathbf{a} = \mathbf{b}\mathbf{c} \qquad \text{or} \qquad \mathbf{a} = \mathbf{b}\mathbf{c}^T$$

This operation creates a second-order tensor (or dyad) out of two vectors. In component notation this notation is equivalent to $a^{ij} = b^i c^j$.

A matrix of derivatives,

$$\frac{\partial \mathbf{a}}{\partial \mathbf{b}},$$

means

$$d\mathbf{a} = \left(\frac{\partial \mathbf{a}}{\partial \mathbf{b}}\right) \cdot d\mathbf{b}.$$

Throughout this guide it will be assumed implicitly that, when a derivative is taken with respect to time, we mean the *material* time derivative; that is, the change in a variable with respect to time whilst looking at a particular material particle. When this is not the case for a particular equation, it will be stated explicitly when the equation appears.

Provided that we are careful about interpreting $(\partial \mathbf{a}/\partial \mathbf{b})$ in the manner illustrated above, standard concepts of elementary calculus clearly hold; for example, if $\mathbf{a}$ is a vector-valued function of the vector-valued function $\mathbf{b}$, which in turn is a vector-valued function of $\mathbf{c}$, that is $\mathbf{a} = \mathbf{a}(\mathbf{b}(\mathbf{c}))$, then

$$d\mathbf{a} = \frac{\partial \mathbf{a}}{\partial \mathbf{b}} \cdot \frac{\partial \mathbf{b}}{\partial \mathbf{c}} \cdot d\mathbf{c},$$

or, if $\mathbf{a}(\mathbf{b}, \mathbf{c})$:

$$d\mathbf{a} = \frac{\partial \mathbf{a}}{\partial \mathbf{b}} \cdot d\mathbf{b} + \frac{\partial \mathbf{a}}{\partial \mathbf{c}} \cdot d\mathbf{c}.$$

Due to these properties many useful results can be obtained quickly and expressed in a compact, easily understood, form.

Components of a vector or a matrix in a coordinate system

In the previous section we introduced the idea that a vector $\mathbf{a}$ or a matrix $\mathbf{a}$ can be written in terms of components associated with some conveniently chosen set of base vectors, $\mathbf{e}_\alpha$. We now show how the components a^α (or $a^{\alpha\beta}$) are obtained. We can do so using the dot product. For each of the three base vectors, $\mathbf{e}_\alpha$, we define a conjugate base vector $\mathbf{e}^\alpha$, as follows. Choose $\mathbf{e}^1$ as normal to $\mathbf{e}_2$ and $\mathbf{e}_3$, such that the dot product $\mathbf{e}^1 \cdot \mathbf{e}_1 = 1$. Similarly, choose $\mathbf{e}^2$ normal to $\mathbf{e}_3$ and $\mathbf{e}_1$, such that $\mathbf{e}^2 \cdot \mathbf{e}_2 = 1$; and $\mathbf{e}^3$ normal to $\mathbf{e}_1$ and $\mathbf{e}_2$, such that $\mathbf{e}^3 \cdot \mathbf{e}_3 = 1$. Thus,

$$\mathbf{e}^1 \cdot \mathbf{e}_1 = 1\ ,\ \mathbf{e}^1 \cdot \mathbf{e}_2 = 0\ ,\ \mathbf{e}^1 \cdot \mathbf{e}_3 = 0$$

$$\mathbf{e}^2 \cdot \mathbf{e}_1 = 0\ ,\ \mathbf{e}^2 \cdot \mathbf{e}_2 = 1\ ,\ \mathbf{e}^2 \cdot \mathbf{e}_3 = 0$$

$$\mathbf{e}^3 \cdot \mathbf{e}_1 = 0\ ,\ \mathbf{e}^3 \cdot \mathbf{e}_2 = 0\ ,\ \mathbf{e}^3 \cdot \mathbf{e}_3 = 1$$

We can write this compactly as

$$\mathbf{e}^\alpha \cdot \mathbf{e}_\beta = \delta^\alpha_\beta,$$

where $\delta^\alpha_\beta = 1$ if $\alpha = \beta$, and $\delta^\alpha_\beta = 0$, otherwise. (δ^α_β is called the "Kronecker delta.") In matrix notation δ^α_β is the unit matrix $\mathbf{I}$: we can also write the above equation defining $\mathbf{e}^1$, $\mathbf{e}^2$, and $\mathbf{e}^3$ in matrix form as

$$\left\{ \begin{array}{l} \lfloor \mathbf{e}^1 \rfloor \\ \lfloor \mathbf{e}^2 \rfloor \\ \lfloor \mathbf{e}^3 \rfloor \end{array} \right\} . \lfloor \{\mathbf{e}_1\}\ \{\mathbf{e}_2\}\ \{\mathbf{e}_3\} \rfloor = \mathbf{I},$$

so that, if one set of base vectors—$\mathbf{e}_i$, say—is known, the others are easily obtained.

With this additional set of base vectors, we can immediately obtain the components of a vector or a matrix as follows.

Consider a vector $\mathbf{a}$. Then $\mathbf{a} \cdot \mathbf{e}^\alpha = a^\beta \mathbf{e}_\beta \cdot \mathbf{e}^\alpha$ (writing $\mathbf{a}$ in component form, using the basis vectors $\mathbf{e}_\beta$), and since $\mathbf{e}_\beta \cdot \mathbf{e}^\alpha = \delta^\alpha_\beta = 1$, only if $\alpha = \beta$,

$$\begin{aligned} \mathbf{a} \cdot \mathbf{e}^\alpha &= a^\beta \mathbf{e}_\beta \cdot \mathbf{e}^\alpha \\ &= a^\beta \delta^\alpha_\beta = a^\alpha. \end{aligned}$$

In exactly the same way we could have written

$$a_\alpha = \mathbf{a} \cdot \mathbf{e}_\alpha$$

by expressing $\mathbf{a}$ as components associated with the $\mathbf{e}^\alpha$ base vectors, $\mathbf{a} = a_\alpha \mathbf{e}^\alpha$.

Similarly, for a matrix,

$$a^{\alpha\beta} = (\mathbf{e}^\alpha)^T \cdot \mathbf{a} \cdot \mathbf{e}^\beta = \mathbf{e}^\alpha \cdot \mathbf{a} \cdot \mathbf{e}^\beta,$$

and

$$a_{\alpha\beta} = (\mathbf{e}_\alpha)^T \cdot \mathbf{a} \cdot \mathbf{e}_\beta = \mathbf{e}_\alpha \cdot \mathbf{a} \cdot \mathbf{e}_\beta.$$

These component definitions are particularly convenient for calculating the dot product of two vectors, for we can write

$$\begin{aligned} \mathbf{a} \cdot \mathbf{b} &= (a_\alpha \mathbf{e}^\alpha) \cdot (b^\beta \mathbf{e}_\beta) = a_\alpha b^\beta \mathbf{e}^\alpha \cdot \mathbf{e}_\beta, \\ &= a_\alpha b^\beta \delta^\alpha_\beta, \\ &= a_\alpha b^\alpha, \end{aligned}$$

which is

$$\mathbf{a} \cdot \mathbf{b} = a_1 b^1 + a_2 b^2 + a_3 b^3.$$

Similarly, the scalar product of two matrices is

$$\mathbf{a} : \mathbf{b} = a_{\alpha\beta} b^{\beta\alpha};$$

that is, we simply multiply corresponding entries in the $a_{\alpha\beta}$ and $b^{\beta\alpha}$ arrays, arranged as matrices, and then sum the products.

Finally, on the computer we need to store only one form of component: a_α, $a_{\alpha\beta}$ or a^α, $a^{\alpha\beta}$. We can always go from one to the other using the "metric tensor," $g_{\alpha\beta}$, and its inverse, $g^{\alpha\beta}$, which are defined as

$$g_{\alpha\beta} = \mathbf{e}_\alpha \cdot \mathbf{e}_\beta,$$

and

$$g^{\alpha\beta} = \mathbf{e}^\alpha \cdot \mathbf{e}^\beta.$$

For

$$\begin{aligned} a^\alpha &= \mathbf{a} \cdot \mathbf{e}^\alpha \quad \text{(from above)}, \\ &= a_\beta \mathbf{e}^\beta \cdot \mathbf{e}^\alpha \quad \text{(expressing } \mathbf{a} \text{ in component form)}, \\ &= a_\beta g^{\beta\alpha} \quad \text{(by the definition of } g^{\beta\alpha}\text{)}. \end{aligned}$$

Thus, $a^\alpha = g^{\beta\alpha} a_\beta$; similarly $a_\alpha = g_{\beta\alpha} a^\beta$, and, by extension, for matrices,

$$a^{\alpha\beta} = g^{\gamma\alpha} g^{\delta\beta} a_{\gamma\delta}$$

and

$$a_{\alpha\beta} = g_{\gamma\alpha} g_{\delta\beta} a^{\gamma\delta}.$$

The metric tensor and its inverse are symmetric:

$$g_{\alpha\beta} = \mathbf{e}_\alpha \cdot \mathbf{e}_\beta = \mathbf{e}_\beta \cdot \mathbf{e}_\alpha = g_{\beta\alpha}.$$

The two sets of base vectors and components of vectors or matrices associated with them are named as follows:

$\mathbf{e}_\alpha$	are covariant base vectors,
$\mathbf{e}^\alpha$	are contravariant base vectors,
a_α (or $a_{\alpha\beta}$)	are covariant components of a vector (or matrix),
a^α (or $a^{\alpha\beta}$)	are contravariant components of a vector (or matrix).

Thus, the contravariant components are those associated with the covariant base vectors, $\mathbf{a} = a^\alpha \mathbf{e}_\alpha$, and vice versa. The simplest case is when the basis is a set of orthogonal unit vectors (a rectangular Cartesian system) because then—from the definition $\mathbf{e}^\alpha \cdot \mathbf{e}_\beta = \delta^\alpha_\beta$—we see that $\mathbf{e}^\alpha = \mathbf{e}_\alpha$, and so $a_\alpha = a^\alpha$ and we need not distinguish the type of component. Whenever possible a rectangular Cartesian system is chosen, so the type of component need not be distinguished. This system is discussed in more detail in the sections on beam elements and shell elements.

Components of a derivative

Consider a vector-valued function, $\mathbf{b}$, which is expressed in component form on a basis system, $\mathbf{e}_\alpha$. Let the vector-valued function $\mathbf{a}$ depend on $\mathbf{b}$: $\mathbf{a}(\mathbf{b})$. Then

$$d\mathbf{a} = \left(\frac{\partial \mathbf{a}}{\partial \mathbf{b}}\right) \cdot \mathbf{e}_\alpha db^\alpha$$

so that the component of $d\mathbf{a}$ associated with a change db^α is

$$\left(\frac{\partial \mathbf{a}}{\partial \mathbf{b}}\right) \cdot \mathbf{e}_\alpha,$$

which we write, for convenience, as

$$\left(\frac{\partial \mathbf{a}}{\partial b^\alpha}\right),$$

meaning

$$d\mathbf{a} = \left(\frac{\partial \mathbf{a}}{\partial b^\alpha}\right) db^\alpha.$$

Now suppose $\mathbf{a}$ is written on a different basis—$\mathbf{E}_\alpha$, say—so that we store $\mathbf{a}$ as the components

$$da_\alpha = d\mathbf{a} \cdot \mathbf{E}_\alpha.$$

Then

$$da_\alpha = \mathbf{E}_\alpha \cdot \frac{\partial \mathbf{a}}{\partial b^\beta} db^\beta.$$

Typically we would then write

$$da_\alpha = H_{\alpha\beta} db^\beta,$$

where

$$H_{\alpha\beta} = \mathbf{E}_\alpha \cdot \frac{\partial \mathbf{a}}{\partial b^\beta} = \mathbf{E}_\alpha \cdot \frac{\partial \mathbf{a}}{\partial \mathbf{b}} \cdot \mathbf{e}_\beta.$$

Readers who are familiar with general curvilinear tensor analysis will recognize $H_{\alpha\beta}$ as the covariant derivative of a_α with respect to b^β, often written as $a_{\alpha|\beta}$. The advantage of the direct matrix notation is clear: because we can imagine $\mathbf{a}$ and $\mathbf{b}$ as vectors in space, we have a physical understanding of what we mean by $\partial\mathbf{a}/\partial\mathbf{b}$; it is the change in the vector-valued function $\mathbf{a}$ as a function of another vector-valued function $\mathbf{b}$. For computations we must express $\mathbf{a}$ and $\mathbf{b}$ in component form. Then

$$H_{\alpha\beta} = a_{\alpha|\beta} = \mathbf{E}_\alpha \cdot \frac{\partial \mathbf{a}}{\partial \mathbf{b}} \cdot \mathbf{e}_\beta$$

provides the necessary components once we have chosen convenient basis systems: $\mathbf{e}_\alpha$ for $\mathbf{b}$ and $\mathbf{E}_\alpha$ for $\mathbf{a}$. Typically $\mathbf{e}_\alpha$ and $\mathbf{E}_\alpha$ will both be the simple rectangular Cartesian bases

$$\begin{aligned}\mathbf{e}_1 &= (1,0,0)\\ \mathbf{e}_2 &= (0,1,0)\\ \mathbf{e}_3 &= (0,0,1)\end{aligned}$$

everywhere. But sometimes we must use more complicated basis systems—examples are when we need quantities associated with the surface of a general shell and when the symmetry of the geometry and, possibly, of the deformation makes it convenient to work in an axisymmetric system. The careful projection of the general results written in direct matrix notation onto the chosen basis system allows us to implement the theory for computation.

As an example, consider the usual expression for strain rate,

$$\dot{\boldsymbol{\varepsilon}} = \frac{1}{2}\left(\frac{\partial \dot{\mathbf{u}}}{\partial \mathbf{x}} + \left[\frac{\partial \dot{\mathbf{u}}}{\partial \mathbf{x}}\right]^T\right),$$

which requires the matrix $\partial \dot{\mathbf{u}}/\partial \mathbf{x}$ to be evaluated, where $\dot{\mathbf{u}}$ is the velocity of the material currently flowing through the point $\mathbf{x}$ in space. Let us now derive the components of $\dot{\boldsymbol{\varepsilon}}$ when the basis system for both $\dot{\mathbf{u}}$ and $\mathbf{x}$ is the cylindrical system that we usually choose for axisymmetric problems, with the basis vectors

$$\begin{aligned}\mathbf{e}_1 \text{ (radial)} &= (\cos\theta, \sin\theta, 0)\\ \mathbf{e}_2 \text{ (axial)} &= (0,0,1)\\ \mathbf{e}_3 \text{ (circumferential)} &= (-\sin\theta, \cos\theta, 0)\end{aligned}$$

(in Abaqus for axisymmetric cases we always take the components in this order—radial, axial, circumferential). These basis vectors are orthogonal and of unit length, so that $\mathbf{e}^\alpha = \mathbf{e}_\alpha$.

We consider position to be defined by the coordinates (r, z, θ), with

$$d\mathbf{x} = dr\,\mathbf{e}_1 + dz\,\mathbf{e}_2 + r\,d\theta\,\mathbf{e}_3,$$

so that

$$dx^1 = dr, \quad dx^2 = dz, \quad \text{and} \quad dx^3 = r\,d\theta.$$

Thus,

$$\frac{\partial \dot{\mathbf{u}}}{\partial x^\alpha} = \left(\frac{\partial \dot{\mathbf{u}}}{\partial r}, \frac{\partial \dot{\mathbf{u}}}{\partial z}, \frac{1}{r}\frac{\partial \dot{\mathbf{u}}}{\partial \theta}\right),$$

where

$$\dot{\mathbf{u}} = \dot{u}_r \mathbf{e}_1 + \dot{u}_z \mathbf{e}_2 + \dot{u}_\theta \mathbf{e}_3,$$

so that

$$\begin{aligned}
\frac{\partial \dot{\mathbf{u}}}{\partial r} &= \frac{\partial \dot{u}_r}{\partial r}\mathbf{e}_1 + \frac{\partial \dot{u}_z}{\partial r}\mathbf{e}_2 + \frac{\partial \dot{u}_\theta}{\partial r}\mathbf{e}_3 \\
\frac{\partial \dot{\mathbf{u}}}{\partial z} &= \frac{\partial \dot{u}_r}{\partial z}\mathbf{e}_1 + \frac{\partial \dot{u}_z}{\partial z}\mathbf{e}_2 + \frac{\partial \dot{u}_\theta}{\partial z}\mathbf{e}_3 \\
\frac{1}{r}\frac{\partial \dot{\mathbf{u}}}{\partial \theta} &= \frac{1}{r}\frac{\partial \dot{u}_r}{\partial \theta}\mathbf{e}_1 + \frac{1}{r}\frac{\partial \dot{u}_z}{\partial \theta}\mathbf{e}_2 + \frac{1}{r}\frac{\partial \dot{u}_\theta}{\partial \theta}\mathbf{e}_3 + \frac{1}{r}\dot{u}_r\frac{\partial \mathbf{e}_1}{\partial \theta} + \frac{1}{r}\dot{u}_\theta\frac{\partial \mathbf{e}_3}{\partial \theta}.
\end{aligned}$$

We know that

$$\frac{\partial \mathbf{e}_1}{\partial \theta} = (-\sin\theta, \cos\theta, 0) = \mathbf{e}_3, \quad \text{and} \quad \frac{\partial \mathbf{e}_3}{\partial \theta} = (-\cos\theta, -\sin\theta, 0) = -\mathbf{e}_1,$$

so that

$$\frac{1}{r}\frac{\partial \dot{\mathbf{u}}}{\partial \theta} = \frac{1}{r}\left(\frac{\partial \dot{u}_r}{\partial \theta} - \dot{u}_\theta\right)\mathbf{e}_1 + \frac{1}{r}\frac{\partial \dot{u}_z}{\partial \theta}\mathbf{e}_2 + \frac{1}{r}\left(\frac{\partial \dot{u}_\theta}{\partial \theta} + \dot{u}_r\right)\mathbf{e}_3,$$

and thus,

$$\frac{\partial \dot{u}_\alpha}{\partial x_\beta} = \begin{bmatrix} \partial \dot{u}_r/\partial r & \partial \dot{u}_r/\partial z & (1/r)(\partial \dot{u}_r/\partial \theta - \dot{u}_\theta) \\ \partial \dot{u}_z/\partial r & \partial \dot{u}_z/\partial z & (1/r)\partial \dot{u}_z/\partial \theta \\ \partial \dot{u}_\theta/\partial r & \partial \dot{u}_\theta/\partial z & (1/r)(\partial \dot{u}_\theta/\partial \theta + \dot{u}_r) \end{bmatrix}.$$

The components of the strain rate are thus

$$\dot{\varepsilon}_{rr} = \frac{\partial \dot{u}_r}{\partial r}, \quad \dot{\varepsilon}_{zz} = \frac{\partial \dot{u}_z}{\partial z}, \quad \dot{\varepsilon}_{\theta\theta} = \frac{1}{r}\left(\frac{\partial \dot{u}_\theta}{\partial \theta} + \dot{u}_r\right),$$

$$\dot{\gamma}_{rz} = 2\dot{\varepsilon}_{rz} = 2\dot{\varepsilon}_{zr} = \frac{\partial \dot{u}_r}{\partial z} + \frac{\partial \dot{u}_z}{\partial r}, \quad \dot{\gamma}_{r\theta} = 2\dot{\varepsilon}_{r\theta} = 2\dot{\varepsilon}_{\theta r} = \frac{1}{r}\left(\frac{\partial \dot{u}_r}{\partial \theta} - \dot{u}_\theta\right) + \frac{\partial \dot{u}_\theta}{\partial r},$$

and

$$\dot{\gamma}_{z\theta} = 2\dot{\varepsilon}_{z\theta} = 2\dot{\varepsilon}_{\theta z} = \frac{1}{r}\frac{\partial \dot{u}_z}{\partial \theta} + \frac{\partial \dot{u}_\theta}{\partial z}.$$

For the case of purely axisymmetric deformation, $\dot{u}_\theta = 0$ and $\partial \dot{u}_r/\partial\theta = \partial \dot{u}_z/\partial\theta = 0$, so these results simplify to the familiar expressions

$$\dot{\varepsilon}_{rr} = \frac{\partial \dot{u}_r}{\partial r}, \quad \dot{\varepsilon}_{zz} = \frac{\partial \dot{u}_z}{\partial z}, \quad \dot{\varepsilon}_{\theta\theta} = \frac{\dot{u}_r}{r}$$

$$\dot{\gamma}_{rz} = \frac{\partial \dot{u}_r}{\partial z} + \frac{\partial \dot{u}_z}{\partial r}, \quad \dot{\gamma}_{r\theta} = \dot{\gamma}_{z\theta} = 0.$$

In summary, direct matrix notation allows us to obtain all our fundamental results without reference to any particular choice of coordinate system. Careful application of the concept of the covariant derivative then allows these general results to be projected into component form for computation.

Virtual quantities

The concepts of virtual displacements and virtual work are fundamental to the development. Virtual quantities are infinitesimally small variations of physical measures, such as displacement, strain, velocity, and so on. The virtual variation of a scalar quantity a is indicated by δa; of a vector or matrix $\mathbf{a}$ by $\delta\mathbf{a}$.

We extend this notation to such expressions as

$$\delta\boldsymbol{\varepsilon} = \text{sym}\left(\frac{\partial \delta\mathbf{v}}{\partial \mathbf{x}}\right),$$

which is the symmetric part of the spatial gradient of a virtual vector field $\delta\mathbf{v}$. This notation corresponds to the virtual rate of deformation (a measure of strain rate) if $\delta\mathbf{v}$ is a virtual velocity field.

Initial and current positions

Most structural problems concern the description of the way a structure behaves as it is loaded and moves from its reference configuration. Thus, we often compare positions of a point in the current (deformed) configuration and a reference configuration that is usually chosen as the configuration when the structure is unloaded or, in the case of geotechnical problems, when the model is subject only to geostatic stresses. To distinguish these configurations, we use lowercase type ($\mathbf{x}$) to indicate the current position and uppercase type ($\mathbf{X}$) to indicate the initial position of the same material point in the same spatial coordinate frame. In Abaqus we almost always store the rectangular Cartesian components of $\mathbf{X}$ and $\mathbf{x}$. The exception is in axisymmetric structures, where radial (r) and axial (z) components are stored.

Nodal variables

So far we have discussed quantities that are considered to be associated with all points in a model. The finite element approximation is based on assuming interpolations, by which displacement, position, and—often—other variables at any material point are defined by a finite number of nodal variables. In this guide we use uppercase superscripts to refer to individual nodal variables or nodal vectors and adopt the summation convention for these indices.

Hence, the interpolation can be written quite generally as

$$\mathbf{a} = \mathbf{N}^N a^N,$$

where $\mathbf{a}$ is some vector-valued function at any point in the structure; $\mathbf{N}^N(g^\alpha)$, $N = 1, 2 \ldots$ up to the total number of variables in the problem, is a set of N vector interpolation functions (these are functions of the material coordinates, g^α); and a^N, $N = 1, 2 \ldots$ is a set of nodal variables.

In some sections in this guide we need to describe operations on nodal variables for the complete system of finite element equations. In these sections we use the classical matrix-vector notation. In this notation $\{a\}$ represents a column vector containing nodal variables, $\lfloor a \rfloor$ represents a row vector, and a matrix is written as $[A]$. Common operations are the scalar product between two vectors,

$$c = \lfloor a \rfloor \{b\}$$

(which is equivalent to $c = a^N b^N$ in index notation) and the matrix-vector product

$$\{c\} = [A] \{b\}$$

(which is equivalent to $c^M = A^{MN} b^N$ in index notation).

1.3 Finite rotations

- “Rotation variables,” Section 1.3.1

1.3.1 ROTATION VARIABLES

Products: Abaqus/Standard Abaqus/Explicit

Since Abaqus contains such capabilities as structural elements (beams and shells) for which it is necessary to define arbitrarily large magnitudes of rotation, a convenient method for storing the rotation at a node is required. The components of a rotation vector $\boldsymbol{\phi}$ are stored as the degrees of freedom 4, 5, and 6 at any node where a rotation is required.

The finite rotation vector, $\boldsymbol{\phi}$, consists of a rotation magnitude, $\phi = \|\boldsymbol{\phi}\|$, and a rotation axis or direction in space, $\mathbf{p} = \boldsymbol{\phi}/\|\boldsymbol{\phi}\|$. Physically, the rotation $\boldsymbol{\phi}$ is interpreted as a rotation by ϕ radians around the axis $\mathbf{p}$. To characterize this finite rotation mathematically, the rotation vector $\boldsymbol{\phi}$ is used to define an orthogonal transformation or rotation matrix. To do so, first define the skew-symmetric matrix $\hat{\boldsymbol{\phi}}$ associated with $\boldsymbol{\phi}$ by the relationships

$$\hat{\boldsymbol{\phi}} \cdot \boldsymbol{\phi} = \mathbf{0} \quad \text{and} \quad \hat{\boldsymbol{\phi}} \cdot \mathbf{v} = \boldsymbol{\phi} \times \mathbf{v} \quad \text{for all vectors} \quad \mathbf{v}\,.$$

$\boldsymbol{\phi}$ is called the axial vector of the skew-symmetric matrix $\hat{\boldsymbol{\phi}}$. In matrix components relative to the standard Euclidean basis, if $\boldsymbol{\phi} = \{\, \phi^1 \quad \phi^2 \quad \phi^3 \,\}^T$, then

$$[\hat{\boldsymbol{\phi}}] = \begin{bmatrix} 0 & -\phi^3 & \phi^2 \\ \phi^3 & 0 & -\phi^1 \\ -\phi^2 & \phi^1 & 0 \end{bmatrix}\,.$$

In what follows, $\hat{\mathbf{a}}$ will be used to denote the skew-symmetric matrix with axial vector $\mathbf{a}$.

A well-known result from linear algebra is that the exponential of a skew-symmetric matrix $\hat{\boldsymbol{\phi}}$ is an orthogonal (rotation) matrix that produces the finite rotation $\boldsymbol{\phi}$. Let the rotation matrix be $\mathbf{C}$, such that $\mathbf{C}^{-1} = \mathbf{C}^T$. Then by definition,

$$\mathbf{C} = \exp[\hat{\boldsymbol{\phi}}] = \mathbf{I} + \hat{\boldsymbol{\phi}} + \frac{1}{2!}\hat{\boldsymbol{\phi}}^2 + \cdots\,.$$

However, the above infinite series has the following closed-form expression:

$$\mathbf{C} = \exp[\hat{\boldsymbol{\phi}}] = \cos\|\boldsymbol{\phi}\|\mathbf{I} + \frac{\sin\|\boldsymbol{\phi}\|}{\|\boldsymbol{\phi}\|}\hat{\boldsymbol{\phi}} + \frac{(1-\cos\|\boldsymbol{\phi}\|)}{\|\boldsymbol{\phi}\|^2}\boldsymbol{\phi}\boldsymbol{\phi}\,. \tag{1.3.1–1}$$

In components,

$$C_{ij} = \cos\phi\,\delta_{ij} + (1-\cos\phi)p_i\,p_j + \sin\phi\,\epsilon_{ikj}p_k,$$

where $\mathbf{p} = \{\, p_1 \quad p_2 \quad p_3 \,\}^T$ and ϵ_{ijk} is the alternator tensor, defined by

$$\epsilon_{123} = \epsilon_{231} = \epsilon_{312} = 1; \quad \epsilon_{132} = \epsilon_{213} = \epsilon_{321} = -1; \text{ all other } \epsilon_{ijk} = 0.$$

It is this closed-form expression that allows the exact and numerically efficient geometric representation of finite rotations.

Quaternion parametrization

Even though Abaqus stores and outputs the rotation vector, quaternion parameters prove to be an efficient and convenient way to treat finite rotations computationally. Let $q_0 \in R$ be a scalar, and let $\mathbf{q} \in R^3$ be a vector field. The quaternion $\boldsymbol{q}$ is simply the pairing

$$\boldsymbol{q} = (q_0, \mathbf{q}) \,.$$

To associate $\boldsymbol{q}$ with the finite rotation vector $\boldsymbol{\phi}$, define the following:

$$q_0 = \cos \|\boldsymbol{\phi}/2\| \quad \text{and} \quad \boldsymbol{q} = \frac{\sin \|\boldsymbol{\phi}/\mathbf{2}\|}{\|\boldsymbol{\phi}\|} \boldsymbol{\phi} \,. \tag{1.3.1–2}$$

By trigonometric identities it follows that the orthogonal matrix $\exp[\hat{\boldsymbol{\phi}}]$ in Equation 1.3.1–1 is given in terms of $\boldsymbol{q}$ as

$$\exp[\hat{\boldsymbol{\phi}}] = (2q_0^2 - 1)\mathbf{I} + 2q_0\hat{\mathbf{q}} + 2\mathbf{q}\mathbf{q} \,. \tag{1.3.1–3}$$

By the convention introduced above, $\hat{\mathbf{q}}$ is the skew-symmetric matrix with axial vector $\mathbf{q}$.

For a more detailed discussion of quaternion algebra and its relation to other representations of finite rotations, see the discussion by Spring (1986).

Compound rotations

A compound rotation is the successive application of two or more rotation fields. In geometrically linear problems compound rotations are obtained simply as the linear superposition of the individual (linearized) rotation vectors. This fact follows directly from the series expansion for $\exp[\hat{\boldsymbol{\phi}}]$. Let $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ be infinitesimal rotations. Thus, $\exp[\hat{\boldsymbol{\phi}}_1] \approx \mathbf{I} + \hat{\boldsymbol{\phi}}_1$, $\exp[\hat{\boldsymbol{\phi}}_2] \approx \mathbf{I} + \hat{\boldsymbol{\phi}}_2$, and

$$\exp[\hat{\boldsymbol{\phi}}_1] \cdot \exp[\hat{\boldsymbol{\phi}}_2] \approx \exp[\hat{\boldsymbol{\phi}}_2] \cdot \exp[\hat{\boldsymbol{\phi}}_1] \approx \mathbf{I} + \hat{\boldsymbol{\phi}}_1 + \hat{\boldsymbol{\phi}}_2 \,.$$

In geometrically nonlinear analysis compound rotations are no longer additive. Furthermore, they are not commutative; that is, the order of application is important. A significant exception occurs when the multiple rotations share the same rotation axis. This special case is investigated further below. A detailed example of a finite compound rotation is given in "Conventions," Section 1.2.2 of the Abaqus Analysis User's Guide.

Let $\mathbf{C}_i$ be the orthogonal transformation representing the compound rotation defined as the product of a set of individual or incremental rotations $\Delta\mathbf{C}_p$, for $p = 1, 2, \ldots, i$. (For the case of specified boundary conditions $\mathbf{C}_i$ is the final product after $\boldsymbol{i}$ steps of all the specified rotations $\Delta\mathbf{C}_p$; for the iterative numerical solution procedure $\mathbf{C}_i$ is the total rotation after $\boldsymbol{i}$ increments, where $\Delta\mathbf{C}_p$, for $p = 1, \ldots, i$,

is the converged rotation field solution at each increment.) By definition, the compound rotation is the product

$$\mathbf{C}_i = \Delta\mathbf{C}_i \cdot \Delta\mathbf{C}_{i-1} \cdots \Delta\mathbf{C}_1 \,,$$

or equivalently by the recursion relation,

$$\mathbf{C}_i = \Delta\mathbf{C}_i \cdot \mathbf{C}_{i-1} \,.$$

It is important to note that $\Delta\mathbf{C}\cdot\mathbf{C}$, which is interpreted as the finite rotation $\Delta\mathbf{C}$ superposed on the finite rotation $\mathbf{C}$, is different from $\mathbf{C}\cdot\Delta\mathbf{C}$, which is interpreted as the finite rotation $\mathbf{C}$ superposed on the finite rotation $\Delta\mathbf{C}$.

Although compound rotations are defined in terms of orthogonal matrices, in a numerical context the rotation vectors (or equivalently the quaternion parameters) associated with the rotation matrices are the degrees of freedom. Compound rotations are performed as follows: Given a quaternion parametrization $\boldsymbol{q} = (\boldsymbol{q}_\mathbf{o}, \mathbf{q})$ and an incremental (finite) rotation $\Delta\boldsymbol{q} = (\Delta q_0, \Delta\mathbf{q})$, where $\Delta\boldsymbol{q}$ is defined in terms of an incremental rotation vector $\Delta\boldsymbol{\phi}$ by Equation 1.3.1–2, the total or compound rotation is given by the quaternion $\boldsymbol{r} = (\boldsymbol{r}_\mathbf{o}, \mathbf{r})$, which is calculated as

$$\mathbf{r} = \Delta\boldsymbol{q} \circ \boldsymbol{q} \,.$$

Here $\circ$ denotes the quaternion product and is defined as

$$\Delta\mathbf{q} \circ \mathbf{q} \stackrel{\text{def}}{=} (\Delta q_0 q_0 - \Delta\mathbf{q}\cdot\mathbf{q}\,,\ \Delta q_0\mathbf{q} + q_0\Delta\mathbf{q} + \Delta\mathbf{q}\times\mathbf{q}) \,. \tag{1.3.1–4}$$

Equation 1.3.1–4 allows for the update of rotation fields without ever calculating the orthogonal matrix from the quaternion and without performing a matrix multiplication. Furthermore, all operations are singularity free regardless of the magnitude of the incremental rotation field $\Delta\boldsymbol{\phi}$. The final (total) rotation vector can be calculated from the quaternion $\mathbf{r}$ by inverting Equation 1.3.1–2.

For the special case when compound rotations share the same rotation axis, the compound rotation reduces to an additive form. Let $\Delta\boldsymbol{q}$ and $\boldsymbol{q}$ have the same rotation axis $\mathbf{p}$. Then $\boldsymbol{q} = (\cos\|\boldsymbol{\phi}/2\|, \sin\|\boldsymbol{\phi}/2\|\mathbf{p})$, $\Delta\boldsymbol{q} = (\cos\|\Delta\boldsymbol{\phi}/2\|, \sin\|\Delta\boldsymbol{\phi}/2\|\mathbf{p})$, and

$$\begin{aligned}\Delta\boldsymbol{q}\circ\boldsymbol{q} = (&\cos\|\Delta\boldsymbol{\phi}/2\|\cos\|\boldsymbol{\phi}/2\| - \sin\|\Delta\boldsymbol{\phi}/2\|\sin\|\boldsymbol{\phi}/2\|\,,\\ &\cos\|\Delta\boldsymbol{\phi}/2\|\sin\|\boldsymbol{\phi}/2\|\mathbf{p} + \cos\|\boldsymbol{\phi}/2\|\sin\|\Delta\boldsymbol{\phi}/2\|\mathbf{p})\,,\end{aligned}$$

which reduces to

$$\Delta\boldsymbol{q}\circ\boldsymbol{q} = (\cos\|(\boldsymbol{\phi}+\Delta\boldsymbol{\phi})/2\|, \sin\|(\boldsymbol{\phi}+\Delta\boldsymbol{\phi})/2\|\mathbf{p}) \,.$$

Rotation vector extraction

For output purposes it is necessary to extract the rotation vector corresponding to a given quaternion. The extraction procedure is as follows: Let $\boldsymbol{r} = (\boldsymbol{r_o}, \mathbf{r})$ be the quaternion, and let $\boldsymbol{\phi}$ be the rotation vector. Thus,

$$\|\boldsymbol{\phi}\| = 2\tan^{-1}\left[\frac{\|\mathbf{r}\|}{r_0}\right] \quad \text{and} \quad \boldsymbol{\phi} = \|\boldsymbol{\phi}\|\frac{\mathbf{r}}{\|\mathbf{r}\|}\,. \tag{1.3.1–5}$$

It is important to note that the extraction of the rotation vector from the quaternion is not unique. The magnitude $\|\boldsymbol{\phi}\|$ is determined only up to the addition of $2n\pi$, $n = \ldots, -1, 0, 1, \ldots$ Abaqus will always choose that rotation vector such that $\|\boldsymbol{\phi}\| < 2\pi$.

Director and rotation field updates

As an example of the utility of the quaternion parameters, consider the incremental update of a director field for either a beam or shell analysis. At some stage of the solution the director field $\mathbf{t}^i$, the quaternion parametrization of the rotation field $\boldsymbol{q^i}$, and the incremental rotation field $\Delta\boldsymbol{\phi}^i$ are known at increment $\boldsymbol{i}$. To update the director field by the incremental rotation to increment $i+1$, proceed as follows: First calculate the quaternion parametrization of the incremental rotation:

$$\Delta q_0^i = \cos\|\Delta\boldsymbol{\phi}^i/2\| \quad \text{and} \quad \Delta\mathbf{q}^i = \frac{\sin\|\Delta\boldsymbol{\phi}^i/2\|}{\|\Delta\boldsymbol{\phi}^i\|}\Delta\boldsymbol{\phi}^i\,.$$

The director field at $i+1$ is then defined as $\mathbf{t}^{i+1} = \exp[\widehat{\Delta\boldsymbol{\phi}}^i]\cdot\mathbf{t}^i$, where $\exp[\widehat{\Delta\boldsymbol{\phi}}^i]$ is calculated with Equation 1.3.1–3. Thus, the director is calculated directly from the quaternion as

$$\mathbf{t}^{i+1} = (2(\Delta q_0^i)^2 - 1)\mathbf{t}^i + 2\Delta q_0^i\Delta\mathbf{q}^i\times\mathbf{t}^i + 2(\Delta\mathbf{q}^i\cdot\mathbf{t}^i)\Delta\mathbf{q}^i\,.$$

Furthermore, the update of the rotation field is obtained by quaternion multiplication $\boldsymbol{q}^{i+1} = \Delta\boldsymbol{q}^i\circ\boldsymbol{q}^i$ and is defined by

$$\boldsymbol{q}^{i+1} = (q_0^{i+1}, \mathbf{q}^{i+1}) = (\Delta q_0^i q_0^i - \Delta\mathbf{q}^i\cdot\mathbf{q}^i, \Delta q_0^i\mathbf{q}^i + q_0^i\Delta\mathbf{q}^i + \Delta\mathbf{q}^i\times\mathbf{q}^i)\,.$$

Variations of the rotation field

In the development of the balance equations, it is necessary to calculate the variation of the rotation field. Consider the vector field $\mathbf{a}$, which is obtained by rotation of the reference vector field $\mathbf{A}$:

$$\mathbf{a} = \mathbf{C}\cdot\mathbf{A}\,.$$

Variations $\delta\mathbf{a}$ in this field are obtained as

$$\delta\mathbf{a} = \delta\mathbf{C}\cdot\mathbf{A}\,,$$

where $\delta\mathbf{C}$ is the linearized rotation matrix; that is, the variation of the orthogonal tensor $\mathbf{C}$. On the other hand, the variation can be defined in terms of the linearized rotation field $\delta\boldsymbol{\theta}$:

$$\delta\mathbf{a} = \delta\boldsymbol{\theta}\times\mathbf{a} = \widehat{\delta\boldsymbol{\theta}}\cdot\mathbf{a} = \widehat{\delta\boldsymbol{\theta}}\cdot\mathbf{C}\cdot\mathbf{A}\,.$$

Consequently, it follows that

$$\delta\mathbf{C} = \widehat{\delta\boldsymbol{\theta}}\cdot\mathbf{C}\,.$$

It is important to note that the linearized rotation $\delta\boldsymbol{\theta}$, which is analogous to the angular velocity in dynamics, is *not* the variation of the rotation vector $\boldsymbol{\phi}$. By a straightforward (but involved) calculation, it can be shown that the variation of the rotation vector ($\delta\boldsymbol{\phi}$) is related to the linearized rotation $\delta\boldsymbol{\theta}$ by

$$\delta\boldsymbol{\theta} = \mathbf{H}(\boldsymbol{\phi})\cdot\delta\boldsymbol{\phi}, \tag{1.3.1–6}$$

where

$$\mathbf{H}(\boldsymbol{\phi}) = \frac{1}{\|\boldsymbol{\phi}\|^2}\boldsymbol{\phi}\boldsymbol{\phi} + \frac{\sin\|\boldsymbol{\phi}\|}{\|\boldsymbol{\phi}\|}\left[\mathbf{I} - \frac{1}{\|\boldsymbol{\phi}\|^2}\boldsymbol{\phi}\boldsymbol{\phi}\right] + \frac{1-\cos\|\boldsymbol{\phi}\|}{\|\boldsymbol{\phi}\|^2}\hat{\boldsymbol{\phi}}\,.$$

The inverse of $\mathbf{H}(\boldsymbol{\phi})$ is

$$\mathbf{H}(\boldsymbol{\phi})^{-1} = \frac{1}{\|\boldsymbol{\phi}\|^2}\boldsymbol{\phi}\boldsymbol{\phi} + \frac{\|\boldsymbol{\phi}\|\sin\|\boldsymbol{\phi}\|}{2(1-\cos\|\boldsymbol{\phi}\|)}\left[\mathbf{I} - \frac{1}{\|\boldsymbol{\phi}\|^2}\boldsymbol{\phi}\boldsymbol{\phi}\right] - \tfrac{1}{2}\hat{\boldsymbol{\phi}}\,.$$

Let $\mathrm{d}\boldsymbol{\theta}$ represent an infinitesimal change in the rotation field. A direct calculation of the variation of $\mathrm{d}\boldsymbol{\theta}$, which is equivalent to calculation of the second variation of either $\mathbf{C}$ or $\mathbf{a}$, leads to an expression that is not symmetric in the variations $\delta\boldsymbol{\theta}$ and the changes $\mathrm{d}\boldsymbol{\theta}$. However, it is shown in Simo (1992) that the correct definition of the Hessian operator—that is, the "covariant" derivative of the weak form of the balance equations—requires only the symmetric part (with respect to the variations) of the second variation. Thus, without loss of generality, we can write

$$\mathrm{d}(\delta\mathbf{C}) = \frac{1}{2}(\widehat{\delta\boldsymbol{\theta}}\cdot\widehat{\mathrm{d}\boldsymbol{\theta}}\cdot\mathbf{C} + \widehat{\mathrm{d}\boldsymbol{\theta}}\cdot\widehat{\delta\boldsymbol{\theta}}\cdot\mathbf{C})\,.$$

Similarly, the second variation of the vector field rotated by $\mathbf{C}$ can be written as

$$\begin{aligned}\mathrm{d}(\delta\mathbf{a}) &= \frac{1}{2}\left[\delta\boldsymbol{\theta}\times(\mathrm{d}\boldsymbol{\theta}\times\mathbf{a}) + \mathrm{d}\boldsymbol{\theta}\times(\delta\boldsymbol{\theta}\times\mathbf{a})\right]\\ &= -(\delta\boldsymbol{\theta}\cdot\mathrm{d}\boldsymbol{\theta})\mathbf{a} + \frac{1}{2}\left[(\delta\boldsymbol{\theta}\cdot\mathbf{a})\mathrm{d}\boldsymbol{\theta} + (\mathrm{d}\boldsymbol{\theta}\cdot\mathbf{a})\delta\boldsymbol{\theta}\right].\end{aligned}$$

Velocity and acceleration

Taking the time derivative of the rotation matrix, we find with the same arguments as used in the calculation of the variations that

$$\dot{\mathbf{C}} = \widehat{\boldsymbol{\omega}} \cdot \mathbf{C}\,,$$
$$\ddot{\mathbf{C}} = \widehat{\dot{\boldsymbol{\omega}}} \cdot \mathbf{C} + \widehat{\boldsymbol{\omega}} \cdot \widehat{\boldsymbol{\omega}} \cdot \mathbf{C}\,,$$

where $\widehat{\boldsymbol{\omega}}$ is the angular velocity matrix. Equivalently, the first and second time derivative of $\mathbf{a}$ are written as

$$\dot{\mathbf{a}} = \boldsymbol{\omega} \times \mathbf{a}\,,$$
$$\ddot{\mathbf{a}} = \dot{\boldsymbol{\omega}} \times \mathbf{a} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{a})\,.$$

The instantaneous angular velocity vector $\boldsymbol{\omega}$ is related to the time rate of change of the rotation vector by the relation

$$\boldsymbol{\omega} = \mathbf{H}(\boldsymbol{\phi}) \cdot \dot{\boldsymbol{\phi}}\,,$$

where $\mathbf{H}(\boldsymbol{\phi})$ is given by Equation 1.3.1–6.

In the linearization of the dynamic balance equations, it is necessary to calculate the variation of the angular velocity, $\mathrm{d}\boldsymbol{\omega}$. This quantity, however, can be calculated only by linearizing the specific algorithm used for the time integration of the dynamic equations.

Coupling of rotations: constant velocity joint

Next, a more rigorous treatment of the two-dimensional constant velocity joint described in "MPC," Section 1.1.14 of the Abaqus User Subroutines Reference Guide, is presented. This derivation exemplifies some of the issues associated with the treatment of finite rotations. "Uniform collapse of straight and curved pipe segments," Section 1.1.5 of the Abaqus Benchmarks Guide, deals with a different finite rotation constraint and tackles additional complications.

Let *a*, *b*, *c* (see Figure 1.3.1–1) be the nodes making up the joint, with *a* the dependent node.The joint is operated by prescribing an axial rotation $\boldsymbol{\phi}^c = \phi^c \mathbf{e}_x$ at *c* and an out-of-plane rotation $\boldsymbol{\phi}^b = \phi^b \mathbf{e}_z$ at *b*. The compounding of these two prescribed rotation fields will determine the total rotation at *a*. We can formally write this constraint as follows:

$$\mathbf{f}(\boldsymbol{\phi}^a, \boldsymbol{\phi}^b, \boldsymbol{\phi}^c) = \boldsymbol{\phi}^a - \boldsymbol{\phi}^b \circ \boldsymbol{\phi}^c = \mathbf{0}.$$

The constraint can be written in terms of the rotation matrices as

$$\mathbf{C}(\boldsymbol{\phi}^a) - \mathbf{C}(\boldsymbol{\phi}^b) \cdot \mathbf{C}(\boldsymbol{\phi}^c) = \mathbf{0}. \tag{1.3.1–7}$$

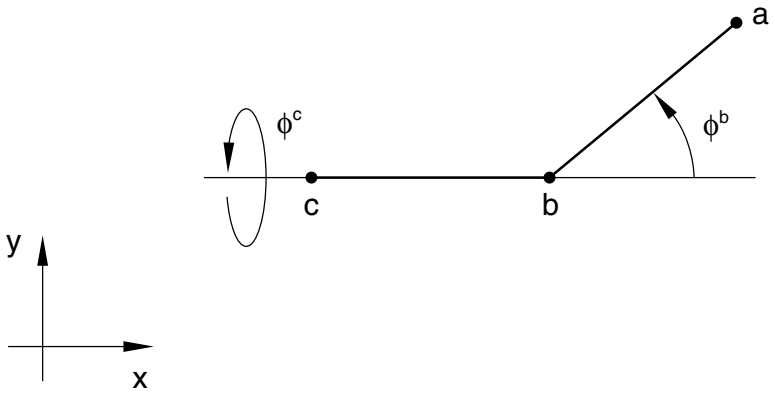


Figure 1.3.1–1 Nonlinear **MPC** example—constant velocity joint.

With the previously defined variational expressions, the constraint can be linearized as

$$\widehat{\delta\boldsymbol{\theta}^a} \cdot \mathbf{C}(\boldsymbol{\phi}^a) - \widehat{\delta\boldsymbol{\theta}^b} \cdot \mathbf{C}(\boldsymbol{\phi}^b) \cdot \mathbf{C}(\boldsymbol{\phi}^c) - \mathbf{C}(\boldsymbol{\phi}^b) \cdot \widehat{\delta\boldsymbol{\theta}^c} \cdot \mathbf{C}(\boldsymbol{\phi}^c) = \mathbf{0}.$$

This expression can be simplified by right-multiplying the expression by $\mathbf{C}^T(\boldsymbol{\phi}^a)$ and by making use of the constraint Equation 1.3.1–7, which yields

$$\widehat{\delta\boldsymbol{\theta}^a} - \widehat{\delta\boldsymbol{\theta}^b} - \mathbf{C}(\boldsymbol{\phi}^b) \cdot \widehat{\delta\boldsymbol{\theta}^c} \cdot \mathbf{C}^T(\boldsymbol{\phi}^b) = \mathbf{0},$$

which can be written in vector form as

$$\delta\boldsymbol{\theta}^a - \delta\boldsymbol{\theta}^b - \mathbf{C}(\boldsymbol{\phi}^b) \cdot \delta\boldsymbol{\theta}^c = \mathbf{0}.$$

Since

$$\mathbf{C}(\boldsymbol{\phi}^b) = \begin{pmatrix} \cos\phi^b & -\sin\phi^b & 0 \\ \sin\phi^b & \cos\phi^b & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

the linearized constraint is indeed identical to the one derived based on simple linear considerations in the Abaqus Analysis User's Guide.

The linearized constraint is used for the calculation of equilibrium. It can also be used for the recovery of the dependent rotation, $\boldsymbol{\phi}^a$, as is done in the Abaqus Analysis User's Guide. The resulting rotation will satisfy the constraint approximately (unless one of the angles ϕ^b or ϕ^c is constant, in which case the constraint is linear and the recovery is exact).

For an exact enforcement of the constraint, user subroutine **MPC** must define the components of the total rotation vector $\boldsymbol{\phi}^a$ exactly. To do so, $\boldsymbol{\phi}^a$ must be updated based on the current values of $\boldsymbol{\phi}^b$ and $\boldsymbol{\phi}^c$. This is most easily accomplished with the aid of the quaternion parameters.

Let $\boldsymbol{q}^b = (\cos(\phi^b/2), \sin(\phi^b/2)\mathbf{e}_z)$ and $\boldsymbol{q}^c = (\cos(\phi^c/2), \sin(\phi^c/2)\mathbf{e}_x)$ be the quaternion parameterizations associated with the finite rotation vectors $\boldsymbol{\phi}^b$ and $\boldsymbol{\phi}^c$, respectively. The total compound rotation $\boldsymbol{\phi}^a$ is given by the quaternion $\boldsymbol{q}^a = (q_0^a, \mathbf{q}^a)$, where

$$
\begin{aligned}
q_0^a &= \cos(\phi^b/2)\cos(\phi^c/2)\,, \\
\mathbf{q}^a &= \cos(\phi^b/2)\sin(\phi^c/2)\mathbf{e}_x + \sin(\phi^b/2)\sin(\phi^c/2)\mathbf{e}_y + \cos(\phi^b/2)\sin(\phi^c/2)\mathbf{e}_z\,,
\end{aligned}
$$

according to the quaternion compound formula Equation 1.3.1–4. The rotation vector $\boldsymbol{\phi}^a$ is extracted from the quaternion $\boldsymbol{q}^a$ as follows:

$$
\boldsymbol{\phi}^a = \phi^a \frac{\mathbf{q}^a}{\|\mathbf{q}^a\|} \quad \text{with} \quad \phi^a = 2\tan^{-1}\left[\frac{\|\mathbf{q}^a\|}{q_0^a}\right],
$$

where $\|\mathbf{q}^a\|$ is the norm of the vector $\mathbf{q}^a$.

"MPC," Section 1.1.14 of the Abaqus User Subroutines Reference Guide, shows the implementation of the linearized form of the constraint in user subroutine **MPC**. The implementation of the exact nonlinear constraint is shown below:

```
      SUBROUTINE MPC(UE,A,JDOF,MDOF,N,JTYPE,X,U,UINIT,MAXDOF,LMPC,
     *     * KSTEP,KINC,TIME,NT,NF,TEMP,FIELD)
      C
            INCLUDE 'ABA_PARAM.INC'
      C
            DIMENSION UE(MDOF), A(MDOF,MDOF,N), JDOF(MDOF,N), X(6,N),
     *     * U(MAXDOF,N), UINIT(MAXDOF,N), TIME(2), TEMP(NT,N),
     *     * FIELD(NF,NT,N)
            PARAMETER( SMALL = 1.E-14 )
      C
            IF ( JTYPE .EQ. 1 ) THEN
              A(1,1,1) =  1.
              A(2,2,1) =  1.
              A(3,3,1) =  1.
              A(3,1,2) = -1.
              A(1,1,3) = -COS(U(6,2))
              A(2,1,3) = -SIN(U(6,2))
      C
              JDOF(1,1) = 4
              JDOF(2,1) = 5
              JDOF(3,1) = 6
              JDOF(1,2) = 6
              JDOF(1,3) = 4
      C
              CPHIB = COS(0.5*U(6,2))
              SPHIB = SIN(0.5*U(6,2))
```

```
            CPHIC = COS(0.5*U(4,3))
            SPHIC = SIN(0.5*U(4,3))
      C
            QA0 = CPHIB*CPHIC
            QAX = CPHIB*SPHIC
            QAY = SPHIB*SPHIC
            QAZ = CPHIB*SPHIC
      C
            QAMAG = SQRT( QAX*QAX + QAY*QAY + QAZ*QAZ )
            IF ( QAMAG .GT. SMALL ) THEN
              PHIA  = 2.*ATAN2( QAMAG , QA0 )
              UE(1) = PHIA*QAX/QAMAG
              UE(2) = PHIA*QAY/QAMAG
              UE(3) = PHIA*QAZ/QAMAG
            ELSE
              UE(1) = 0.
              UE(2) = 0.
              UE(3) = 0.
            END IF
          END IF
      C
          RETURN
          END
```

Reference

- “Conventions,” Section 1.2.2 of the Abaqus Analysis User’s Guide

1.4 Deformation, strain, and strain rates

- "Deformation," Section 1.4.1
- "Strain measures," Section 1.4.2
- "Rate of deformation and strain increment," Section 1.4.3
- "The additive strain rate decomposition," Section 1.4.4

1.4.1 DEFORMATION

Products: Abaqus/Standard Abaqus/Explicit

In any structural problem the analyst describes the initial configuration of the structure and is interested in its deformation throughout the history of loading. The material particle initially located at some position $\mathbf{X}$ in space will move to a new position $\mathbf{x}$: since we assume material cannot appear or disappear, there will be a one-to-one correspondence between $\mathbf{x}$ and $\mathbf{X}$, so we can always write the history of the location of a particle as

$$\mathbf{x} = \mathbf{x}(\mathbf{X}, t) \tag{1.4.1–1}$$

and this relationship can be inverted—we know $\mathbf{X}$ when we know $\mathbf{x}$ and t. Now consider two neighboring particles, located at $\mathbf{X}$ and at $\mathbf{X} + d\mathbf{X}$ in the initial configuration. In the current configuration we must have

$$d\mathbf{x} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} \cdot d\mathbf{X} \tag{1.4.1–2}$$

using the "mapping" Equation 1.4.1–1.

The matrix

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} \tag{1.4.1–3}$$

is called the deformation gradient matrix, and Equation 1.4.1–2 is written

$$d\mathbf{x} = \mathbf{F} \cdot d\mathbf{X}. \tag{1.4.1–4}$$

As the material behavior depends on the straining of the material and not on its rigid body motion, those parts of the motion in the vicinity of a material point must be distinguished. Looking at an infinitesimal gauge length $d\mathbf{X}$ emanating from the particle initially at $\mathbf{X}$, we can measure its initial and current lengths as

$$dL^2 = d\mathbf{X}^T \cdot d\mathbf{X} \quad \text{and} \quad dl^2 = d\mathbf{x}^T \cdot d\mathbf{x},$$

so the "stretch ratio" of this gauge length is

$$\lambda = \frac{dl}{dL} = \sqrt{\frac{d\mathbf{x}^T \cdot d\mathbf{x}}{d\mathbf{X}^T \cdot d\mathbf{X}}}. \tag{1.4.1–5}$$

If $\lambda = 1$, there is no strain of this infinitesimal gauge length—it has undergone rigid body motion only. Now using Equation 1.4.1–4,

$$d\mathbf{x}^T \cdot d\mathbf{x} = d\mathbf{X}^T \cdot \mathbf{F}^T \cdot \mathbf{F} \cdot d\mathbf{X},$$

so that, from Equation 1.4.1–5,

$$\begin{aligned} \lambda^2 &= \frac{d\mathbf{X}^T}{\sqrt{d\mathbf{X}^T \cdot d\mathbf{X}}} \cdot \mathbf{F}^T \cdot \mathbf{F} \cdot \frac{d\mathbf{X}}{\sqrt{d\mathbf{X}^T \cdot d\mathbf{X}}} \\ &= \mathbf{N}^T \cdot \mathbf{F}^T \cdot \mathbf{F} \cdot \mathbf{N}, \end{aligned} \tag{1.4.1–6}$$

where $\mathbf{N}$ is a unit vector in the direction of the gauge length $d\mathbf{X}$.

Equation 1.4.1–6 shows how to measure the stretch ratio associated with any direction, $\mathbf{N}$, at any material point defined by $\mathbf{X}$ (or by $\mathbf{x}$). Useful results are obtained when we vary the direction defined by $\mathbf{N}$ at a particular material point and look for stationary values of the stretch ratio, λ. Since $\mathbf{N}$ must always be a unit vector, stationary values of λ^2 are obtained by solving the constrained variational equation

$$\delta \left\{ \mathbf{N}^T \cdot \mathbf{F}^T \cdot \mathbf{F} \cdot \mathbf{N} - p \left(\mathbf{N}^T \cdot \mathbf{N} - 1 \right) \right\} = 0,$$

where p is a Lagrange multiplier, introduced to retain the constraint

$$\mathbf{N}^T \cdot \mathbf{N} = 1.$$

Taking the variation gives back the constraint (conjugate to δp) and, conjugate to $\delta \mathbf{N}$, gives

$$(\mathbf{F}^T \cdot \mathbf{F} - p\mathbf{I}) \cdot \mathbf{N} = 0. \tag{1.4.1–7}$$

Taking the dot product of the left-hand side of this equation with $\mathbf{N}$ and comparing with Equation 1.4.1–6 identifies $p = \lambda^2$, so Equation 1.4.1–7 is

$$(\mathbf{F}^T \cdot \mathbf{F} - \lambda^2\mathbf{I}) \cdot \mathbf{N} = 0. \tag{1.4.1–8}$$

This problem is an eigenvalue one that can be solved for the three extreme values of λ^2. Since λ is always real and positive (and nonzero), $\lambda^2 > 0$, and hence $\mathbf{F}^T \cdot \mathbf{F}$ must be positive definite. Equation 1.4.1–8 thus gives three real, positive eigenvalues, λ_I, λ_{II}, λ_{III}, the "principal stretches," with three corresponding eigenvectors, $\mathbf{N}_I$, $\mathbf{N}_{II}$, $\mathbf{N}_{III}$, which will be orthogonal, by Equation 1.4.1–8, if the corresponding eigenvalues are different, and can be orthogonalized otherwise. The $\mathbf{N}_I$ are the principal directions of strain.

Now let $\mathbf{n}_I, \mathbf{n}_{II}, \mathbf{n}_{III}$ be unit vectors corresponding to $\mathbf{N}_I, \mathbf{N}_{II}, \mathbf{N}_{III}$, but in the current configuration, so that, using Equation 1.4.1–4,

$$\mathbf{n}_I = \frac{1}{\lambda_I} \mathbf{F} \cdot \mathbf{N}_I, \text{etc.}$$

Then

$$\begin{aligned}\mathbf{n}_I{}^T \cdot \mathbf{n}_{II} &= \frac{1}{\lambda_I \lambda_{II}} \mathbf{N}_I{}^T \cdot \mathbf{F}^T \cdot \mathbf{F} \cdot \mathbf{N}_{II} \\ &= \frac{1}{\lambda_I \lambda_{II}} \lambda_{II}^2 \mathbf{N}_I{}^T \cdot \mathbf{N}_{II} \\ &= 0\end{aligned}$$

by the orthogonality results just mentioned. Thus, $\mathbf{n}_I$, $\mathbf{n}_{II}$, and $\mathbf{n}_{III}$ will also be an orthogonal set. Since each is a unit vector,

$$\begin{aligned}\mathbf{n}_I &= \mathbf{R} \cdot \mathbf{N}_I, \\ \mathbf{n}_{II} &= \mathbf{R} \cdot \mathbf{N}_{II}, \\ \mathbf{n}_{III} &= \mathbf{R} \cdot \mathbf{N}_{III},\end{aligned}$$

where $\mathbf{R}$ is the same pure rigid body rotation matrix in each of these equations. A pure rigid body motion matrix has the property that its inverse is its transpose: $\mathbf{R}^T = \mathbf{R}^{-1}$. Comparing the principal stretch directions in the current and original configurations, therefore, isolates the rigid body rotation and the stretch. Finding the principal stretch ratios and their directions thus provides one solution to the problem of isolating straining motion and rigid body motion in the vicinity of a material point.

Now consider a gauge length in the reference configuration, $\mathrm{d}\mathbf{X}_I$, directed along $\mathbf{N}_I$. The same infinitesimal material line in the current configuration will be along $\mathbf{n}_I$ and will be stretched by λ_I, so that

$$d\mathbf{x}_I = \lambda_I \, \mathbf{R} \cdot d\mathbf{X}_I.$$

Similarly, along the other principal directions,

$$d\mathbf{x}_{II} = \lambda_{II} \, \mathbf{R} \cdot d\mathbf{X}_{II}$$

and

$$d\mathbf{x}_{III} = \lambda_{III} \, \mathbf{R} \cdot d\mathbf{X}_{III}.$$

Since $(\mathbf{N}_I, \mathbf{N}_{II}, \mathbf{N}_{III})$ is an orthonormal set of base vectors in the reference configuration, any infinitesimal material line (gauge length) $d\mathbf{X}$ at $\mathbf{X}$ can be written in terms of its components in this basis:

$$d\mathbf{X} = d\mathbf{X}_I + d\mathbf{X}_{II} + d\mathbf{X}_{III},$$

where

$$d\mathbf{X}_I = \mathbf{N}_I \, \mathbf{N}_I{}^T \cdot d\mathbf{X}, \quad \text{etc.}$$

Each of the vectors $d\mathbf{X}_I$ moves and stretches to the corresponding $d\mathbf{x}_I$, as defined above. Thus, the current gauge length, $d\mathbf{x}$, is

$$
\begin{aligned}
d\mathbf{x} &= d\mathbf{x}_I + d\mathbf{x}_{II} + d\mathbf{x}_{III} \\
&= \lambda_I \, \mathbf{R} \cdot d\mathbf{X}_I + \lambda_{II} \, \mathbf{R} \cdot d\mathbf{X}_{II} + \lambda_{III} \, \mathbf{R} \cdot d\mathbf{X}_{III} \\
&= \left(\lambda_I \, \mathbf{R} \cdot \mathbf{N}_I \, {\mathbf{N}_I}^T + \lambda_{II} \, \mathbf{R} \cdot \mathbf{N}_{II} \, {\mathbf{N}_{II}}^T + \lambda_{III} \, \mathbf{R} \cdot \mathbf{N}_{III} \, {\mathbf{N}_{III}}^T \right) \cdot d\mathbf{X} \\
&= \left(\lambda_I \, \mathbf{n}_I \, {\mathbf{N}_I}^T + \lambda_{II} \, \mathbf{n}_{II} \, {\mathbf{N}_{II}}^T + \lambda_{III} \, \mathbf{n}_{III} \, {\mathbf{N}_{III}}^T \right) \cdot d\mathbf{X} \\
&= \left(\lambda_I \, \mathbf{n}_I \, {\mathbf{n}_I}^T + \lambda_{II} \, \mathbf{n}_{II} \, {\mathbf{n}_{II}}^T + \lambda_{III} \, \mathbf{n}_{III} \, {\mathbf{n}_{III}}^T \right) \cdot \mathbf{R} \cdot d\mathbf{X}
\end{aligned}
$$

which we write as

$$d\mathbf{x} = \mathbf{V} \cdot \mathbf{R} \cdot d\mathbf{X}, \tag{1.4.1–9}$$

where

$$\mathbf{V} = (\lambda_I \, \mathbf{n}_I \, {\mathbf{n}_I}^T + \lambda_{II} \, \mathbf{n}_{II} \, {\mathbf{n}_{II}}^T + \lambda_{III} \mathbf{n}_{III} \mathbf{n}_{III}{}^T) \tag{1.4.1–10}$$

is the "left stretch" matrix, which is the sum of three dyadic products.

Comparison with the definition of the deformation gradient, Equation 1.4.1–4, shows that

$$\mathbf{F} = \mathbf{V} \cdot \mathbf{R}, \tag{1.4.1–11}$$

which is the polar decomposition theorem—that any motion can be represented as a pure rigid body rotation, followed by a pure stretch of three orthogonal directions. The polar decomposition theorem is important because it allows us to distinguish the straining part of the motion from the rigid body rotation. Specifically, $\mathbf{F}$ completely defines the relative motions of material particles in the infinitesimal neighborhood of the material particle that was at $\mathbf{X}$ in the reference configuration; and the left stretch matrix, $\mathbf{V}$, completely defines the deformation of the material particles at $\mathbf{X}$. The rotation matrix $\mathbf{R}$ defines the rigid body rotation of the principal directions of strain ($\mathbf{N}_I$ in the reference configuration; $\mathbf{n}_I$ in the current configuration). $\mathbf{R}$ represents only the rigid body rotation of the material at the point under consideration in some average sense: in a general motion, each infinitesimal gauge length emanating from a material particle has a different amount of rotation. This distinction between the rotation of the principal directions of strain, $\mathbf{R}$, and the rotations of individual directions in the material becomes significant when we must discuss large deformations of nonisotropic materials. Nevertheless, we have established an important result: if $\mathbf{F} = \mathbf{R}$ only, we know there is no deformation of the material in the immediate neighborhood of the point originally at $\mathbf{X}$ and currently at $\mathbf{x}$, since in this case $\mathbf{V} = \mathbf{I}$ and so $\lambda_I = \lambda_{II} = \lambda_{III} = 1$. $\mathbf{V} - \mathbf{I}$ must be nonzero for there to be any deformation of the material at the point in question: in this sense $\mathbf{V} - \mathbf{I}$ (and, hence, $\mathbf{V}$ itself) is sufficient to define the deforming part of the motion (it contains complete information about all except pure rigid body rotation of the point). For this reason—so that, later in the development, we will be able to link the kinematics to the stressing of the material—we will need to be able to isolate $\mathbf{V}$ from $\mathbf{F}$. It is easy to obtain $\mathbf{V} \cdot \mathbf{V}$, for

$$\begin{aligned}\mathbf{F}\cdot\mathbf{F}^T &= \mathbf{V}\cdot\mathbf{R}\cdot\mathbf{R}^T\cdot\mathbf{V}^T\\ &= \mathbf{V}\cdot\mathbf{V},\end{aligned}$$

since $\mathbf{R}^T = \mathbf{R}^{-1}$ and $\mathbf{V}$ is symmetric.

Since we originally defined $\mathbf{V}$ from the principal stretches and their principal directions in the current configuration as

$$\mathbf{V} = \lambda_I \mathbf{n}_I\,\mathbf{n}_I{}^T + \lambda_{II}\mathbf{n}_{II}\,\mathbf{n}_{II}{}^T + \lambda_{III}\mathbf{n}_{III}\,\mathbf{n}_{III}{}^T,$$

then

$$\mathbf{F}\cdot\mathbf{F}^T = \mathbf{V}\cdot\mathbf{V} = \lambda_I{}^2\mathbf{n}_I\,\mathbf{n}_I{}^T + \lambda_{II}{}^2\mathbf{n}_{II}\,\mathbf{n}_{II}{}^T + \lambda_{III}{}^2\mathbf{n}_{III}\,\mathbf{n}_{III}{}^T. \tag{1.4.1–12}$$

We see that the eigenvalues of $\mathbf{F}\cdot\mathbf{F}^T$, are $\lambda_I{}^2$, $\lambda_{II}{}^2$, and $\lambda_{III}{}^2$, and the corresponding eigenvectors are $\mathbf{n}_I$, $\mathbf{n}_{II}$, and $\mathbf{n}_{III}$. We can then construct $\mathbf{V}$. The deformation at the point is, thus, readily obtained by multiplying a 3×3 matrix with its transpose ($\mathbf{F}\cdot\mathbf{F}^T$) and solving the real eigenproblem for the resulting (symmetric) matrix. We can then obtain the rotation $\mathbf{R}$ as

$$\mathbf{R} = \mathbf{V}^{-1}\cdot\mathbf{F}.$$

Since $\mathbf{V}$ has been constructed from its eigenvalues and eigenvectors, its inverse is immediately available:

$$\mathbf{V}^{-1} = \frac{1}{\lambda_I}\mathbf{n}_I\,\mathbf{n}_I^T + \frac{1}{\lambda_{II}}\mathbf{n}_{II}\,\mathbf{n}_{II}^T + \frac{1}{\lambda_{III}}\mathbf{n}_{III}\,\mathbf{n}_{III}^T.$$

So far we have written the results quite generally, without reference to any particular coordinate system. To perform computations we must choose a basis system to express these results as arrays of individual numbers. We now do so with some generality with respect to the choice of basis system. The justification for retaining generality at this stage is twofold: as an exercise, to provide a little more familiarity in the notation system we have chosen to use in this guide, and because we do need some—but, as it turns out, not all—of the generality when we have to deal with shell elements, where it is undesirable to use the rectangular Cartesian base vectors of the global, spatial system because the natural orientation of the shell reference surface causes us to prefer to choose two of the base vectors to be tangent to the shell's reference surface and the other to be normal to this surface. This preference causes us to need two basis systems: one associated with the body in its current configuration, when the point in question is at $\mathbf{x}$, and one associated with the body in its reference configuration, when the same point was at $\mathbf{X}$, because the orientation of the shell's reference surface—which determines our choice of basis vectors—will be quite different in these two configurations. We will write $\mathbf{e}_\alpha$, $\alpha = 1,2,3$, as the basis vectors chosen to write components associated with the current configuration (so that any vector $\mathbf{a}$ associated with the current configuration is written as $\mathbf{a} = a^\alpha\mathbf{e}_\alpha$) and $\mathbf{E}_\alpha$, $\alpha = 1,2,3$, as the basis at the same material point but in the reference configuration. (Since we assume that both of these basis systems are adequate to express any vector-valued function by its components in the basis system—that is, the basis vectors are not linearly dependent—either would, in principal, serve for both configurations. We introduce two distinct systems by preference, because each is chosen as particularly suitable for a particular

configuration.) Since we do not yet impose any particular restrictions on the $\mathbf{e}_\alpha$ or the $\mathbf{E}_\alpha$ (except for the requirement that the vectors must not be linearly dependent), we cannot assume that they will be orthogonal or of unit length: we will, therefore, need to use the corresponding contravariant vectors defined by

$$\mathbf{e}^\alpha \cdot \mathbf{e}_\beta = \delta^\alpha_\beta \quad \text{and} \quad \mathbf{E}^\alpha \cdot \mathbf{E}_\beta = \delta^\alpha_\beta,$$

and the contravariant metric tensors

$$g^{\alpha\beta} = \mathbf{e}^\alpha \cdot \mathbf{e}^\beta \quad \text{and} \quad G^{\alpha\beta} = \mathbf{E}^\alpha \cdot \mathbf{E}^\beta.$$

We can express the deformation gradient, $\mathbf{F}$, numerically by projecting it onto the bases:

$$\mathbf{F} = \mathbf{e}^\alpha F_{\alpha\beta} \mathbf{E}^\beta. \tag{1.4.1–13}$$

Recall the definition of $\mathbf{F}$:

$$d\mathbf{x} = \mathbf{F} \cdot d\mathbf{X} \left(= \frac{\partial \mathbf{x}}{\partial \mathbf{X}} \cdot d\mathbf{X} \right).$$

Since the components of $d\mathbf{x}$ along $\mathbf{e}_\alpha$ are $dx_\alpha = d\mathbf{x} \cdot \mathbf{e}_\alpha$ and we can write $d\mathbf{X} = dX^\beta \mathbf{E}_\beta$,

$$\begin{aligned} dx_\alpha &= \mathbf{e}_\alpha \cdot \mathbf{F} \cdot \mathbf{E}_\beta \, dX^\beta \\ &= \mathbf{e}_\alpha \cdot \frac{\partial \mathbf{x}}{\partial X^\beta} \, dX^\beta. \end{aligned}$$

Thus, writing $dx_\alpha = F_{\alpha\beta} \, dX^\beta$ defines

$$F_{\alpha\beta} = \mathbf{e}_\alpha \cdot \mathbf{F} \cdot \mathbf{E}_\beta = \mathbf{e}_\alpha \cdot \frac{\partial \mathbf{x}}{\partial X^\beta}.$$

We must continue to bear in mind that the first index of $F_{\alpha\beta}$ is associated with a component of $\mathbf{F}$ along a base vector in the current configuration ($\mathbf{e}_\alpha$ in this case), while its second index is associated with a component of $\mathbf{F}$ along a base vector in the reference configuration ($\mathbf{E}_\beta$).

From Equation 1.4.1–13 we can write

$$\begin{aligned} \mathbf{F} \cdot \mathbf{F}^T &= \mathbf{e}^\alpha F_{\alpha\beta} \mathbf{E}^\beta \cdot \mathbf{E}^\gamma F_{\delta\gamma} \mathbf{e}^\delta \\ &= \mathbf{e}^\alpha F_{\alpha\beta} G^{\beta\gamma} F_{\delta\gamma} \mathbf{e}^\delta, \end{aligned}$$

where $G^{\alpha\beta}$ is the contravariant metric of the basis system that we have chosen in the reference configuration.

The eigenproblem for the squared principal stretch ratios and their directions is solved by finding the eigenvalues of the matrix of numbers $F_{\alpha\beta} G^{\beta\gamma} F_{\delta\gamma}$. The eigenvectors will appear as the components $(n_I)_\alpha$ along the $\mathbf{e}^\alpha$ base vectors in the current configuration. Since we have defined the left stretch on the current configuration as

$$\mathbf{V} = \lambda_I \mathbf{n}_I \, \mathbf{n}_I{}^T + \lambda_{II} \mathbf{n}_{II} \, \mathbf{n}_{II}{}^T + \lambda_{III} \mathbf{n}_{III} \, \mathbf{n}_{III}{}^T,$$

we will write its components on the basis in the current configuration as

$$V_{\alpha\beta} = \lambda_I n_{I\alpha}\, n_{I\beta} + \lambda_{II} n_{II\alpha}\, n_{II\beta} + \lambda_{III} n_{III\alpha}\, n_{III\beta} = \mathbf{e}_\alpha \cdot \mathbf{V} \cdot \mathbf{e}_\beta;$$

and, since

$$\mathbf{V}^{-1} = \frac{1}{\lambda_I}\mathbf{n}_I\, \mathbf{n}_I{}^T + \frac{1}{\lambda_{II}}\mathbf{n}_{II}\, \mathbf{n}_{II}{}^T + \frac{1}{\lambda_{III}}\mathbf{n}_{III}\, \mathbf{n}_{III}{}^T,$$

$$(V^{-1})_{\alpha\beta} = \frac{1}{\lambda_I} n_{I\alpha}\, n_{I\beta} + \frac{1}{\lambda_{II}} n_{II\alpha}\, n_{II\beta} + \frac{1}{\lambda_{III}} n_{III\alpha}\, n_{III\beta}.$$

The polar decomposition gives

$$\begin{aligned}\mathbf{R} &= \mathbf{V}^{-1} \cdot \mathbf{F} \\ &= \mathbf{e}^\alpha (V^{-1})_{\alpha\gamma} \mathbf{e}^\gamma \cdot \mathbf{e}^\delta F_{\delta\beta} \mathbf{E}^\beta \\ &= \mathbf{e}^\alpha (V^{-1})_{\alpha\gamma} g^{\gamma\delta} F_{\delta_\beta} \mathbf{E}^\beta,\end{aligned}$$

so

$$R_{\alpha\beta} = (V^{-1})_{\alpha\gamma} g^{\gamma\delta} F_{\delta_\beta},$$

where $g^{\gamma\delta}$ is the contravariant metric tensor of the basis system we have chosen to use in the current configuration and—as with $F_{\alpha\beta}$—we see that the first index of $R_{\alpha\beta}$ is associated with the contravariant base vector $\mathbf{e}^\alpha$ in the current configuration, while the second index is associated with the contravariant base vector $\mathbf{E}^\beta$ in the reference configuration.

We should take care to understand the distinction between the direct matrix expression of the rigid body rotation of the principal directions of strain of the material, $\mathbf{R}$, and the components of $\mathbf{R}$ expressed on a particular basis. Suppose, for example, that the rigid body rotation at a point is zero (that is, $\mathbf{R} = \mathbf{I}$) but we, nevertheless, have chosen different basis systems $\mathbf{e}_\alpha$ and $\mathbf{E}_\alpha$. In this case $R_{\alpha\beta} = \mathbf{e}_\alpha \cdot \mathbf{R} \cdot \mathbf{E}_\beta = \mathbf{e}_\alpha \cdot \mathbf{I} \cdot \mathbf{E}_\beta = \mathbf{e}_\alpha \cdot \mathbf{E}_\beta$. This implies that, even though $\mathbf{R}$ is a unit matrix (in the sense that operating on any vector with this matrix makes no change in that vector), the numerical values we have chosen to store the matrix—the $R_{\alpha\beta}$—do not form a unit matrix of numbers unless the $\mathbf{e}_\alpha$ and the $\mathbf{E}_\alpha$ are coincident and orthonormal. Thus, our choice of quite general basis systems that are not the same in the current and reference configurations (introduced as being "natural" for writing results for shells) somewhat complicates the interpretation of the numbers we store.

In the previous few paragraphs we have chosen to explore the expression of the basic results we have derived so far for the kinematics of the total motion in terms of quite general basis systems, $\mathbf{e}_\alpha$ and $\mathbf{E}_\alpha$. In Abaqus we wish to express results as simply and directly as possible, and we can do so by choosing particular sets of basis vectors that offer the most convenience for our purposes. First, we take the $\mathbf{e}_\alpha$ (and, by extension, the $\mathbf{E}_\alpha$, since these are just the $\mathbf{e}_\alpha$ at the beginning of the motion) to be a local, orthonormal system at each point. Although it is not possible to construct a Cartesian system with orthonormal base vectors over a general shell surface, we can always project the general results onto such a system when that system is chosen specifically at each point where we need to make the projection—typically at the integration points of the

elements. The choice of which system is used as this local orthonormal basis is made in Abaqus at two levels: we distinguish continuum ("solid") elements from structural (shell and beam) elements, and we distinguish the default choice of directions from the particular choice of directions (orientation) specified by the user. For continuum elements the default $\mathbf{E}_\alpha$ are unit vectors along the axes of the global Cartesian system chosen for the problem. At points where the orientation is defined by the user, the specified $\mathbf{E}_\alpha$ are used. For shells (and membranes) we take $\mathbf{E}_1$ and $\mathbf{E}_2$ tangent to the shell's reference surface and $\mathbf{E}_3$ normal to that surface at the point under consideration. By default, $\mathbf{E}_1$ is the projection of the global x-axis onto the reference surface or, if the global x-axis is almost normal to that surface at the point, $\mathbf{E}_1$ is the projection of the global z-axis onto the surface. If the orientation is defined by the user, $\mathbf{E}_1$ and $\mathbf{E}_2$ are the projections of the two specified axes onto the reference surface at the point. In all cases $\mathbf{E}_3$ is normal to the shell's reference surface. For beams $\mathbf{E}_1$ is along the beam axis, with $\mathbf{E}_2$ and $\mathbf{E}_3$ defined from the beam section definition option and beam normals given as part of the nodal coordinate definition. For continuum elements the same schemes are applied by default to define the basis system in the current configuration. For continuum elements with the orientation specified by the user and in all cases for shells, beams, and membranes, the $\mathbf{e}_\alpha$ are defined by

$$\mathbf{e}_\alpha = \mathbf{R} \cdot \mathbf{E}_\alpha.$$

These schemes all have the same property: at any point in time the $\mathbf{e}_\alpha$ are orthonormal vectors: $\mathbf{e}_\alpha \cdot \mathbf{e}_\beta = \delta_{\alpha\beta}$, so $\mathbf{e}^\alpha = \mathbf{e}_\alpha$ and, thus, $g_{\alpha\beta} = g^{\alpha\beta} = g^\beta_\alpha = \delta_{\alpha\beta}$, and—in particular—$\mathbf{E}_\alpha \cdot \mathbf{E}_\beta = \delta_{\alpha\beta}$ and, thus, $G^{\alpha\beta} = \delta^{\alpha\beta}$. This simplifies the understanding of all the quantities we write, since the components of any tensor $T_{\alpha\beta\ldots}$ are always the physical projections of that tensor-valued quantity on the local orthogonal basis system $\mathbf{e}_\alpha$ and we need not distinguish covariant and contravariant components as we did in the general development above. In practical terms the only price we must pay for this simplicity is in shells when we have to use a separate basis system at each point under study, since we cannot construct a single system with the orthonormal property on a general curved surface. (In an axisymmetric system we also have to use $dx^3 = r\,d\theta$ to ensure that the $\mathbf{e}_3$ base vector is a unit vector, but this is a minor point.) The simplifications are valuable and, from our perspective of studying finite element formulations, they are bought at modest cost, since we generally only consider a single integration point at a time. Throughout the rest of this guide, whenever we need to write down particular components of a tensor, we shall assume that the basis on which they are written has the orthonormal property $\mathbf{e}_\alpha \cdot \mathbf{e}_\beta = \delta_{\alpha\beta}$.

The material also undergoes rigid body translation, but this is not important in the development since we need consider only relative motion of neighboring points because we are interested in the deformation of the material to link the kinematics of the motion to the material's constitutive behavior. Numerically, rigid body translation is significant only for two reasons. One is that the spatial discretization must allow rigid body translation without giving strain, which is important in choosing interpolation functions for the finite elements. The other is that care must be exercised to ensure that the strain and rotation are calculated accurately when the rigid body motion is large, since then the strain and rotation depend on the difference between two very large motions.

Reference

- "Conventions," Section 1.2.2 of the Abaqus Analysis User's Guide

1.4.2 STRAIN MEASURES

Products: Abaqus/Standard Abaqus/Explicit

Strain measures used in general motions are most simply understood by first considering the concept of strain in one dimension and then generalizing this to arbitrary motions by using the polar decomposition theorem just derived.

Strain in one dimension

We already have a measure of deformation—the stretch ratio λ. In fact, λ is itself an adequate measure of "strain" for a number of problems. To see where it is useful and where not, first notice that the unstrained value of λ is 1.0. A typical soft rubber component (such as a rubber band) can change length by a large factor when it is loaded, so the stretch ratio λ would often have values of 2 or more. In contrast, a typical structural steel component will be designed to respond elastically to its working loads. Such a material has an elastic modulus of about 200×10^3 MPa (30×10^6 lb/in^2) at room temperature and a yield stress of about 200 MPa (30×10^3 lb/in^2), so the stretch at yield will be about 1.001 in tension, 0.999 in compression. The stretch ratio is an unsatisfactory way of measuring deformation for this case because the numbers of interest begin in the fourth significant digit. To avoid this inconvenience, the concept of strain is introduced, the basic idea being that the strain is zero at $\lambda = 1$, when the material is "unstrained." In one dimension, along some "gauge length" $d\mathbf{X}$, we define strain as a function of the stretch ratio, λ, of that gauge length:

$$\varepsilon = f(\lambda).$$

The objective of introducing the concept of strain is that the function $\boldsymbol{f}$ is chosen for convenience. To see what this implies, suppose ε is expanded in a Taylor series about the unstrained state:

$$\varepsilon = f(1) + (\lambda - 1)\frac{df}{d\lambda} + \frac{1}{2!}(\lambda - 1)^2\frac{d^2 f}{d\lambda^2} + \cdots \tag{1.4.2–1}$$

We must have $f(1) = 0$, so $\varepsilon = 0$ at $\lambda = 1$ (this was the main reason for introducing this idea of "strain" instead of just using the stretch ratio). In addition, we choose $df/d\lambda = 1$ at $\lambda = 1$ so that for small strains we have the usual definition of strain as the "change in length per unit length." This ensures that, in one dimension, all strain measures defined in this way will give the same numerical value to the order of the approximation when strains are small (because then the higher-order terms in the Taylor series are all negligible)—regardless of the magnitude of any rigid body rotation. Finally, we require that $df/d\lambda > 0$ for all physically reasonable values of λ (that, is for all $\lambda > 0$) so that strain increases monotonically with stretch; hence, to each value of stretch there corresponds a unique value of strain. (The choice of $df/d\lambda > 0$ is arbitrary: we could equally well choose $df/d\lambda < 0$, implying that the strain is positive in compression when $\lambda < 1$. This alternative choice is often made in geomechanics textbooks because geotechnical problems usually involve compressive stress and strain. The choice is a matter of

convenience. In Abaqus we always use the convention that positive direct strains represent tension when $\lambda > 1$. This choice is retained consistently in Abaqus, including in the geotechnical options.)

With these reasonable restrictions ($f = 0$ and $df/d\lambda = 1$ at $\lambda = 1$, and $df/d\lambda > 0$ for all $\lambda > 0$), many strain measures are possible, and several are commonly used. Some examples are

$$\text{Nominal strain (Biot's strain):} \quad f(\lambda) = \lambda - 1.$$

In a uniformly strained uniaxial specimen, where l is the current and L the original gauge length, this strain is measured as $(l/L) - 1$. This definition is the most familiar one to engineers who perform uniaxial testing of stiff specimens.

$$\text{Logarithmic strain:} \quad f(\lambda) = \ln \lambda.$$

This strain measure is commonly used in metal plasticity. One motivation for this choice in this case is that, when "true" stress (force per current area) is plotted against log strain, tension, compression and torsion test results coincide closely. Later we will see that this strain measure is mathematically appropriate for such materials because, for these materials, the elastic part of the strain can be assumed to be small.

$$\text{Green's strain:} \quad f(\lambda) = \frac{1}{2}\left(\lambda^2 - 1\right).$$

This strain measure is convenient computationally for problems involving large motions but only small strains, because, as we will show later, its generalization to a strain tensor in any three-dimensional motion can be computed directly from the deformation gradient without requiring solution for the principal stretch ratios and their directions.

All of these strains satisfy the basic restrictions. Obviously many strain functions are possible: the choice is strictly a matter of convenience. Since strain is usually the link between the kinematic and the constitutive theories, the convenience of this choice in the context of finite elements is based on two considerations: the ease with which the strain can be computed from the displacements, since the latter are usually the basic variables in the finite element model, and the appropriateness of the strain measure with respect to the particular constitutive model. For example, as mentioned above, it appears that log strain is particularly appropriate to plasticity, while large-strain elasticity analysis (for rubbers and similar materials) can be done quite satisfactorily without ever using any "strain" measure except the stretch ratio λ.

Strain in general three-dimensional motions

Having defined the basic concept of "strain" in one dimension, we now generalize the idea to three dimensions. In "Deformation," Section 1.4.1, we established that the deforming part of the motion in the immediate neighborhood of a material point is completely characterized by six variables: the three principal stretch ratios (λ_I, λ_{II}, and λ_{III}) and the orientation of the three principal stretch directions in the current (or in the reference) configuration. This immediately gives the generalization of the one-dimensional strain function introduced above. We first choose the function f that will be used as the

strain measure. $\varepsilon_I = f(\lambda_I)$ will be the strain along the first principal direction, $\mathbf{n}_I$; $\varepsilon_{II} = f(\lambda_{II})$ will be the strain along $\mathbf{n}_{II}$; and $\varepsilon_{III} = f(\lambda_{III})$ will be the strain along $\mathbf{n}_{III}$.

The matrix

$$\boldsymbol{\varepsilon} = \varepsilon_I \, \mathbf{n}_I \mathbf{n}_I{}^T + \varepsilon_{II} \, \mathbf{n}_{II} \mathbf{n}_{II}{}^T + \varepsilon_{III} \, \mathbf{n}_{III} \mathbf{n}_{III}{}^T \tag{1.4.2–2}$$

completely characterizes the state of strain at the material point. Notice the resemblance to the definition of the stretch matrix, Equation 1.4.1–10: we might consider $\boldsymbol{\varepsilon}$ to be defined by the matrix function

$$\boldsymbol{\varepsilon} = f(\mathbf{V}),$$

where we understand a matrix function to mean that the two matrices have the same principal directions with their principal values related by the definition of f, which is a convenient shorthand way of indicating a relationship between two matrices.

In Equation 1.4.2–2 we have written the matrix $\boldsymbol{\varepsilon}$ by using the principal strain directions in the current configuration. We could equally have begun with the polar decomposition into a stretch followed by rotation of the principal directions of stretch: $\boldsymbol{\varepsilon}$ would be defined in a similar way and would then be associated with its principal directions in the reference configuration. Abaqus generally reports strains referred to directions in the current configuration. There is no obvious reason for this choice: either approach would suffice so long as the user knows which is being used. The strain measures reported by Abaqus are enumerated in "Conventions," Section 1.2.2 of the Abaqus Analysis User's Guide.

In a finite element code the deformation gradient $\mathbf{F}$ is usually computed at each material calculation point from the displacement solution at the nodes of each element and the interpolation function chosen for the element. We now need an algorithm to obtain $\boldsymbol{\varepsilon}$, given a choice of strain measure. This algorithm is available immediately from Equation 1.4.1–12: the eigenvalues and eigenvectors of the 3×3 matrix $\mathbf{F} \cdot \mathbf{F}^T$ are ${\lambda_I}^2$; ${\lambda_{II}}^2$ and ${\lambda_{III}}^2$; and $\mathbf{n}_I$, $\mathbf{n}_{II}$, and $\mathbf{n}_{III}$. We can then calculate $\varepsilon_I = f(\lambda_I)$, etc. for the function f chosen as the strain measure and, thus, construct

$$\boldsymbol{\varepsilon} = \varepsilon_I \, \mathbf{n}_I \mathbf{n}_I{}^T + \varepsilon_{II} \, \mathbf{n}_{II} \mathbf{n}_{II}{}^T + \varepsilon_{III} \, \mathbf{n}_{III} \mathbf{n}_{III}{}^T.$$

This algorithm also gives principal strain and stretch values—often a useful output because they give a concise description of the state of deformation at a point. However, the algorithm requires computation of the eigenvalues and eigenvectors of a 3×3 matrix at each of many points in the model at each of many iterations, which involves some computational cost. Thus, it would be useful if $\boldsymbol{\varepsilon}$ could be computed less expensively from $\mathbf{F}$, which is possible only for certain choices of the strain measure, $f(\lambda)$. We now consider one such possibility.

The unit matrix $\mathbf{I}$ can be written as

$$\mathbf{I} = \mathbf{n}_I \mathbf{n}_I{}^T + \mathbf{n}_{II} \mathbf{n}_{II}{}^T + \mathbf{n}_{III} \mathbf{n}_{III}{}^T.$$

Using Equation 1.4.1–12,

$$\begin{aligned}\mathbf{F}\cdot\mathbf{F}^T-\mathbf{I}&=\mathbf{V}\cdot\mathbf{V}-\mathbf{I}\\&=(\lambda_I^2-1)\,\mathbf{n}_I{\mathbf{n}_I}^T+(\lambda_{II}^2-1)\,\mathbf{n}_{II}{\mathbf{n}_{II}}^T+(\lambda_{III}^2-1)\,\mathbf{n}_{III}{\mathbf{n}_{III}}^T.\end{aligned}\tag{1.4.2–3}$$

Green's strain was defined in one dimension as

$$\varepsilon^G=\frac{1}{2}(\lambda^2-1).$$

Comparing this one-dimensional definition with Equation 1.4.2–2 and Equation 1.4.2–3, we see that

$$\boldsymbol{\varepsilon}^g=\frac{1}{2}(\mathbf{F}\cdot\mathbf{F}^T-\mathbf{I})$$

is then a generalization of Green's strain in one dimension. (The more standard definition of Green's strain matrix is obtained by using $\mathbf{F}^T\cdot\mathbf{F}$ instead of $\mathbf{F}\cdot\mathbf{F}^T$, so the strain matrix is taken on the reference configuration instead of the current configuration as a basis:

$$\boldsymbol{\varepsilon}^G=\frac{1}{2}(\mathbf{F}^T\cdot\mathbf{F}-\mathbf{I}).$$

The definition we have adopted is consistent with taking the strain matrix on the current configuration. The only difference between the two definitions is the configuration in which the matrix is defined—whether we think of the motion as rigid body rotation of the principal axes of stretch, $\mathbf{R}$, followed by stretch along those axes, $\mathbf{V}$, or stretch along the principal axes, $\mathbf{U}$, followed by rigid body rotation of those axes, $\mathbf{R}$. The choice is arbitrary.)

Green's strain matrix is, thus, available directly from the deformation gradient without first having to solve for the principal directions. This advantage makes Green's strain computationally attractive. Recall that strain is the link between the kinematics and the constitutive theory, so the strain choice should be optimal based on the two considerations of convenience and appropriateness. We have already suggested that logarithmic strain is the most appropriate for elastic-plastic or elastic-viscoplastic materials in which the elastic strains are always small (because the yield stress is small compared to the elastic modulus), so it appears that the computational convenience of Green's strain cannot be used to advantage. However, the choice of a strain function, $f(\lambda)$, was restricted so that, for small strains but arbitrary rotations, all strain measures are the same to the order of the approximation. Thus, for such cases Green's strain is a very convenient choice for computing the strain. The small-strain, large-rotation approximation is often useful—especially in structural problems (shells and beams) because there the thinness or slenderness of the members often allows large rotations to occur with quite small-strains—and Green's strain is commonly used in large-rotation, small-strain formulations for such problems as shell buckling.

Finally, it is worth remarking that the familiar "small-strain" measure used in most elementary elasticity textbooks,

$$\boldsymbol{\varepsilon}=\frac{1}{2}\left(\frac{\partial\mathbf{u}}{\partial\mathbf{X}}+\left[\frac{\partial\mathbf{u}}{\partial\mathbf{X}}\right]^T\right),$$

is useful only for small displacement gradients—that is, both the strains and the rotations must be small for this strain measure to be appropriate. This can be demonstrated by considering pure rotation of a specimen: even though the material is not stretched, the components of this measure of strain become nonzero as the rotation increases.

Reference

- "Conventions," Section 1.2.2 of the Abaqus Analysis User's Guide

1.4.3 RATE OF DEFORMATION AND STRAIN INCREMENT

Products: Abaqus/Standard Abaqus/Explicit

Many of the materials we need to model are path dependent, so usually the constitutive relationships are defined in rate form, which requires a definition of strain rate. The velocity of a material particle is

$$\mathbf{v} = \frac{\partial \mathbf{x}}{\partial t},$$

where the partial differentiation with respect to time (t) means the rate of change of the spatial position, $\mathbf{x}$, of a fixed material particle. Here, again, we take the Lagrangian viewpoint: we observe a material particle and follow it through the motion, rather than looking at a fixed point in space and watching the material flowing through this point. The Lagrangian point of view is used for the mechanical modeling capabilities in Abaqus because we are usually dealing with history-dependent materials and the Lagrangian perspective makes it easy to record and update the state of a material point since the mesh is glued to the material.

The velocity difference between two neighboring particles in the current configuration is

$$d\mathbf{v} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}} \cdot d\mathbf{x} = \mathbf{L} \cdot d\mathbf{x},$$

where

$$\mathbf{L} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}} \tag{1.4.3–1}$$

is the velocity gradient in the current configuration.

In "Deformation," Section 1.4.1, we introduced the definition of the deformation gradient matrix, $\mathbf{F}$:

$$d\mathbf{x} = \mathbf{F} \cdot d\mathbf{X},$$

so

$$d\mathbf{v} = \mathbf{L} \cdot d\mathbf{x} = \mathbf{L} \cdot \mathbf{F} \cdot d\mathbf{X}.$$

We could also obtain the velocity difference directly by

$$d\mathbf{v} = \frac{\partial}{\partial t}(\mathbf{F} \cdot d\mathbf{X}) = \dot{\mathbf{F}} \cdot d\mathbf{X},$$

where

$$\dot{\mathbf{F}} = \frac{\partial \mathbf{F}}{\partial t},$$

because $d\mathbf{v}$ is defined as the velocity difference between two neighboring material particles and, having chosen these particles, the gauge length between them in the reference configuration, $d\mathbf{X}$, is the same throughout the motion and, so, has no time derivative.

Comparing the two expressions for $d\mathbf{v}$ in terms of the reference configuration gauge length $d\mathbf{X}$, we see that

$$\mathbf{L} \cdot \mathbf{F} = \dot{\mathbf{F}}$$

or

$$\mathbf{L} = \dot{\mathbf{F}} \cdot \mathbf{F}^{-1}.$$

Now $\mathbf{L}$ will be composed of a rate of deformation and a rate of rotation or spin. Since these are rate quantities, the spin can be treated as a vector; thus, we can decompose $\mathbf{L}$ into a symmetric strain rate matrix and an antisymmetric rotation rate matrix, just as in small motion theory we decompose the infinitesimal displacement gradient into an infinitesimal strain and an infinitesimal rotation. The symmetric part of the decomposition is the strain rate (it is called the rate of deformation tensor in many textbooks and is also commonly denoted as $\mathbf{D}$) and is

$$\begin{aligned}\dot{\boldsymbol{\varepsilon}} &= \frac{1}{2}\left(\mathbf{L} + \mathbf{L}^T\right) \\ &= \frac{1}{2}\left(\left[\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right] + \left[\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right]^T\right).\end{aligned}$$

The antisymmetric part of the decomposition is the spin matrix,

$$\begin{aligned}\mathbf{W} &= \frac{1}{2}\left(\mathbf{L} - \mathbf{L}^T\right) \\ &= \frac{1}{2}\left(\left[\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right] - \left[\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right]^T\right).\end{aligned}$$

These are particularly simple and familiar forms; for example, $\dot{\boldsymbol{\varepsilon}}$ is identical to the elementary definition of "small strain" if we replace the particle velocity, $\mathbf{v}$, with the displacement, $\mathbf{u}$. In one dimension $\dot{\boldsymbol{\varepsilon}}$ is

$$\dot{\varepsilon} = \frac{dv}{dx},$$

which identifies $\dot{\varepsilon}$ as the rate of logarithmic strain,

$$\varepsilon = \ln \lambda.$$

This interpretation would also be correct if the principal directions of strain rotate along with the rigid body motion (because the identification can be applied to each principal value of the logarithmic strain matrix). In the general case, when the principal strain directions rotate independent of the material, $\dot{\boldsymbol{\varepsilon}}$ is not integrable

into a total strain measure. Nevertheless, the identification of $\dot{\boldsymbol{\varepsilon}}$ with the rate of logarithmic strain in the particular case of nonrotating principal directions provides a useful interpretation of the logarithmic measure of strain as a "natural" strain if we think of $\dot{\boldsymbol{\varepsilon}}$, as it is defined above as the symmetric part of the velocity gradient with respect to current spatial position, as a "natural" measure of strain rate.

The typical inelastic constitutive model requires as input a small but finite strain increment $\Delta\boldsymbol{\varepsilon}$, as well as vector and tensor valued state variables (such as the stress) that are written on the current configuration. In Abaqus/Explicit and for shell and membrane elements in Abaqus/Standard, a slightly different algorithm is used to calculate $\Delta\mathbf{R}$. For most element types in Abaqus/Standard we approach this problem by first using the polar decomposition in the increment to define the change in the average material rotation over the increment, $\Delta\mathbf{R}$, from the total deformation in the increment, $\Delta\mathbf{F}$:

$$\Delta\mathbf{F} = \Delta\mathbf{V} \cdot \Delta\mathbf{R}.$$

All vectors and tensors associated with the material (whose values are available at the beginning of the increment from previous calculations) can now be rotated to the configuration at the end of the increment, solely to account for the rigid body rotation in the increment:

$$\mathbf{a} \Rightarrow \Delta\mathbf{R} \cdot \mathbf{a}$$

for a vector, and

$$\mathbf{A} \Rightarrow \Delta\mathbf{R} \cdot \mathbf{A} \cdot \Delta\mathbf{R}^T$$

for a tensor.

These rotated variables are now passed to the constitutive routines, which may provide further updates to them because of constitutive effects. These constitutive effects will be associated with deformation, which must be supplied in the form of the strain increment $\Delta\boldsymbol{\varepsilon}$. For this we proceed as follows.

Since we assume $\Delta\mathbf{R}$ rotates the deformation basis—in the sense that it rotates the principal axes of deformation and, thus, provides a measure of average material rotation—we can define the velocity gradient $\mathbf{L}$ at any time during the increment, referred to the fixed basis at $t + \Delta t$, as

$$\check{\mathbf{L}} = \Delta\mathbf{R}(t+\Delta t) \cdot \Big(\Delta\mathbf{R}(t+\tau)^T \cdot \mathbf{L} \cdot \Delta\mathbf{R}(t+\tau) \Big) \cdot \Delta\mathbf{R}(t+\Delta t)^T, \quad 0 \le \tau \le \Delta t. \tag{1.4.3–2}$$

Then our integration of $\dot{\boldsymbol{\varepsilon}}$ is the matrix $\Delta\boldsymbol{\varepsilon}$, on the basis at the end of the increment, and defined by

$$\Delta\boldsymbol{\varepsilon} = \int_0^{\Delta t} \operatorname{sym}\big(\check{\mathbf{L}}(t+\tau)\big)\, d\tau.$$

Using Equation 1.4.3–2, this is

$$\Delta\boldsymbol{\varepsilon} = \frac{1}{2}\Delta\mathbf{R}(t+\Delta t) \cdot \int_0^{\Delta t} \Delta\mathbf{R}(t+\tau)^T \cdot \Big(\mathbf{L}(t+\tau) + \mathbf{L}^T(t+\tau) \Big) \cdot \Delta\mathbf{R}(t+\tau)\, d\tau \cdot \Delta\mathbf{R}(t+\Delta t)^T. \tag{1.4.3–3}$$

Since

$$\mathbf{L} = \frac{d}{dt}(\Delta\mathbf{F}) \cdot \Delta\mathbf{F}^{-1},$$

we can make use of the polar decomposition of the increment of deformation into a stretch on the axes at the start of the increment followed by rotation ($\Delta\mathbf{F} = \Delta\mathbf{R} \cdot \Delta\mathbf{U}$) to write

$$\begin{aligned}\mathbf{L} &= \frac{d}{dt}(\Delta\mathbf{F}) \cdot \Delta\mathbf{F}^{-1} \\ &= \left(\frac{d}{dt}(\Delta\mathbf{R}) \cdot \Delta\mathbf{U} + \Delta\mathbf{R} \cdot \frac{d}{dt}\Delta\mathbf{U}\right) \cdot \left(\Delta\mathbf{U}^{-1} \cdot \Delta\mathbf{R}^{T}\right) \\ &= \frac{d}{dt}(\Delta\mathbf{R}) \cdot \Delta\mathbf{R}^{T} + \Delta\mathbf{R} \cdot \frac{d}{dt}(\Delta\mathbf{U}) \cdot \Delta\mathbf{U}^{-1} \cdot \Delta\mathbf{R}^{T}\end{aligned}$$

so that the integrand in the definition of the increment of strain is

$$\Delta\mathbf{R}^{T} \cdot \left(\mathbf{L} + \mathbf{L}^{T}\right) \cdot \Delta\mathbf{R} = \frac{d}{dt}(\Delta\mathbf{U}) \cdot \Delta\mathbf{U}^{-1} + \Delta\mathbf{U}^{-1} \cdot \frac{d}{dt}(\Delta\mathbf{U}).$$

We now assume that the incremental stretch at any time in the increment written on the basis at the beginning of the increment, $\Delta\mathbf{U}$, always has the same principal directions $\mathbf{N}_I, \mathbf{N}_{II}, \mathbf{N}_{III}$, so that

$$\begin{aligned}\Delta\mathbf{U} = &\left(1 + (\Delta\lambda_I - 1)\frac{\tau}{\Delta t}\right)\mathbf{N}_I\,\mathbf{N}_I + \left(1 + (\Delta\lambda_{II} - 1)\frac{\tau}{\Delta t}\right)\mathbf{N}_{II}\,\mathbf{N}_{II} \\ &+ \left(1 + (\Delta\lambda_{III} - 1)\frac{\tau}{\Delta t}\right)\mathbf{N}_{III}\,\mathbf{N}_{III}\end{aligned}$$

and, hence,

$$\frac{d}{dt}(\Delta\mathbf{U}) = \frac{1}{\Delta t}\left((\Delta\lambda_I - 1)\mathbf{N}_I\,\mathbf{N}_I + (\Delta\lambda_{II} - 1)\mathbf{N}_{II}\,\mathbf{N}_{II} + (\Delta\lambda_{III} - 1)\mathbf{N}_{III}\,\mathbf{N}_{III}\right)$$

and

$$\begin{aligned}\Delta\mathbf{U}^{-1} = &\left(\frac{1}{1 + (\Delta\lambda_I - 1)(\tau/\Delta t)}\right)\mathbf{N}_I\,\mathbf{N}_I + \left(\frac{1}{1 + (\Delta\lambda_{II} - 1)(\tau/\Delta t)}\right)\mathbf{N}_{II}\,\mathbf{N}_{II} \\ &+ \left(\frac{1}{1 + (\Delta\lambda_{III} - 1)(\tau/\Delta t)}\right)\mathbf{N}_{III}\,\mathbf{N}_{III}.\end{aligned}$$

We can, thus, write

$$\begin{aligned}\frac{1}{2}\left(\frac{d}{dt}(\Delta\mathbf{U}) \cdot \Delta\mathbf{U}^{-1} + \Delta\mathbf{U}^{-1} \cdot \frac{d}{dt}(\Delta\mathbf{U})\right) = &\frac{(\Delta\lambda_I - 1)(1/\Delta t)}{1 + (\Delta\lambda_I - 1)(\tau/\Delta t)}\mathbf{N}_I\,\mathbf{N}_I \\ &+ \frac{(\Delta\lambda_{II} - 1)(1/\Delta t)}{1 + (\Delta\lambda_{II} - 1)(\tau/\Delta t)}\mathbf{N}_{II}\,\mathbf{N}_{II} + \frac{(\Delta\lambda_{III} - 1)(1/\Delta t)}{1 + (\Delta\lambda_{III} - 1)(\tau/\Delta t)}\mathbf{N}_{III}\,\mathbf{N}_{III}\end{aligned}$$

and, hence,

$$\int_0^{\Delta t} \frac{1}{2}\left(\frac{d}{dt}(\Delta\mathbf{U})\cdot\Delta\mathbf{U}^{-1}+\Delta\mathbf{U}^{-1}\cdot\frac{d}{dt}(\Delta\mathbf{U})\right)\,d\tau = \ln(\Delta\lambda_I)\mathbf{N}_I\,\mathbf{N}_I + \ln(\Delta\lambda_{II})\mathbf{N}_{II}\,\mathbf{N}_{II} + \ln(\Delta\lambda_{III})\mathbf{N}_{III}\,\mathbf{N}_{III}$$

so that, finally, from Equation 1.4.3–3,

$$\begin{aligned}\Delta\boldsymbol{\varepsilon} &= \ln(\Delta\lambda_I)\mathbf{n}_I\,\mathbf{n}_I + \ln(\Delta\lambda_{II})\mathbf{n}_{II}\,\mathbf{n}_{II} + \ln(\Delta\lambda_{III})\mathbf{n}_{III}\,\mathbf{n}_{III}\\ &= \ln\boldsymbol{\Delta}\mathbf{V}.\end{aligned}$$

Thus, as long as we assume that the stretch at any time during the increment has the same principal directions as the total increment of stretch (on the fixed basis at the start of the increment), the logarithmic definition of incremental strain provides the required integral of the strain rate expressed as the rate of deformation. This assumption amounts to requiring that the components of stretch grow proportionally during the increment: that $\Delta\mathbf{U}(t+\tau) = p\Delta\mathbf{U}(t+\Delta t)$, where p is any scalar that we take to grow monotonically from 0 to 1 during $0 \le \tau \le \Delta t$. This assumption might be questionable if the increments are very large, but it is consistent with the levels of approximation used in the integration of the inelastic constitutive models. We, therefore, have a simple method for calculating the strain increment for use in this type of constitutive model without any additional loss of accuracy compared to what we already accept in the constitutive integration itself.

Reference

- "Conventions," Section 1.2.2 of the Abaqus Analysis User's Guide

1.4.4 THE ADDITIVE STRAIN RATE DECOMPOSITION

Products: Abaqus/Standard Abaqus/Explicit

Many useful materials, such as conventional structural metals, can carry only very small amounts of elastic strain (the elastic modulus is typically two or three orders of magnitude larger than the yield stress). We can take advantage of this behavior to simplify the description of the deformation of such a material. Since the behavior is so common, the assumption that the elastic strains are always small forms the basis of almost all of the inelastic material models provided in Abaqus. This section discusses the description of the deformation for this case.

We begin by assuming that the material has a natural elastic reference state in the sense that, at any time in the deformation, we can imagine isolating the immediate neighborhood of a single point in the material, preventing any further inelastic deformation, removing all external forces from the isolated piece, and allowing the material to unload: the deformation associated with this unloading will then be $(\mathbf{F}^{el})^{-1}$, the reverse of the elastic deformation. The deformation between the original reference state and this elastically unloaded state is then the inelastic deformation, $\mathbf{F}^{pl}$:

$$\mathbf{F}^{pl} = (\mathbf{F}^{el})^{-1} \cdot \mathbf{F}.$$

The total deformation can, thus, be decomposed as

$$\mathbf{F} = \mathbf{F}^{el} \cdot \mathbf{F}^{pl}, \tag{1.4.4–1}$$

from which we can obtain the velocity gradient with respect to position in the current configuration, $\mathbf{L} = \dot{\mathbf{F}} \cdot \mathbf{F}^{-1}$, as

$$\mathbf{L} = \dot{\mathbf{F}}^{el} \cdot (\mathbf{F}^{el})^{-1} + \mathbf{F}^{el} \cdot \dot{\mathbf{F}}^{pl} \cdot (\mathbf{F}^{pl})^{-1} \cdot (\mathbf{F}^{el})^{-1},$$

which we write as

$$\mathbf{L} = \mathbf{L}^{el} + \mathbf{F}^{el} \cdot \bar{\mathbf{L}}^{pl} \cdot (\mathbf{F}^{el})^{-1}, \tag{1.4.4–2}$$

by defining the elastic and plastic velocity gradients, $\mathbf{L}^{el} = \dot{\mathbf{F}}^{el} \cdot (\mathbf{F}^{el})^{-1}$ and $\bar{\mathbf{L}}^{pl} = \dot{\mathbf{F}}^{pl} \cdot (\mathbf{F}^{pl})^{-1}$, by analogy with the definition of the total velocity gradient.

For the materials of concern here, we now assume that the elastic strains, $\boldsymbol{\varepsilon}^{el}$, are very small compared to unity. Using this together with the left polar decomposition of the elastic deformation, we can write

$$\begin{aligned} \mathbf{F}^{el} &= \mathbf{V}^{el} \cdot \mathbf{R}^{el} \\ &= (\mathbf{I} + \vartheta \mathbf{A}) \cdot \mathbf{R}^{el}, \end{aligned}$$

where $|\vartheta| << 1$, $\mathbf{A} = \mathbf{A}^T$, and $sup||\mathbf{A}|| = 1$. We now use this decomposition of $\mathbf{F}^{el}$ in Equation 1.4.4–2 to obtain

$$\begin{aligned}\mathbf{L} &= \mathbf{L}^{el} + (\mathbf{I} + \vartheta\mathbf{A}) \cdot \mathbf{R}^{el} \cdot \bar{\mathbf{L}}^{pl} \cdot \mathbf{R}^{el^T} \cdot (\mathbf{I} - \vartheta\mathbf{A}) \\ &= \mathbf{L}^{el} + \mathbf{R}^{el} \cdot \bar{\mathbf{L}}^{pl} \cdot \mathbf{R}^{el^T} \\ &+ \vartheta\mathbf{A} \cdot \mathbf{R}^{el} \cdot \bar{\mathbf{L}}^{pl} \cdot \mathbf{R}^{el^T} - \vartheta\mathbf{R}^{el} \cdot \bar{\mathbf{L}}^{pl} \cdot \mathbf{R}^{el^T} \cdot \mathbf{A} \\ &- \vartheta^2\mathbf{A} \cdot \mathbf{R}^{el} \cdot \bar{\mathbf{L}}^{pl} \cdot \mathbf{R}^{el^T} \cdot \mathbf{A}.\end{aligned}$$

We now define

$$\begin{aligned}\mathbf{L} &= \mathbf{D} + \mathbf{W}, \\ \mathbf{L}^{el} &= \mathbf{D}^{el} + \mathbf{W}^{el}, \quad \text{and} \\ \bar{\mathbf{L}}^{pl} &= \bar{\mathbf{D}}^{pl} + \bar{\mathbf{W}}^{pl},\end{aligned}$$

where $\mathbf{D}$ and $\mathbf{W}$ denote the symmetric and antisymmetric parts of each velocity gradient, respectively. Using these definitions and neglecting the higher-order term, the velocity gradient can now be expressed as

$$\begin{aligned}\mathbf{L} &= \mathbf{D}^{el} + \mathbf{W}^{el} + \mathbf{R}^{el} \cdot (\bar{\mathbf{D}}^{pl} + \bar{\mathbf{W}}^{pl}) \cdot \mathbf{R}^{el^T} \\ &+ \vartheta\mathbf{A} \cdot \mathbf{R}^{el} \cdot (\bar{\mathbf{D}}^{pl} + \bar{\mathbf{W}}^{pl}) \cdot \mathbf{R}^{el^T} - \vartheta\mathbf{R}^{el} \cdot (\bar{\mathbf{D}}^{pl} + \bar{\mathbf{W}}^{pl}) \cdot \mathbf{R}^{el^T} \cdot \mathbf{A}.\end{aligned}$$

Taking the symmetric part of this expression gives

$$\mathbf{D} = \mathbf{D}^{el} + \mathbf{R}^{el} \cdot \bar{\mathbf{D}}^{pl} \cdot \mathbf{R}^{el^T} + \vartheta\mathbf{A} \cdot \mathbf{R}^{el} \cdot \bar{\mathbf{W}}^{pl} \cdot \mathbf{R}^{el^T} - \vartheta\mathbf{R}^{el} \cdot \bar{\mathbf{W}}^{pl} \cdot \mathbf{R}^{el^T} \cdot \mathbf{A}.$$

We now make the assumption that $\bar{\mathbf{W}}^{pl} = \mathbf{0}$, which holds for isotropy; and the last expression reduces to

$$\mathbf{D} = \mathbf{D}^{el} + \mathbf{D}^{pl}, \tag{1.4.4–3}$$

where we introduce the notation $\mathbf{D}^{pl} = \mathbf{R}^{el} \cdot \bar{\mathbf{D}}^{pl} \cdot \mathbf{R}^{el^T}$. Equation 1.4.4–3 is the classical "additive rate of deformation decomposition" of plasticity theory—see Aravas (1991) for an example. We see that it derives from the general decomposition (Equation 1.4.4–1) when we use the symmetric part of the velocity gradient with respect to current position and when the total elastic strain is always small compared to one. The rate of deformation decomposition is used in this form in almost all the inelastic constitutive models in Abaqus, and it is denoted as $\dot{\varepsilon} = \dot{\varepsilon}^{el} + \dot{\varepsilon}^{pl}$.

Reference

- "Inelastic behavior," Section 23.1.1 of the Abaqus Analysis User's Guide

1.5 Equilibrium, stress, and state storage

- "Equilibrium and virtual work," Section 1.5.1
- "Stress measures," Section 1.5.2
- "Stress rates," Section 1.5.3
- "State storage," Section 1.5.4
- "Energy balance," Section 1.5.5

1.5.1 EQUILIBRIUM AND VIRTUAL WORK

Products: Abaqus/Standard Abaqus/Explicit

Many of the problems to which Abaqus is applied involve finding an approximate (finite element) solution for the displacements, deformations, stresses, forces, and—possibly—other state variables such as temperature in a solid body that is subjected to some history of "loading," where "loading" implies some series of events to which the body's response is sought. The exact solution of such a problem requires that both force and moment equilibrium be maintained at all times over any arbitrary volume of the body. The displacement finite element method is based on approximating this equilibrium requirement by replacing it with a weaker requirement, that equilibrium must be maintained in an average sense over a finite number of divisions of the volume of the body. In this section we develop the exact equilibrium statement and write it in the form of the virtual work statement for later reduction to the approximate form of equilibrium used in a finite element model.

Let V denote a volume occupied by a part of the body in the current configuration, and let S be the surface bounding this volume. (Again, we should emphasize that we are adopting a Lagrangian viewpoint: the volume being considered is a volume of material in the body—specifically, V is the volume of space occupied by this material at the "current" point in time, which is distinct from the Eulerian approach, where we are examining a volume in space and watch material flowing through that volume.) Let the surface traction at any point on S be the force $\mathbf{t}$ per unit of current area, and let the body force at any point within the volume of material under consideration be $\mathbf{f}$ per unit of current volume. Force equilibrium for the volume is then

$$\int_S \mathbf{t}\, dS + \int_V \mathbf{f}\, dV = 0. \tag{1.5.1–1}$$

The "true" or Cauchy stress matrix $\boldsymbol{\sigma}$ at a point of S is defined by

$$\mathbf{t} = \mathbf{n} \cdot \boldsymbol{\sigma}, \tag{1.5.1–2}$$

where $\mathbf{n}$ is the unit outward normal to S at the point. Using this definition, Equation 1.5.1–2 is

$$\int_S \mathbf{n} \cdot \boldsymbol{\sigma}\, dS + \int_V \mathbf{f}\, dV = 0.$$

Gauss's theorem allows us to rewrite a surface integral as a volume integral according to

$$\int_S \mathbf{n} \cdot (\,)\, dS = \int_V \left(\frac{\partial}{\partial \mathbf{x}} \right) \cdot (\,), dV,$$

where $(\,)$ is any continuous function—scalar, vector, or tensor.

Applying the Gauss theorem to the surface integral in the equilibrium equation gives

$$\int_S \mathbf{n} \cdot \boldsymbol{\sigma}\, dS = \int_V \left(\frac{\partial}{\partial \mathbf{x}}\right) \cdot \boldsymbol{\sigma}\, dV.$$

Since the volume is arbitrary, this equation must apply pointwise in the body, thus providing the differential equation of translational equilibrium:

$$\left(\frac{\partial}{\partial \mathbf{x}}\right) \cdot \boldsymbol{\sigma} + \mathbf{f} = 0. \tag{1.5.1–3}$$

These are the three familiar differential equations of force equilibrium. In deriving them we have made no approximation with respect to the magnitude of the deformation or rotation—the equations are an exact statement of equilibrium so long as we are precise about our definitions of surface tractions, body forces, stress (Cauchy stress, defined by Equation 1.5.1–2), volume, and area.

Moment equilibrium is most simply written in the general case by taking moments about the origin:

$$\int_S (\mathbf{x} \times \mathbf{t})\, dS + \int_V (\mathbf{x} \times \mathbf{f})\, dV = 0.$$

Use of the Gauss theorem with this equation then leads to the result that the "true" (Cauchy) stress matrix must be symmetric:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^T, \tag{1.5.1–4}$$

so that at each point there are only six independent components of stress. Conversely, by taking the stress matrix to be symmetric, we automatically satisfy moment equilibrium and, therefore, need only consider translational equilibrium when explicitly writing the equilibrium equations. (The moment equilibrium equation written above assumes that there are no point couples acting on the volume. If there are, the stress matrix does not have the symmetry property of Equation 1.5.1–4. Continuum mechanics models that allow for such point couples have been developed, but they are not relevant to any of the models provided in Abaqus.)

The basis for the development of a displacement-interpolation finite element model is the introduction of some locally based spatial approximation to parts of the solution. To develop such an approximation, we begin by replacing the three equilibrium equations represented by Equation 1.5.1–3 by an equivalent "weak form"—a single scalar equation over the entire body, which is obtained by multiplying the pointwise differential equations by an arbitrary, vector-valued "test function," defined, with suitable continuity, over the entire volume, and integrating. As the test function is quite arbitrary, the differential equilibrium statement in any particular direction at any particular point can always be recovered by choosing the test function to be nonzero only in that direction at that point. For this case of equilibrium with a general stress matrix, this equivalent "weak form" is the virtual work principle. The test function can be imagined to be a "virtual" velocity field, $\delta\mathbf{v}$, which is completely arbitrary except that it must obey any prescribed kinematic constraints and have sufficient continuity: the dot product of this test function with the equilibrium force field then represents the "virtual" work rate.

Taking the dot product of Equation 1.5.1–3 with $\delta\mathbf{v}$ results in a single scalar equation at each material point that is then integrated over the entire body to give

$$\int_V \left[\left(\frac{\partial}{\partial \mathbf{x}}\right)\cdot\boldsymbol{\sigma} + \mathbf{f}\right]\cdot\delta\mathbf{v}\,dV = 0. \tag{1.5.1–5}$$

The chain rule allows us to write

$$\left(\frac{\partial}{\partial \mathbf{x}}\right)\cdot(\boldsymbol{\sigma}\cdot\delta\mathbf{v}) = \left[\left(\frac{\partial}{\partial \mathbf{x}}\right)\cdot\boldsymbol{\sigma}\right]\cdot\delta\mathbf{v} + \boldsymbol{\sigma} : \left(\frac{\partial\delta\mathbf{v}}{\partial \mathbf{x}}\right),$$

so that

$$\begin{aligned}\int_V \left[\left(\frac{\partial}{\partial \mathbf{x}}\right)\cdot\boldsymbol{\sigma}\right]\cdot\delta\mathbf{v}\,dV &= \int_V \left[\left(\frac{\partial}{\partial \mathbf{x}}\right)\cdot(\boldsymbol{\sigma}\cdot\delta\mathbf{v}) - \boldsymbol{\sigma} : \left(\frac{\partial\delta\mathbf{v}}{\partial \mathbf{x}}\right)\right] dV \\ &= \int_S \mathbf{n}\cdot\boldsymbol{\sigma}\cdot\delta\mathbf{v}\,dS - \int_V \boldsymbol{\sigma} : \left(\frac{\partial\delta\mathbf{v}}{\partial \mathbf{x}}\right) dV, \\ &\quad \text{(using the Gauss theorem with the first term)} \\ &= \int_S \mathbf{t}\cdot\delta\mathbf{v}\,dS - \int_V \boldsymbol{\sigma} : \left(\frac{\partial\delta\mathbf{v}}{\partial \mathbf{x}}\right) dV \\ &\quad \text{(using the definition of Cauchy stress with the first term).}\end{aligned}$$

Thus, the virtual work statement, Equation 1.5.1–5, can be written

$$\int_S \mathbf{t}\cdot\delta\mathbf{v}\,dS + \int_V \mathbf{f}\cdot\delta\mathbf{v}\,dV = \int \boldsymbol{\sigma} : \left(\frac{\partial\delta\mathbf{v}}{\partial \mathbf{x}}\right) dV.$$

From the previous section we recognize

$$\frac{\partial\delta\mathbf{v}}{\partial \mathbf{x}} = \delta\mathbf{L}$$

as the virtual velocity gradient in the current configuration. We can decompose the gradient into a symmetric and an antisymmetric part:

$$\delta\mathbf{L} = \delta\mathbf{D} + \delta\mathbf{W},$$

where

$$\delta\mathbf{D} = \mathrm{sym}(\delta\mathbf{L}) = \frac{1}{2}(\delta\mathbf{L} + \delta\mathbf{L}^T)$$

is the virtual strain rate (the virtual rate of deformation) and

$$\delta\mathbf{W} = \mathrm{asym}(\delta\mathbf{L}) = \frac{1}{2}(\delta\mathbf{L} - \delta\mathbf{L}^T)$$

is the virtual rate of spin. With these definitions

$$\boldsymbol{\sigma} : \delta\mathbf{L} = \boldsymbol{\sigma} : \delta\mathbf{D} + \boldsymbol{\sigma} : \delta\mathbf{W}.$$

Since $\boldsymbol{\sigma}$ is symmetric,

$$\boldsymbol{\sigma} : \delta\mathbf{W} = \frac{1}{2}\boldsymbol{\sigma} : \delta\mathbf{L} - \frac{1}{2}\boldsymbol{\sigma} : \delta\mathbf{L}^T = \frac{1}{2}\boldsymbol{\sigma} : \delta\mathbf{L} - \frac{1}{2}\boldsymbol{\sigma} : \delta\mathbf{L} = 0.$$

Finally, we obtain the virtual work equation in the classical form

$$\int_V \boldsymbol{\sigma} : \delta\mathbf{D}\, dV = \int_S \delta\mathbf{v} \cdot \mathbf{t}\, dS + \int_V \delta\mathbf{v} \cdot \mathbf{f}\, dV. \tag{1.5.1–6}$$

Recall that $\mathbf{t}$, $\mathbf{f}$, and $\boldsymbol{\sigma}$ are an equilibrium set,

$$\mathbf{t} = \mathbf{n} \cdot \boldsymbol{\sigma}, \quad \left(\frac{\partial}{\partial \mathbf{x}}\right) \cdot \boldsymbol{\sigma} + \mathbf{f} = 0, \quad \boldsymbol{\sigma} = \boldsymbol{\sigma}^T;$$

$\delta\mathbf{D}$ and $\delta\mathbf{v}$ are compatible,

$$\delta\mathbf{D} = \frac{1}{2}\left(\frac{\partial \delta\mathbf{v}}{\partial \mathbf{x}} + \left[\frac{\partial \delta\mathbf{v}}{\partial \mathbf{x}}\right]^T\right);$$

and $\delta\mathbf{v}$ is compatible with all kinematic constraints. We can show that any two of these three statements (virtual work, equilibrium, and compatibility of the test function $\delta\mathbf{v}$) imply the other: we can thus use the virtual work principle, with a suitable test function, as a statement of equilibrium.

The virtual work statement has a simple physical interpretation: the rate of work done by the external forces subjected to any virtual velocity field is equal to the rate of work done by the equilibrating stresses on the rate of deformation of the same virtual velocity field. The principle of virtual work is the "weak form" of the equilibrium equations and is used as the basic equilibrium statement for the finite element formulation that will be introduced in "Procedures: overview and basic equations," Section 2.1.1. Its advantage in this regard is that it is a statement of equilibrium cast in the form of an integral over the volume of the body: we can introduce approximations by choosing test functions for the virtual velocity field that are not entirely arbitrary, but whose variation is restricted to a finite number of nodal values. This approach provides a stronger mathematical basis for studying the approximation than the alternative of direct discretization of the derivative in the differential equation of equilibrium at a point, which is the typical starting point for a finite difference approach to the same problem.

Reference

- "Conventions," Section 1.2.2 of the Abaqus Analysis User's Guide

1.5.2 STRESS MEASURES

Products: Abaqus/Standard Abaqus/Explicit

The virtual work statement (Equation 1.5.1–6) expresses equilibrium in terms of Cauchy ("true") stress and the conjugate virtual strain rate, the rate of deformation. (Here "conjugate" means work conjugate, in the sense that the product of the stress and the strain rate defines work per current volume.) It is natural to think of stress and strain as conjugate quantities, but so far we only have "true" stress and a wide range of possible strain measures. By defining the concept of conjugacy more precisely, we can define a stress matrix conjugate to any strain matrix that we might choose to use. This exercise has some value, although—as we develop the argument—it is worth remembering that the Cauchy ("true") stress is—from the engineer's viewpoint—probably the only measure of stress of practical interest as an output value from a computer code like Abaqus, because it is a direct measure of the traction being carried per unit area by any internal surface in the body under study. For this reason Abaqus always reports the stress as the Cauchy stress. One of the alternative stress definitions developed in this section (Kirchhoff stress) is relevant to the constitutive development in Chapter 4, "Mechanical Constitutive Theories." The other (second Piola-Kirchhoff stress) is discussed because it is frequently mentioned in standard texts.

It is convenient to think of a solid material as having a natural, elastic, reference state to which it will return upon unloading. For a fully elastic material like rubber, this state will always be the original, unstressed, state. For a material that yields, such as a metal, this reference state will be modified by the inelastic deformation to which the material is subjected. Further, we expect the elasticity of the material to be derivable from a thermodynamic potential written about this reference state so that, for isothermal deformations, there will be a potential function for the elastic strain energy per unit of the natural reference volume. On this basis we formalize the concept of conjugacy by writing the work rate per unit of volume in this elastic reference state as

$$dW^0 = \boldsymbol{\tau} : d\boldsymbol{\varepsilon}, \tag{1.5.2–1}$$

where $\boldsymbol{\varepsilon}$ is a particular choice of strain matrix, derived on the basis of the discussion in "Strain measures," Section 1.4.2, and $\boldsymbol{\tau}$ is now the stress matrix that is work conjugate to $d\boldsymbol{\varepsilon}$. Equation 1.5.2–1 defines a conjugate stress measure for any chosen strain measure.

The internal virtual work rate was expressed in Equation 1.5.1–6 directly in terms of Cauchy stress, and the virtual velocity gradient. This internal virtual work rate may be rewritten as an integral over the natural reference volume:

$$\int_V \boldsymbol{\sigma} : \mathbf{D}\, dV = \int_{V^0} J\boldsymbol{\sigma} : \mathbf{D}\, dV^0,$$

where $J = dV/dV^0$ is the Jacobian of the elastic deformation between the natural reference and the current volume—the ratio of the material's volume in the current and natural configurations. According to the work conjugacy concept just defined, the stress measure defined by

$$\boldsymbol{\tau} = J\boldsymbol{\sigma} \tag{1.5.2–2}$$

is work conjugate to the strain measure whose rate is the rate of deformation

$$\mathbf{D} = \mathrm{sym}\left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right).$$

This measure of stress is called Kirchhoff stress. It is useful in the development of constitutive models at large strain because it is the most directly available stress measure when we wish to think of the strain rate measured by the rate of deformation and are considering a material with an elastic reference state.

The discussion of strain suggested that Green's strain is convenient for the description of problems involving small strains but rotations that are not small, because Green's strain matrix, $\boldsymbol{\varepsilon}^G$, can be computed directly from the deformation gradient $\mathbf{F}$. We now develop the stress measure work conjugate to Green's strain. From "Strain measures," Section 1.4.2, the standard Green's strain matrix was defined with respect to the reference configuration as

$$\boldsymbol{\varepsilon}^G = \frac{1}{2}(\mathbf{F}^T \cdot \mathbf{F} - \mathbf{I}),$$

so the rate of Green's strain is

$$\dot{\boldsymbol{\varepsilon}}^G = \frac{1}{2}(\dot{\mathbf{F}}^T \cdot \mathbf{F} + \mathbf{F}^T \cdot \dot{\mathbf{F}}).$$

From the discussion of the rate of deformation ("Rate of deformation and strain increment," Section 1.4.3) we have $\dot{\mathbf{F}} = \mathbf{L} \cdot \mathbf{F}$ so that

$$\begin{aligned}\dot{\boldsymbol{\varepsilon}}^G &= \frac{1}{2}\mathbf{F}^T \cdot \left(\mathbf{L} + \mathbf{L}^T\right) \cdot \mathbf{F} \\ &= \mathbf{F}^T \cdot \mathbf{D} \cdot \mathbf{F}\end{aligned}$$

and, thus,

$$\mathbf{D} = \mathbf{F}^{-T} \cdot \dot{\boldsymbol{\varepsilon}}^G \cdot \mathbf{F}^{-1},$$

where $\mathbf{F}^{-T}$ means the inverse of the transpose of $\mathbf{F}$.

Since the work rate per unit reference volume is

$$dW^0 = J\boldsymbol{\sigma} : \mathbf{D},$$

it follows that

$$dW^0 = J\boldsymbol{\sigma} : (\mathbf{F}^{-T} \cdot \dot{\boldsymbol{\varepsilon}}^G \cdot \mathbf{F}^{-1}) = J(\mathbf{F}^{-1} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-T}) : \dot{\boldsymbol{\varepsilon}}^G.$$

(This last manipulation is most readily seen by looking at the equation in component form.) Thus,

$$\mathbf{S} = J\mathbf{F}^{-1} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-T}$$

is the stress that is work conjugate to $\dot{\boldsymbol{\varepsilon}}^G$. This stress measure, $\mathbf{S}$, is known as the second Piola-Kirchhoff stress tensor.

For general motions including large strains $\mathbf{S}$ is not readily interpreted physically. But for the important case of large rotations and small strains, the second Piola-Kirchhoff stress is readily interpreted. We can perform the polar decomposition as $\mathbf{F} = \mathbf{R} \cdot \mathbf{U}$, where $\mathbf{R}$ is the rotation of the principal axes of deformation and $\mathbf{U}$ is the right stretch matrix (the stretch written on the reference configuration). If we write the principal stretches in terms of nominal principal strains,

$$\lambda_I = 1 + \varepsilon_I, \ \lambda_{II} = 1 + \varepsilon_{II} \lambda_{III} = 1 + \varepsilon_{III},$$

the right stretch tensor can be written as

$$\mathbf{U} = (1 + \varepsilon_I)\mathbf{N}_I \, \mathbf{N}_I{}^T + (1 + \varepsilon_{II})\mathbf{N}_{II} \, \mathbf{N}_{II}{}^T + (1 + \varepsilon_{III})\mathbf{N}_{III} \, \mathbf{N}_{III}{}^T = \mathbf{I} + \boldsymbol{\varepsilon}.$$

The deformation gradient can be written as

$$\mathbf{F} = \mathbf{R} \cdot (\mathbf{I} + \boldsymbol{\varepsilon}),$$

and the inverse deformation gradient can be approximated by

$$\mathbf{F}^{-1} = (\mathbf{I} + \boldsymbol{\varepsilon})^{-1} \cdot \mathbf{R}^T \approx (\mathbf{I} - \boldsymbol{\varepsilon}) \cdot \mathbf{R}^T,$$

since—for the small-strain case—all entries in $\boldsymbol{\varepsilon}$ are very much smaller than one. In addition,

$$J = \det \mathbf{F} = \det(\mathbf{R}) \det(\mathbf{I} + \boldsymbol{\varepsilon}) = 1 + 0(\boldsymbol{\varepsilon}).$$

Therefore,

$$\mathbf{S} \approx (1 + 0(\boldsymbol{\varepsilon}))(\mathbf{I} - \boldsymbol{\varepsilon}) \cdot \mathbf{R}^T \cdot \boldsymbol{\sigma} \cdot \mathbf{R} \cdot (\mathbf{I} - \boldsymbol{\varepsilon})^T.$$

Neglecting terms of order strain compared to unity (since this is the small-strain approximation), we obtain

$$\mathbf{S} = \mathbf{R}^T \cdot \boldsymbol{\sigma} \cdot \mathbf{R}. \tag{1.5.2–3}$$

This result gives a very simple physical interpretation of the second Piola-Kirchhoff stress for small strains but arbitrarily large rotations: the components of $\mathbf{S}$ are the rotated axis components of $\boldsymbol{\sigma}$. That is, the components of $\mathbf{S}$ are the stress components, associated with directions in the reference configuration. Thus, if we use a rectangular Cartesian basis system, the $(1, 1)$ component of $\mathbf{S}$, S^{11}, is the normal component of force per unit area acting on a surface that was normal to the $\boldsymbol{X}$-axis in the reference configuration, regardless of the current orientation of that surface.

For example, consider a beam whose axis was initially parallel to the $\boldsymbol{X}$-axis. Then, throughout the deformation, S^{11} will always be the axial stress in the beam, no matter how much the beam is rotated or bent (provided the strains remain small compared to unity). Thus, for this case we can think of $\mathbf{S}$ as a "material" or "corotational" stress: the material stress and strain are unique, to the order of the approximation, provided strains remain small.

When the small-strain approximation is no longer valid, it is essential to use appropriate measures of stress and strain. From a constitutive viewpoint we have already introduced the basic idea of the approach we will follow: we identify the natural reference for the material's elastic response and use stress and strain measures that provide a conjugate pairing so that the elastic potential can be readily expressed. Since we are often interested in the rate behavior of a material, and also because we prefer to use Cauchy stress as the most natural expression of the stress at a point, it is attractive to consider the usage of the strain measure whose rate is the rate of deformation. (We have identified this in one dimension as the logarithmic strain.) We then use the Kirchhoff stress, $\boldsymbol{\tau} = J\boldsymbol{\sigma}$, with respect to the reference state for the material's elasticity, as the stress measure for our constitutive definitions; it is this stress measure that is used in forming constitutive models in Abaqus at large strains, as will be seen in Chapter 4, "Mechanical Constitutive Theories."

The work conjugacy principle implies that, for "small strains," all stress measures are indistinguishable, because in this case the strain measures are the same. One interpretation of this is that, if the stress-strain curve for a material is plotted using different stress and strain measures (for example, "true" stress versus log strain, and as nominal stress versus engineering strain) the small-strain approximation is no longer appropriate at strain levels where these two plots differ to any degree considered important to the analysis.

Reference

- "Conventions," Section 1.2.2 of the Abaqus Analysis User's Guide

1.5.3 STRESS RATES

Products: Abaqus/Standard Abaqus/Explicit

Many of the materials we wish to model with Abaqus are history dependent, and it is common for the constitutive equations to appear in rate form. In "Stress measures," Section 1.5.2, it was suggested that an appropriate stress measure for stress-sensitive materials (such as yielding materials) is the Kirchhoff stress. We, therefore, need to define the rate of Kirchhoff stress for use in the constitutive equations. This definition is not simply the material time rate of Kirchhoff stress, because the Kirchhoff stress components are associated with spatial directions in the current configuration (recall that the Kirchhoff stress is $J\boldsymbol{\sigma}$, where J is the volume change from the reference configuration and $\boldsymbol{\sigma}$ is the Cauchy stress, defined by $\mathbf{t} = \boldsymbol{\sigma} \cdot \mathbf{n}$, where $\mathbf{t}$ and $\mathbf{n}$ are vectors in the current configuration).

To illustrate the issue, consider a uniaxial tension specimen under constant axial force P, lying along the x-axis at time t_1 and rotated—with the axial force held constant—to lie along the y-axis at time t_2 (see Figure 1.5.3–1). Write the stress components on the global $(1, 2, 3)$ rectangular Cartesian basis. At time t_1, $\sigma_{11} = P/A$, and all other $\sigma_{ij} = 0$, while at time t_2, $\sigma_{22} = P/A$, and all other $\sigma_{ij} = 0$. Obviously during $t_1 \rightarrow t_2$, $d\sigma_{11} \neq 0$ and $d\sigma_{22} \neq 0$, but equally clearly this rate of change of stress has nothing to do with the constitutive response of the material making up the bar. (A materially based stress, such as the second Piola-Kirchhoff stress, would stay constant during the above rotation, because its components are associated with a material basis.) The problem, then, is that the components of $\boldsymbol{\sigma}$ or $\boldsymbol{\tau}$ are associated with current directions in space and, therefore, $d\boldsymbol{\tau}$ and $d\boldsymbol{\sigma}$ will be nonzero if there is pure rigid body rotation, even though from a constitutive point of view the material is unchanged. Thus, we must divide the increment of $\boldsymbol{\sigma}$ or $\boldsymbol{\tau}$ into two parts—one attributable to rigid body motion only and a remainder that is then, presumably, associated with the rate form of the stress-strain law.

We can derive a simple result for this purpose for any matrix whose components are associated with spatial directions. At some time t imagine attaching to a material point a set of base vectors, $\mathbf{e}_\alpha$, $\alpha = 1, 2, 3$. These vectors cannot stretch but are defined to spin with the same spin as the material. Recall that the spatial gradient of the material particle velocity at a point, $\partial \mathbf{v}/\partial \mathbf{x}$, was decomposed into a rate of deformation and a spin,

$$\mathbf{D} = \frac{1}{2}\left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \left[\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right]^T\right) \quad \text{and} \quad \mathbf{W} = \frac{1}{2}\left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}} - \left[\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right]^T\right).$$

One of the concepts of the motion of the base vectors $\mathbf{e}_\alpha$ in Abaqus is that

$$\dot{\mathbf{e}}_\alpha = \mathbf{W} \cdot \mathbf{e}_\alpha.$$

Another concept of the motion of the base vectors $\mathbf{e}_\alpha$ used in Abaqus is

$$\dot{\mathbf{e}}_\alpha = \boldsymbol{\Omega} \cdot \mathbf{e}_\alpha,$$

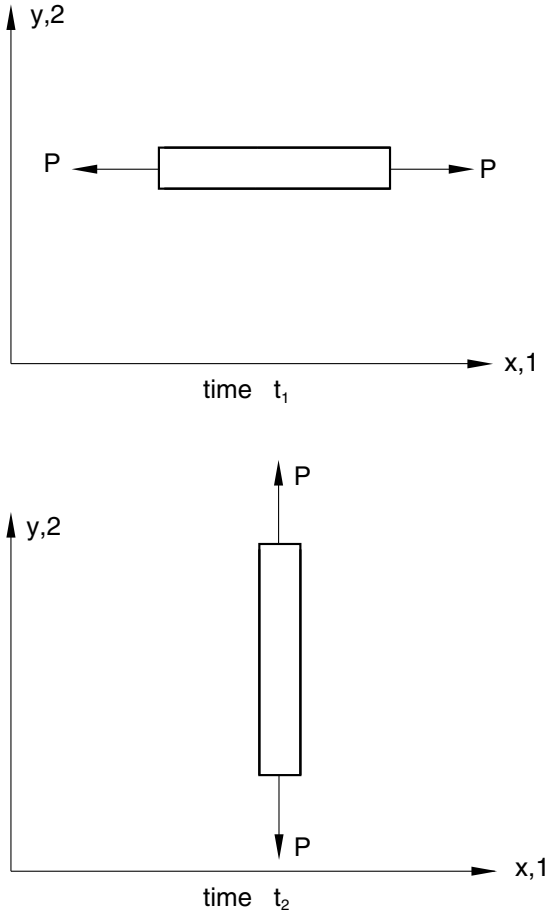


Figure 1.5.3–1 Rotated specimen.

where $\mathbf{\Omega} = \dot{\mathbf{R}} \cdot \mathbf{R}^T$. Here $\mathbf{R}$ is the rigid body rotation in the polar decomposition of the deformation gradient $\mathbf{F}$. The differences between these two concepts are significant only if finite rotation of a material point is accompanied by finite shear.

Now consider any matrix $\mathbf{T}$ based on the current configuration: we can write it in terms of its components in the $\mathbf{e}_\alpha$ directions:

$$\mathbf{T} = T^{\alpha\beta}\mathbf{e}_\alpha\,\mathbf{e}_\beta^T.$$

Taking the time derivative then gives

$$\dot{\mathbf{T}} = \dot{T}^{\alpha\beta}\mathbf{e}_\alpha\,\mathbf{e}_\beta^T + T^{\alpha\beta}\dot{\mathbf{e}}_\alpha\,\mathbf{e}_\beta^T + T^{\alpha\beta}\mathbf{e}_\alpha\,\dot{\mathbf{e}}_\beta^T.$$

The second and third terms are the rate of $\mathbf{T}$ caused by the rigid body spin, so the first term is that part of $\mathbf{T}$ caused by other effects (in the case of stress, the rate associated with the constitutive response), called the corotational rate of $\mathbf{T}$. From the definitions of $\mathbf{e}_\alpha$ as rigid base vectors that can be considered to spin with either $\mathbf{W}$ or $\boldsymbol{\Omega}$, we can write two corotational rates of $\mathbf{T}$ as

$$\mathbf{T}^{\nabla J} = \dot{\mathbf{T}} - \mathbf{W}\cdot\mathbf{T} + \mathbf{T}\cdot\mathbf{W},$$

and

$$\mathbf{T}^{\nabla G} = \dot{\mathbf{T}} - \boldsymbol{\Omega}\cdot\mathbf{T} + \mathbf{T}\cdot\boldsymbol{\Omega},$$

where $\mathbf{T}^{\nabla J}$ and $\mathbf{T}^{\nabla G}$ are called the Jaumann and Green-Naghdi rates, respectively.

We, thus, have the total rate of any matrix associated with spatial directions in the current configuration as the sum of the corotational rate of the matrix and a rate caused purely by the local spin or rigid body rotation. For example, the Jaumann rate of change of Kirchhoff stress can be written as

$$\frac{d^{\nabla J}}{dt}(J\boldsymbol{\sigma}) = \frac{d}{dt}(J\boldsymbol{\sigma}) - J\left(\mathbf{W}\cdot\boldsymbol{\sigma} - \boldsymbol{\sigma}\cdot\mathbf{W}\right).$$

We are assuming that the constitutive theory will define $\frac{d^{\nabla J}}{dt}(J\boldsymbol{\sigma})$, the corotational stress rate per reference volume, in terms of the rate of deformation and past history, so this equation provides a convenient link between that material model and the overall change in "true" (Cauchy) stress (which is the stress measure defined directly from the equilibrium equations). In Chapter 4, "Mechanical Constitutive Theories," where the constitutive models in Abaqus are discussed, "stress rate" per reference volume will mean $\frac{d^{\nabla}}{dt}(J\boldsymbol{\sigma})$, the corotational rate of Kirchhoff stress, which is the stress measure work conjugate to the rate of deformation.

Stress rates used in Abaqus

The objective stress rates used in Abaqus are summarized in Table 1.5.3–1. Objective rates are relevant only for rate form constitutive equations (e.g., elastoplasticity). For hyperelastic materials a total formulation is used; hence, the concept of an objective rate is not relevant for the constitutive law. However, when material orientations are defined, the objective rate governs the evolution of the orientations and the output will be affected.

Table 1.5.3–1 Objective stress rates.

Solver	Element Type	Constitutive Model	Objective Rate
Abaqus/Standard	Solid (Continuum)	All built-in and user-defined materials	Jaumann
	Structural (Shells, Membranes, Beams, Trusses)	All built-in and user-defined materials	Green-Naghdi

Solver	Element Type	Constitutive Model	Objective Rate
Abaqus/Explicit	Solid (Continuum)	All except viscoelastic, brittle cracking, and **VUMAT**	Jaumann
	Solid (Continuum)	Viscoelastic, brittle cracking, and **VUMAT**	Green-Naghdi
	Structural (Shells, Membranes, Beams, Trusses)	All built-in and user-defined materials	Green-Naghdi

Reference

- "Conventions," Section 1.2.2 of the Abaqus Analysis User's Guide

1.5.4 STATE STORAGE

Products: Abaqus/Standard Abaqus/Explicit

Many of the constitutive models in Abaqus require tensors to be stored to define the state at a material calculation point. Such "material state tensors" are stored as their components in a local, orthonormal, system at the material calculation point. The orientation of that system with respect to the global (X, Y, Z) spatial system is stored as a rotation from the global axis system. The purpose of this section is to define the manner in which such tensors are stored and updated.

Three types of local basis are used in Abaqus for material calculations. For isotropic materials in continuum elements the global, spatial, (X, Y, Z) system is used—the material basis is fixed in time. For isotropic materials in structural surface elements (shells and membranes) the local system is defined by the standard Abaqus convention described in "Conventions," Section 1.2.2 of the Abaqus Analysis User's Guide; and for beams and trusses it is defined with the 1-direction along the axis of the member and the 2- and 3-directions in material directions in the cross-section. Thus, with isotropic materials the material basis is always the same as the element basis, although for structural elements the material basis changes with time. For anisotropic materials the material basis must be defined by the user and rotates with the average rigid body spin of the material. In this case the material basis and the element basis are not the same.

We refer to this local material basis at time t as $\mathbf{e}_i^M|_t$, where the superscript M indicates that the basis is associated with material calculations and $_t$ means that the basis is taken at time t. In this section Latin subscripts (like the i above) take the range 1–3, while Greek subscripts will take the range 1–2.

Any tensor associated with the material's state, $\mathbf{a}$, say (such as the stress tensor $\boldsymbol{\sigma}$), is stored in terms of its components along the material basis:

$$\mathbf{a} = a_{ij}\, \mathbf{e}_i^M\, \mathbf{e}_j^M.$$

The increment from time t to time $t + \Delta t$ of local motion at the material calculation point is defined by the incremental deformation gradient,

$$\Delta\mathbf{F} \stackrel{\text{def}}{=} \frac{\partial \mathbf{x}_{t+\Delta t}}{\partial \mathbf{x}_t}.$$

This matrix is calculated from the gradient interpolator of the finite element and the coordinates of the element's nodes at times t and $t + \Delta t$.

The polar decomposition of $\Delta\mathbf{F}$ is

$$\Delta\mathbf{F} = \Delta\mathbf{V} \cdot \Delta\mathbf{R},$$

where $\Delta\mathbf{R}$ is the average rigid body rotation at the material point and $\Delta\mathbf{V}$ is a pure stretch matrix:

$$\Delta\mathbf{V} = \sum_{I=1}^{3} \Delta\lambda_I \mathbf{n}_I \mathbf{n}_I$$

(here $\Delta\lambda_I$ is a principal stretch ratio and $\mathbf{n}_I$ is a principal stretch direction).

During an increment any material state tensor changes according to

$$\mathbf{a}|_{t+\Delta t} = \Delta\mathbf{a}^{\nabla} + \Delta\mathbf{R}\cdot\mathbf{a}|_t\cdot\Delta\mathbf{R}^T,$$

where $\Delta\mathbf{a}^{\nabla}$ is the change in $\mathbf{a}$ caused by constitutive behavior and $\Delta\mathbf{R}$ is the average incremental rigid body rotation of the material. Since material tensors are written in terms of their components in the material basis system, this update is computed as

$$(a_{ij})|_{t+\Delta t} = \Delta a_{ij}^{\nabla} + \Delta R_{ik}(a_{kl})|_t\Delta R_{jl}.$$

It is, therefore, necessary to project $\Delta\mathbf{R}$ onto the material basis systems at the start and end of the increment to define the update of material tensor components:

$$\Delta R_{ij} = \mathbf{e}_i^M|_{t+\Delta t}\cdot\Delta\mathbf{R}\cdot\mathbf{e}_j^M|_t.$$

For isotropic materials the $\mathbf{e}_i^M$ have been chosen for geometric convenience only, so the ΔR_{ij} are quite general.

For anisotropic materials the material basis system, $\mathbf{e}_i^M$, rotates with the average rigid body rotation of the material, $\Delta\mathbf{R}$, and so is updated by

$$\mathbf{e}_i^M|_{t+\Delta t} = \Delta\mathbf{R}\cdot\mathbf{e}_i^M|_t.$$

In this case we see that

$$\Delta R_{ij} = \delta_{ij},$$

and so the update of a material tensor's components simplifies to

$$(a_{ij})|_{t+\Delta t} = \Delta a_{ij}^{\nabla} + (a_{ij})|_t.$$

However, since in this case the material basis system is not the same as the element basis system, $\mathbf{e}_i^E$ (which is chosen for geometric convenience for element calculations), transformations must be done to change basis system. Specifically, at the start of the material calculation routines, the strain increments are rotated from the element basis into the material basis:

$$\Delta\varepsilon_{ij}^M = \Delta\varepsilon_{kl}^E[\mathbf{e}_i^M\cdot\mathbf{e}_k^E][\mathbf{e}_j^M\cdot\mathbf{e}_l^E].$$

Likewise, at the end of the material calculation routines, the stress increments are rotated back to the element basis for integration into the discretized approximation to the equilibrium equations:

$$\Delta\sigma_{ij}^E = \Delta\sigma_{kl}^M[\mathbf{e}_i^E\cdot\mathbf{e}_k^M][\mathbf{e}_j^E\cdot\mathbf{e}_l^M].$$

In addition, the material stiffness matrix,

$$D_{ijkl} \stackrel{\text{def}}{=} \frac{\partial \sigma_{ij}}{\partial \varepsilon_{kl}}$$

must also be rotated from the material basis to the element basis:

$$D^E_{ijkl} = D^M_{mnpq}[\mathbf{e}^E_i \cdot \mathbf{e}^M_m][\mathbf{e}^E_j \cdot \mathbf{e}^M_n][\mathbf{e}^E_k \cdot \mathbf{e}^M_p][\mathbf{e}^E_l \cdot \mathbf{e}^M_q].$$

For a shell or membrane only two-dimensional rotations are required—for example,

$$\Delta\varepsilon^M_{\alpha\beta} = \Delta\varepsilon^E_{\gamma\delta}[\mathbf{e}^M_\alpha \cdot \mathbf{e}^E_\gamma][\mathbf{e}^M_\beta \cdot \mathbf{e}^E_\delta],$$

since $\mathbf{e}^M_3 \equiv \mathbf{e}^E_3$ because both are unit vectors along the normal to the surface.

Reference

- “Conventions,” Section 1.2.2 of the Abaqus Analysis User’s Guide

1.5.5 ENERGY BALANCE

Products: Abaqus/Standard Abaqus/Explicit

The conservation of energy implied by the first law of thermodynamics states that the time rate of change of kinetic energy and internal energy for a fixed body of material is equal to the sum of the rate of work done by the surface and body forces. This can be expressed as

$$\frac{d}{dt}\int_V (\frac{1}{2}\rho \mathbf{v}\cdot\mathbf{v} + \rho U)dV = \int_S \mathbf{v}\cdot\mathbf{t}dS + \int_V \mathbf{f}\cdot\mathbf{v}dV, \tag{1.5.5–1}$$

where

ρ is the current mass density,
$\mathbf{v}$ is the velocity field vector,
U is the internal energy per unit mass,
$\mathbf{t}$ is the surface traction vector,
$\mathbf{f}$ is the body force vector, and
$\mathbf{n}$ is the normal direction vector on boundary S.

Using Gauss' theorem and the identity that $\mathbf{t} = \boldsymbol{\sigma}\cdot\mathbf{n}$ on the boundary S, the first term of the right-hand side of Equation 1.5.5–1 can be rewritten as

$$\begin{aligned}\int_S \mathbf{v}\cdot\mathbf{t}dS &= \int_V (\frac{\partial}{\partial\mathbf{x}})\cdot(\mathbf{v}\cdot\boldsymbol{\sigma})dV \\ &= \int_V [(\frac{\partial}{\partial\mathbf{x}}\cdot\boldsymbol{\sigma})\cdot\mathbf{v} + \frac{\partial\mathbf{v}}{\partial\mathbf{x}} : \boldsymbol{\sigma}]dV \\ &= \int_V [(\frac{\partial}{\partial\mathbf{x}}\cdot\boldsymbol{\sigma})\cdot\mathbf{v} + \dot{\boldsymbol{\varepsilon}} : \boldsymbol{\sigma}]dV,\end{aligned} \tag{1.5.5–2}$$

where we have used the fact that $\boldsymbol{\sigma}$ is symmetric, and we also know (see "Equilibrium and virtual work," Section 1.5.1) that

$$\frac{\partial\mathbf{v}}{\partial\mathbf{x}} : \boldsymbol{\sigma} = \dot{\boldsymbol{\varepsilon}} : \boldsymbol{\sigma},$$

where $\dot{\boldsymbol{\varepsilon}}$ is the strain rate tensor (see "Rate of deformation and strain increment," Section 1.4.3). Substituting Equation 1.5.5–2 into Equation 1.5.5–1 yields

$$\frac{d}{dt}\int_V (\frac{1}{2}\rho \mathbf{v}\cdot\mathbf{v} + \rho U)dV = \int_V [(\frac{\partial}{\partial\mathbf{x}}\cdot\boldsymbol{\sigma} + \mathbf{f})\cdot\mathbf{v} + \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}]dV. \tag{1.5.5–3}$$

From Cauchy's equation of motion we have

$$\frac{\partial}{\partial \mathbf{x}} \cdot \boldsymbol{\sigma} + \mathbf{f} = \rho \frac{d\mathbf{v}}{dt}.$$

Substituting this into Equation 1.5.5–3 gives

$$\begin{aligned} \frac{d}{dt}\int_V (\frac{1}{2}\rho \mathbf{v}\cdot\mathbf{v} + \rho U)dV &= \int_V (\rho \frac{d\mathbf{v}}{dt}\cdot\mathbf{v} + \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}})dV \\ &= \int_V [\frac{d}{dt}(\frac{1}{2}\rho\mathbf{v}\cdot\mathbf{v}) + \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}]dV. \end{aligned}$$

From this we get the energy equation

$$\rho \frac{dU}{dt} = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}.$$

Integrating this equation we find

$$\int_0^t \left(\int_V \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}} dV \right) dt = \int_V \rho U dV + U_0,$$

where U_0 is the energy at time 0. To make the energy balance (Equation 1.5.5–1) more convenient to use, we integrate it in time:

$$\int_V \frac{1}{2}\rho \mathbf{v}\cdot\mathbf{v} dV + \int_V \rho U dV = \int_0^t \dot{E}_{WF} d\tau + \text{constant},$$

or

$$E_K + E_U = \int_0^t \dot{E}_{WF} d\tau + \text{constant},$$

where

$$\dot{E}_{WF} = \int_S \mathbf{v}\cdot\mathbf{t} dS + \int_V \mathbf{f}\cdot\mathbf{v} dV,$$

defined as the rate of work done to the body by external forces and contact friction forces between the contact surfaces. E_K, the kinetic energy, is given by

$$E_K = \int_V \frac{1}{2}\rho \mathbf{v}\cdot\mathbf{v} dV,$$

and E_U is defined as

$$E_U = \int_V \rho U dV = \int_0^t \left(\int_V \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}} dV \right) d\tau - U_0.$$

To track physically distinguishable engineering phenomena more narrowly, we introduce decompositions of the stress, strain, and tractions.

We can split the traction, $\mathbf{t}$, into the surface distributed load, $\mathbf{t}^l$, the solid infinite element radiation traction, $\mathbf{t}^{qb}$, and the frictional traction, $\mathbf{t}^f$. Then $\dot{E}_{WF}$ can be written as

$$\dot{E}_{WF} = \left(\int_S \mathbf{v} \cdot \mathbf{t}^l dS + \int_V \mathbf{f} \cdot \mathbf{v} dV \right) - \left(- \int_S \mathbf{v} \cdot \mathbf{t}^f dS \right) - \left(- \int_S \mathbf{v} \cdot \mathbf{t}^{qb} dS \right) \equiv \dot{E}_W - \dot{E}_F - \dot{E}_{QB},$$

where $\dot{E}_W$ is the rate of work done to the body by external forces, $\dot{E}_{QB}$ is the rate of energy dissipated by the damping effect of solid medium infinite elements, and $\dot{E}_F$ is the rate of energy dissipated by contact friction forces between the contact surfaces. An energy balance for the entire model can then be written as

$$E_U + E_K + E_F - E_W - E_{QB} = \text{constant}. \tag{1.5.5–4}$$

For convenience, the dissipated portions of the internal energy are split off:

$$\begin{aligned} E_U &= \int_0^t \left(\int_V \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}} dV \right) d\tau = \int_0^t \left[\int_V (\boldsymbol{\sigma}^c + \boldsymbol{\sigma}^v) : \dot{\boldsymbol{\varepsilon}} dV \right] d\tau \\ &= \int_0^t \left(\int_V \boldsymbol{\sigma}^c : \dot{\boldsymbol{\varepsilon}} dV \right) d\tau + \int_0^t \left(\int_V \boldsymbol{\sigma}^v : \dot{\boldsymbol{\varepsilon}} dV \right) d\tau \\ &\equiv E_I + E_V, \end{aligned}$$

where $\boldsymbol{\sigma}^c$ is the stress derived from the user-specified constitutive equation, without viscous dissipation effects included; $\boldsymbol{\sigma}^{el}$ is the elastic stress; $\boldsymbol{\sigma}^v$ is the viscous stress (defined for bulk viscosity, material damping, and dashpots); E_V is the energy dissipated by viscous effects; and E_I is the remaining energy, which we continue to call the internal energy. If we introduce the strain decomposition, $\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}}^{el} + \dot{\boldsymbol{\varepsilon}}^{pl} + \dot{\boldsymbol{\varepsilon}}^{cr}$ (where $\dot{\boldsymbol{\varepsilon}}^{el}$, $\dot{\boldsymbol{\varepsilon}}^{pl}$, and $\dot{\boldsymbol{\varepsilon}}^{cr}$ are elastic, plastic, and creep strain rates, respectively), the internal energy, E_I, can be expressed as

$$\begin{aligned} E_I &= \int_0^t \left(\int_V \boldsymbol{\sigma}^c : \dot{\boldsymbol{\varepsilon}} dV \right) d\tau \\ &= \int_0^t \left(\int_V \boldsymbol{\sigma}^c : \dot{\boldsymbol{\varepsilon}}^{el} dV \right) d\tau + \int_0^t \left(\int_V \boldsymbol{\sigma}^c : \dot{\boldsymbol{\varepsilon}}^{pl} dV \right) d\tau + \int_0^t \left(\int_V \boldsymbol{\sigma}^c : \dot{\boldsymbol{\varepsilon}}^{cr} dV \right) d\tau \\ &= E_S + E_P + E_C, \end{aligned} \tag{1.5.5–5}$$

where E_S is the applied elastic strain energy, E_P is the energy dissipated by plasticity, and E_C is the energy dissipated by time-dependent deformation (creep, swelling, and viscoelasticity).

If damage occurs in the material, not all of the applied elastic strain energy is recoverable. At any given time, the stress, $\boldsymbol{\sigma}^c$, can be expressed in terms of the "undamaged" stress, $\boldsymbol{\sigma}^u$, and the continuum damage parameter, d:

$$\boldsymbol{\sigma}^c = (1 - d)\boldsymbol{\sigma}^u.$$

The damage parameter, ***d***, starts at zero (undamaged material) and increases to a maximum value of no more than one (fully damaged material). Hence, we can write

$$E_S = \int_0^t \left(\int_V (1-d)\boldsymbol{\sigma}^u : \dot{\boldsymbol{\varepsilon}}^{el} dV \right) d\tau.$$

We assume that, upon unloading, the damage parameter remains fixed at the value attained at time ***t***. Therefore, the recoverable strain energy is equal to

$$E_E = \int_0^t \left(\int_V (1-d_t)\boldsymbol{\sigma}^u : \dot{\boldsymbol{\varepsilon}}^{el} dV \right) d\tau = \int_0^t \left(\int_V \frac{(1-d_t)}{(1-d)}\boldsymbol{\sigma}^c : \dot{\boldsymbol{\varepsilon}}^{el} dV \right) d\tau,$$

and the energy dissipated through damage is equal to

$$E_D = \int_0^t \left(\int_V (d_t-d)\boldsymbol{\sigma}^u : \dot{\boldsymbol{\varepsilon}}^{el} dV \right) d\tau = \int_0^t \left(\int_V \frac{(d_t-d)}{(1-d)}\boldsymbol{\sigma}^c : \dot{\boldsymbol{\varepsilon}}^{el} dV \right) d\tau.$$

If we define

$$f^u\left(\boldsymbol{\varepsilon}^{el}\right) = \int_0^t \boldsymbol{\sigma}^u : \dot{\boldsymbol{\varepsilon}}^{el} d\tau$$

as the undamaged elastic energy function, we can write

$$E_E = \int_0^t \left(\int_V (1-d_t)\dot{f}^u dV \right) d\tau$$

and

$$E_D = \int_0^t \left(\int_V (d_t-d)\dot{f}^u dV \right) d\tau.$$

Interchanging the integrals yields

$$E_E = \int_V \left(\int_0^t (1-d_t)\dot{f}^u d\tau \right) dV = \int_V \left((1-d_t) f^u \right) dV$$

and

$$E_D = \int_V \left(\int_0^t (d_t-d)\dot{f}^u d\tau \right) dV = \int_V \left[(d_t-d) f^u|_0^t + \int_0^t \dot{d} f^u d\tau \right] dV.$$

The first term in the last expression vanishes, since at time ***t***, $d = d_t$ and at time zero, $f^u = 0$. If we now define the damage strain energy function

$$f^c = (1-d) f^u,$$

then

$$E_D = \int_V \left(\int_0^t \frac{\dot{d}}{1-d} f^c d\tau \right) dV = \int_0^t \int_V \frac{\dot{d}}{1-d} f^c dV d\tau.$$

For a linear elastic energy function

$$f^c = \frac{1}{2} \boldsymbol{\sigma}^u : \boldsymbol{\varepsilon}^{el},$$

and, hence,

$$E_D = \int_0^t \int_V \frac{\dot{d}}{2(1-d)} \boldsymbol{\sigma}^u : \boldsymbol{\varepsilon}^{el} \, dV d\tau.$$

References

- "Abaqus/Standard output variable identifiers," Section 4.2.1 of the Abaqus Analysis User's Guide
- "Abaqus/Explicit output variable identifiers," Section 4.2.2 of the Abaqus Analysis User's Guide

2. Procedures

Overview	2.1
Nonlinear solution methods	2.2
Buckling and postbuckling	2.3
Nonlinear dynamics	2.4
Modal dynamics	2.5
Complex harmonic oscillations	2.6
Steady-state transport analysis	2.7
Analysis of porous media	2.8
Coupled fluid-solid analysis	2.9
Piezoelectric analysis	2.10
Heat transfer	2.11
Coupled thermal-electrical analysis	2.12
Mass diffusion	2.13
Substructuring	2.14
Submodeling	2.15
Fracture mechanics	2.16
Stress linearization	2.17
Design sensitivity analysis	2.18

2.1 Overview

- “Procedures: overview and basic equations,” Section 2.1.1

2.1.1 PROCEDURES: OVERVIEW AND BASIC EQUATIONS

Abaqus is designed as a flexible tool for finite element modeling. An important aspect of this flexibility is the manner in which Abaqus allows the user to step through the history to be analyzed. This is accomplished by defining *analysis procedures.*

A basic concept in Abaqus is the division of the problem history into *steps*. A step is any convenient phase of the history—a thermal transient, a creep hold, a dynamic transient, etc. In its simplest form in Abaqus/Standard, a "step" is just a static analysis of a load change from one magnitude to another.

In each "step" the user chooses a *procedure*, thus defining the type of analysis to be performed during the step: dynamic stress analysis, eigenvalue buckling, transient heat transfer analysis, etc. The procedure choice can be changed from step to step in any meaningful way. Since the state of the model (stresses, strains, temperatures, etc.) is updated throughout all analysis steps, the effects of previous history are always included in the response in each new step. Thus, for example, if natural frequency extraction is performed after a geometrically nonlinear static analysis step, the preload stiffness will be included.

Abaqus/Standard provides both linear and nonlinear response options. The program is truly integrated, so linear analysis is always considered as linear perturbation analysis about the state at the time when the linear analysis procedure is introduced. This linear perturbation approach allows general application of linear analysis techniques in cases where the linear response depends on preloading or the nonlinear response history of the model.

In nonlinear problems the objective is to obtain a convergent solution at a minimum cost. The nonlinear procedures in Abaqus/Standard offer two approaches to this. Direct user control of increment size is one choice, whereby the user specifies the incrementation scheme. Automatic control is the alternate approach: the user defines the step and specifies certain tolerances or error measures. Abaqus/Standard then automatically selects the increments as it develops the response in the step. This approach is usually more efficient, because the user cannot predict the response ahead of time. Automatic control is particularly valuable in cases where the time or load increment varies widely through the step, as is often the case in diffusion type problems (such as creep, heat transfer, and consolidation).

In Abaqus/Explicit the time incrementation is controlled by the stability limit of the central difference operator. The time incrementation scheme is, hence, fully automatic and requires no user intervention. User-specified time incrementation is not available because it would always be nonoptimal.

Abaqus/Standard and Abaqus/Explicit are separate program modules with different data structures; hence, the explicit dynamics procedure cannot be used in the same analysis as any of the procedures in Abaqus/Standard. However, Abaqus provides a capability to import a deformed mesh and associated material state from Abaqus/Explicit into Abaqus/Standard and vice versa. This procedure is described in "Transferring results between Abaqus/Explicit and Abaqus/Standard," Section 9.2.2 of the Abaqus Analysis User's Guide.

In this chapter the basic equations for the most important analysis procedures in Abaqus/Standard and Abaqus/Explicit are described. In some sections specific aspects of an analysis procedure (i.e., damping, cavity radiation, etc.) are discussed.

Basic finite element equations

This section describes the basic equations for standard displacement-based finite element analysis. We begin with the equilibrium statement, written as the virtual work principle, Equation 1.5.1–6:

$$\int_V \boldsymbol{\sigma} : \delta\mathbf{D}\, dV = \int_S \mathbf{t}^T \cdot \delta\mathbf{v}\, dS + \int_V \mathbf{f}^T \cdot \delta\mathbf{v}\, dV.$$

Following the discussion in "Equilibrium and virtual work," Section 1.5.1, the left-hand side of this equation (the internal virtual work rate term) is replaced with the integral over the reference volume of the virtual work rate per reference volume defined by any conjugate pairing of stress and strain:

$$\int_{V^0} \boldsymbol{\tau}^c : \delta\boldsymbol{\varepsilon}\, dV^0 = \int_S \mathbf{t}^T \cdot \delta\mathbf{v}\, dS + \int_V \mathbf{f}^T \cdot \delta\mathbf{v}\, dV, \qquad (2.1.1–1)$$

where $\boldsymbol{\tau}^c$ and $\boldsymbol{\varepsilon}$ are any conjugate pairing of material stress and strain measures. The particular choice of $\boldsymbol{\varepsilon}$ depends on the individual element—see Chapter 3, "Elements."

The finite element interpolator can be written in general as

$$\mathbf{u} = \mathbf{N}_N u^N,$$

where $\mathbf{N}_N$ are interpolation functions that depend on some material coordinate system, u^N are nodal variables, and the summation convention is adopted for the uppercase subscripts and superscripts that indicate nodal variables.

The virtual field, $\delta\mathbf{v}$, must be compatible with all kinematic constraints. Introducing the above interpolation constrains the displacement to have a certain spatial variation, so $\delta\mathbf{v}$ must also have the same spatial form:

$$\delta\mathbf{v} = \mathbf{N}_N \delta v^N.$$

The continuum variational statement Equation 2.1.1–1 is, thus, approximated by a variation over the finite set δv^N.

Now $\delta\boldsymbol{\varepsilon}$ is the virtual rate of material strain associated with $\delta\mathbf{v}$, and because it is a rate form, it must be linear in $\delta\mathbf{v}$. Hence, the interpolation assumption gives

$$\delta\boldsymbol{\varepsilon} = \boldsymbol{\beta}_N \delta v^N,$$

where $\boldsymbol{\beta}_N$ is a matrix that depends, in general, on the current position, $\mathbf{x}$, of the material point being considered. The matrix $\boldsymbol{\beta}_N$ that defines the strain variation from the variations of the kinematic variables is derivable immediately from the interpolation functions once the particular strain measure to be used is defined.

Without loss of generality we can write $\boldsymbol{\beta}_N = \boldsymbol{\beta}_N(\mathbf{x},\ \mathbf{N}_N)$, and—with this notation—the equilibrium equation is approximated as

$$\delta v^N \int_{V^0} \boldsymbol{\beta}_N : \boldsymbol{\tau}^c \, dV^0 = \delta v^N \left[\int_S \mathbf{N}_N^T \cdot \mathbf{t} \, dS + \int_V \mathbf{N}_N^T \cdot \mathbf{f} \, dV \right];$$

since the δv^N are independent variables, we can choose each one to be nonzero and all others zero in turn, to arrive at a system of nonlinear equilibrium equations:

$$\int_{V^0} \boldsymbol{\beta}_N : \boldsymbol{\tau}^c \, dV^0 = \int_S \mathbf{N}_N^T \cdot \mathbf{t} \, dS + \int_V \mathbf{N}_N^T \cdot \mathbf{f} \, dV. \tag{2.1.1–2}$$

This system of equations forms the basis for the (standard) assumed displacement finite element analysis procedure and is of the form

$$F^N(u^M) = 0$$

discussed above. The above equations are valid for static and dynamic analysis if the body force is assumed to contain the inertia contribution. In dynamic analysis, however, the inertia contribution is more commonly considered separately, leading to the equations

$$M^{NM} \ddot{u}^M + F^N(u^M) = 0.$$

For the Newton algorithm (or for the linear perturbation procedure) used in Abaqus/Standard, we need the Jacobian of the finite element equilibrium equations. To develop the Jacobian, we begin by taking the variation of Equation 2.1.1–1, giving

$$\begin{aligned} \int_{V^0} (d\boldsymbol{\tau}^c : \delta\boldsymbol{\varepsilon} + \boldsymbol{\tau}^c : d\delta\boldsymbol{\varepsilon}) dV^0 - \int_S d\mathbf{t}^T \cdot \delta\mathbf{v} \, dS - \int_S \mathbf{t}^T \cdot \delta\mathbf{v} dA_r \frac{1}{A_r} \, dS \\ - \int_V d\mathbf{f}^T \cdot \delta\mathbf{v} dV - \int_V \mathbf{f}^T \cdot \delta\mathbf{v} dJ \frac{1}{J} \, dV = 0, \end{aligned} \tag{2.1.1–3}$$

where $d(\,)$ represents the linear variation of the quantity $(\,)$ with respect to the basic variables (the degrees of freedom of the finite element model). In the above expression $J = |dV/dV^0|$ is the volume change between the reference and the current volume occupied by a piece of the structure and, likewise, $A_r = |dS/dS^0|$ is the surface area ratio between the reference and the current configuration. The Jacobian matrix is obtained by restricting the above variation, allowing variations in the nodal variables, u^N, only. Let such a restricted variation be indicated by $\partial_N = \partial/\partial u^N$. Examining Equation 2.1.1–3 term by term with this in mind, we proceed as follows. The first term contains $d\boldsymbol{\tau}^c$. We now assume that the constitutive theory allows us to write

$$d\boldsymbol{\tau}^c = \mathbf{H} : d\boldsymbol{\varepsilon} + \mathbf{g},$$

where $\mathbf{H}$ and $\mathbf{g}$ are defined in terms of the current state, direction of straining, etc., and on the kinematic assumptions used to form the generalized strains. See Chapter 4, "Mechanical Constitutive Theories," for a detailed discussion of forming $\mathbf{H}$ and $\mathbf{g}$ for the material models currently available in Abaqus. From the choice of generalized strain measure and interpolation function,

$$\partial_N \boldsymbol{\varepsilon} = \frac{\partial \boldsymbol{\varepsilon}}{\partial u^N} = \boldsymbol{\beta}_N.$$

From the above constitutive assumption,

$$\partial_N \boldsymbol{\tau}^c = \mathbf{H} : \boldsymbol{\beta}_N.$$

Now, since $\delta\boldsymbol{\varepsilon}$ is the first variation of $\boldsymbol{\varepsilon}$ with respect to nodal variables,

$$\delta\boldsymbol{\varepsilon} = \partial_M \boldsymbol{\varepsilon} \delta u^M = \boldsymbol{\beta}_M \delta u^M.$$

Thus, the first term in the Jacobian matrix is

$$\int_{V^0} \boldsymbol{\beta}_M : \mathbf{H} : \boldsymbol{\beta}_N dV^0,$$

the usual "small-displacement stiffness matrix," except that, since the strain measure $\boldsymbol{\varepsilon}$ will always be nonlinear in displacement, the $\boldsymbol{\beta}_N$ in this term will be a function of displacement.

The second term in Equation 2.1.1–3 is

$$\int_{V^0} \boldsymbol{\tau}^c : d\delta\boldsymbol{\varepsilon}\, dV^0.$$

This is rewritten as

$$\int_{V^0} \boldsymbol{\tau}^c : \partial_N \delta\boldsymbol{\varepsilon}\, dV^0,$$

which is

$$\int_{V^0} \boldsymbol{\tau}^c : \partial_N \boldsymbol{\beta}_M\, dV^0.$$

This term contributes to the Jacobian and is the "initial stress matrix."

The external load rate terms in Equation 2.1.1–3 are considered next. In general, these load vectors can be written

$$\mathbf{t} = \mathbf{t}(\lambda,\ \mathbf{x}) \text{ and } \mathbf{f} = \mathbf{f}(\lambda,\ \mathbf{x}),$$

where λ represents the externally prescribed loading parameters. Whether the load depends on position or not depends on the particular load type, but common types of loading (pressure, centrifugal load) do depend on position—for example, if $\mathbf{t}$ is caused by pressure on the surface, $\mathbf{t}$ depends on the pressure magnitude, on the direction of the normal to the surface, and on the current surface area: the latter two are functions of the current position of points on the surface. The variation of the load vector with nodal variables can then be written symbolically as

$$\partial_N \mathbf{t} + \mathbf{t}\frac{1}{A_r}\partial_N A_r = \mathbf{Q}_N^S,$$

$$\partial_N \mathbf{f} + \mathbf{f}\frac{1}{J}\partial_N J = \mathbf{Q}_N^V,$$

and then writing

$$\delta\mathbf{v} = \mathbf{N}_M \delta v^M,$$

where $\mathbf{N}_M$ is obtained directly from the interpolation functions, we can write the Jacobian terms pertaining to the last four terms of Equation 2.1.1–3 as

$$-\int_S \mathbf{N}_M^T \cdot \mathbf{Q}_N^S \, dS - \int_V \mathbf{N}_M^T \cdot \mathbf{Q}_N^V \, dV.$$

These are commonly called the "load stiffness matrix." The actual form of the load stiffness is very much dependent on the type of load being considered—see Chapter 3, "Elements," and Hibbitt (1979) for examples.

The complete Jacobian matrix is then

$$K_{MN} = \int_{V^0} \boldsymbol{\beta}_M : \mathbf{H} : \boldsymbol{\beta}_N \, dV^0 + \int_{V^0} \boldsymbol{\tau}^c : \partial_N \boldsymbol{\beta}_M \, dV^0 - \int_S \mathbf{N}_M^T \cdot \mathbf{Q}_N^S \, dS - \int_V \mathbf{N}_M^T \cdot \mathbf{Q}_N^V \, dV. \quad (2.1.1\text{–}4)$$

Thus, Equation 2.1.1–2 and Equation 2.1.1–4 provide the basis for the Newton incremental solution, given specification of the interpolation function and constitutive theories to be used.

Reference

- "Defining an analysis," Section 6.1.2 of the Abaqus Analysis User's Guide

2.2 Nonlinear solution methods

- "Nonlinear solution methods in Abaqus/Standard," Section 2.2.1
- "Quasi-Newton solution technique," Section 2.2.2
- "Direct cyclic algorithm," Section 2.2.3

2.2.1 NONLINEAR SOLUTION METHODS IN Abaqus/Standard

Product: Abaqus/Standard

The finite element models generated in Abaqus are usually nonlinear and can involve from a few to thousands of variables. In terms of these variables the equilibrium equations obtained by discretizing the virtual work equation can be written symbolically as

$$F^N(u^M) = 0, \tag{2.2.1–1}$$

where F^N is the force component conjugate to the Nth variable in the problem and u^M is the value of the Mth variable. The basic problem is to solve Equation 2.2.1–1 for the u^M throughout the history of interest.

Many of the problems to which Abaqus will be applied are history-dependent, so the solution must be developed by a series of "small" increments. Two issues arise: how the discrete equilibrium statement Equation 2.2.1–1 is to be solved at each increment, and how the increment size is chosen.

Abaqus/Standard generally uses Newton's method as a numerical technique for solving the nonlinear equilibrium equations. The motivation for this choice is primarily the convergence rate obtained by using Newton's method compared to the convergence rates exhibited by alternate methods (usually modified Newton or quasi-Newton methods) for the types of nonlinear problems most often studied with Abaqus. The basic formalism of Newton's method is as follows. Assume that, after an iteration i, an approximation u_i^M, to the solution has been obtained. Let c_{i+1}^M be the difference between this solution and the exact solution to the discrete equilibrium equation Equation 2.2.1–1. This means that

$$F^N(u_i^M + c_{i+1}^M) = 0.$$

Expanding the left-hand side of this equation in a Taylor series about the approximate solution u_i^M then gives

$$F^N(u_i^M) + \frac{\partial F^N}{\partial u^P}(u_i^M)c_{i+1}^P + \frac{\partial^2 F^N}{\partial u^P \partial u^Q}(u_i^M)c_{i+1}^P c_{i+1}^Q + \ldots = 0.$$

If u_i^M is a close approximation to the solution, the magnitude of each c_{i+1}^M will be small, and so all but the first two terms above can be neglected giving a linear system of equations:

$$K_i^{NP} c_{i+1}^P = -F_i^N, \tag{2.2.1–2}$$

where

$$K_i^{NP} = \frac{\partial F^N}{\partial u^P}(u_i^M)$$

is the Jacobian matrix and

$$F_i^N = F^N(u_i^M).$$

The next approximation to the solution is then

$$u_{i+1}^M = u_i^M + c_{i+1}^M,$$

and the iteration continues.

Convergence of Newton's method is best measured by ensuring that all entries in F_i^N and all entries in c_{i+1}^N are sufficiently small. Both these criteria are checked by default in an Abaqus/Standard solution. Abaqus/Standard also prints peak values in the force residuals, incremental displacements, and corrections to the incremental displacements at each iteration so that the user can check for these contingencies himself.

Newton's method is usually avoided in large finite element codes, apparently for two reasons. First, the complete Jacobian matrix is sometimes difficult to formulate; and for some problems it can be impossible to obtain this matrix in closed form, so it must be calculated numerically—an expensive (and not always reliable) process. Secondly, the method is expensive per iteration, because the Jacobian must be formed and solved at each iteration. The most commonly used alternative to Newton is the modified Newton method, in which the Jacobian in Equation 2.2.1–2 is recalculated only occasionally (or not at all, as in the initial strain method of simple contained plasticity problems). This method is attractive for mildly nonlinear problems involving softening behavior (such as contained plasticity with monotonic straining) but is not suitable for severely nonlinear cases. (In some cases Abaqus/Standard uses an approximate Newton method if it is either not able to compute the exact Jacobian matrix or if an approximation would result in a quicker total solution time. For example, several of the models in Abaqus/Standard result in a nonsymmetric Jacobian matrix, but the user is allowed to choose a symmetric approximation to the Jacobian on the grounds that the resulting modified Newton method converges quite well and that the extra cost of solving the full nonsymmetric system does not justify the savings in iteration achieved by the quadratic convergence of the full Newton method. In other cases the user is allowed to drop interfield coupling terms in coupled procedures for similar reasons.)

Another alternative is the quasi-Newton method, in which Equation 2.2.1–2 is symbolically rewritten

$$c_{i+1}^P = -[K_i^{NP}]^{-1} F_i^N$$

and the inverse Jacobian is obtained by an iteration process.

There are a wide range of quasi-Newton methods. The more appropriate methods for structural applications appear to be reasonably well behaved in all but the most extremely nonlinear cases—the trade-off is that more iterations are required to converge, compared to Newton. While the savings in forming and solving the Jacobian might seem large, the savings might be offset by the additional arithmetic involved in the residual evaluations (that is, in calculating the F_i), and in the cascading vector transformations associated with the quasi-Newton iterations. Thus, for some practical cases quasi-Newton methods are more economic than full Newton, but in other cases they are more expensive. Abaqus/Standard offers the "BFGS" quasi-Newton method: it is described in "Quasi-Newton solution technique," Section 2.2.2.

When any iterative algorithm is applied to a history-dependent problem, the intermediate, nonconverged solutions obtained during the iteration process are usually not on the actual solution path; thus, the integration of history-dependent variables must be performed completely over the increment at each iteration and not

obtained as the sum of integrations associated with each Newton iteration, c_i. In Abaqus/Standard this is done by assuming that the basic nodal variables, u, vary linearly over the increment, so that

$$u(\tau) = \big(1 - \frac{\tau}{\Delta t}\big)u(t) + \frac{\tau}{\Delta t}u(t + \Delta t),$$

where $0 \leq \tau \leq \Delta t$ represents "time" during the increment. Then, for any history-dependent variable, $g(t)$, we compute

$$g(t + \Delta t) = g(t) + \int_t^{t+\Delta t} \frac{dg}{d\tau}(\tau)\, d\tau$$

at each iteration.

The issue of choosing suitable time steps is a difficult problem to resolve. First of all, the considerations are quite different in static, dynamic, or diffusion cases. It is always necessary to model the response as a function of time to some acceptable level of accuracy. In the case of dynamic or diffusion problems time is a physical dimension for the problem and the time stepping scheme must provide suitable steps to allow accurate modeling in this dimension. Even if the problem is linear, this accuracy requirement imposes restrictions on the choice of time step. In contrast, most static problems have no imposed time scale, and the only criterion involved in time step choice is accuracy in modeling nonlinear effects. In dynamic and diffusion problems it is exceptional to encounter discontinuities in the time history, because inertia or viscous effects provide smoothing in the solution. (One of the exceptions is impact. The technique used in Abaqus/Standard for this is discussed in "Intermittent contact/impact," Section 2.4.2.) However, in static cases sharp discontinuities (such as bifurcations caused by buckling) are common. Softening systems, or unconstrained systems, require special consideration in static cases but are handled naturally in dynamic or diffusion cases. Thus, the considerations upon which time step choice is made are quite different for the three different problem classes.

Abaqus provides both "automatic" time step choice and direct user control for all classes of problems. Direct user control can be useful in cases where the problem behavior is well understood (as might occur when the user is carrying out a series of parameter studies) or in cases where the automatic algorithms do not handle the problem well. However, the automatic schemes in Abaqus are based on extensive experience with a wide range of problems and, therefore, generally provide a reliable approach.

For static problems a number of schemes have been suggested for automatic step control (see, for example, Bergan et al., 1978). Abaqus/Standard uses a scheme based predominantly on the maximum force residuals following each iteration. By comparing consecutive values of these quantities, Abaqus/Standard determines whether convergence is likely in a reasonable number of iterations. If convergence is deemed unlikely, Abaqus/Standard adjusts the load increment; if convergence is deemed likely, Abaqus/Standard continues with the iteration process. In this way excessive iteration is eliminated in cases where convergence is unlikely, and an increment that appears to be converging is not aborted because it needed a few more iterations. One other ingredient in this algorithm is that a minimum increment size is specified, which prevents excessive computation in cases where buckling, limit load, or some modeling error causes the solution to stall. This control is handled internally, with user override if needed. Several other controls are built into the algorithm; for example, it will cut back the increment size if an element inverts due to excessively large geometry changes. These detailed controls are based on empirical testing.

In dynamic analysis when implicit integration is used, the automatic time stepping is based on the concept of half-increment residuals (Hibbitt and Karlsson, 1979). The basic idea is that the time stepping operator defines the velocities and accelerations at the end of the step $(t + \Delta t)$ in terms of displacement at the end of the step and conditions at the beginning of the step. Equilibrium is then established at $(t + \Delta t)$, which ensures an equilibrium solution at the end of each time step and, thus, at the beginning and end of any individual time step. However, these equilibrium solutions do not guarantee equilibrium throughout the step. The time step control is based on measuring the equilibrium error (the force residuals) at some point during the time step, by using the integration operator, together with the solution obtained at $(t + \Delta t)$, to interpolate within the time step. The evaluation is performed at the half step $(t+\Delta t/2)$. If the maximum entry in this residual vector—the maximum "half-increment residual"—is greater than a user-specified tolerance, the time step is considered to be too big and is reduced by an appropriate factor. If the maximum half-increment residual is sufficiently below the user-specified tolerance, the time step can be increased by an appropriate factor for the next increment. Otherwise, the time step is deemed adequate. The algorithm is somewhat more complicated at traumatic events such as impact. Here, the time step can also be adjusted based on the magnitude of residuals in the first or second iteration following such events. Clearly, if these residuals are several orders of magnitude greater than those permitted, convergence is unlikely and the time step is altered immediately to avoid unproductive iteration. These algorithms are discussed in more detail in "Intermittent contact/impact," Section 2.4.2, as well as in the Abaqus Analysis User's Guide. They are products of experience and many numerical experiments and have been shown to be effective in several problem areas of interest.

References

- "Convergence criteria for nonlinear problems," Section 7.2.3 of the Abaqus Analysis User's Guide
- "Time integration accuracy in transient problems," Section 7.2.4 of the Abaqus Analysis User's Guide

2.2.2 QUASI-NEWTON SOLUTION TECHNIQUE

Product: Abaqus/Standard

A major contribution to the computational effort involved in nonlinear analysis is the solution of the nonlinear equations (Equation 2.2.1–1). In most cases Abaqus/Standard uses Newton's method to solve these equations, as described in "Nonlinear solution methods in Abaqus/Standard," Section 2.2.1. The principal advantage of Newton's method is its quadratic convergence rate when the approximation at iteration i is within the "radius of convergence"—that is, when the gradients defined by K_i^{NM} provide an improvement to the solution. The method has two major disadvantages: the Jacobian matrix has to be calculated, and this same matrix has to be solved. The calculation of the Jacobian matrix is a problem because, in many important cases, it is difficult to derive the form of the matrix algebraically. The solution of the Jacobian is a problem because of the computational effort involved: as the problem size increases, the direct solution of the linear equations can dominate the entire computational effort.

There are a number of important nonlinear applications where the Jacobian is symmetric, is fairly well conditioned, and does not change greatly from one iteration to the next. Examples are implicit dynamic time integration with small time increments relative to the periods of the natural vibrations that participate in the response or small-displacement elastic-plastic analysis where the yielding is confined (such as occurs in many practical fracture mechanics applications). In such cases, especially when the problem is large, it can be less expensive to use an alternative to the Newton approach to the solution of the nonlinear equations. The "quasi-Newton" methods are such an approach; and Matthies and Strang (1979) have shown that, for systems of equations with a symmetric Jacobian matrix, the BFGS (Broyden, Fletcher, Goldfarb, Shanno) method can be written in a simple form that is especially effective on the computer and is successful in such applications. This method is implemented in Abaqus/Standard and is described in this section. The user must select this method explicitly: by default, Abaqus/Standard uses the standard Newton method.

The basis of quasi-Newton methods is to obtain a series of improved approximations to the Jacobian matrix, $\widetilde{K}_i^{NM}$, that satisfy the secant condition:

$$F^N(u_i^M) - F^N(u_{i-1}^M) = \widetilde{K}_i^{NM} \left(u_i^M - u_{i-1}^M\right), \tag{2.2.2–1}$$

so that $\widetilde{K}_i^{NM}$ approaches K_i^{NM} as the iterations proceed. Equation 2.2.2–1 is the basic quasi-Newton equation.

For convenience we define the change in the residual from one iteration to the next as

$$\gamma_i^N = F_i^N - F_{i-1}^N,$$

so that Equation 2.2.2–1 can be written

$$\gamma_i^N = \widetilde{K}_i^{NM} c_i^M, \tag{2.2.2–2}$$

where c_i^{-M} is the correction to the solution from the previous iteration, defined in "Nonlinear solution methods in Abaqus/Standard," Section 2.2.1.

Matthies and Strang's implementation of the BFGS method is a computationally inexpensive method of creating a series of approximations to $\left[\widetilde{K}_i^{NM}\right]^{-1}$ that satisfy Equation 2.2.2–1 and retain the symmetry and positive definiteness of $\widetilde{K}_i^{NM}$. They accomplish this by updating $\left[\widetilde{K}_{i-1}^{NM}\right]^{-1}$ to $\left[\widetilde{K}_i^{NM}\right]^{-1}$ using a "product plus increment" form:

$$\left[\widetilde{K}_i^{MN}\right]^{-1} = \left[I^{NL} - \rho_i c_i^N \gamma_i^L\right] \left[\widetilde{K}_{i-1}^{PL}\right]^{-1} \left[I^{PM} - \rho_i \gamma_i^P c_i^M\right] + \rho_i c_i^N c_i^M, \tag{2.2.2–3}$$

where

$$\rho_i = \left(c_i^M \gamma_i^M\right)^{-1}.$$

In the actual implementation of this version of the BFGS method, each $\left[\widetilde{K}_i^{MN}\right]^{-1}$ is not stored: rather, a "kernel" matrix, $\left[\widetilde{K}_I^{MN}\right]^{-1}$, is used (as the decomposition of $\widetilde{K}_I^{MN}$), and the update is accomplished by premultiplication of the kernel matrix by the terms

$$\left[I^{NL} - \rho_j c_j^N \gamma_j^L\right]$$

and postmultiplication of the kernel matrix by the terms

$$\left[I^{PM} - \rho_j \gamma_j^P c_j^M\right]$$

for $j = I+1, I+2, \ldots i$. Because of the form of these terms, the premultiplication and postmultiplication operations result in inner products of vectors and the scaling of vectors by constants: it is this organization that makes the method computationally attractive. However, too many such products ($i - I$ being bigger than, say, 5–10) are not attractive, so usually a new kernel matrix is formed and stored after some iterations. In the Abaqus/Standard implementation the kernel is the actual Jacobian matrix $\partial F^N / \partial u^M$. It is formed whenever a specified number of iterations have been done without obtaining a convergent solution; the default number of iterations is 8. Abaqus/Standard does not reform the kernel unless this value is exceeded, so the same kernel can be used for several increments if the BFGS updates are successful.

In general, the rate of convergence of the quasi-Newton method is slower than the quadratic rate of convergence of Newton's method, though faster than the linear rate of convergence of the modified Newton method.

Reference

- "Convergence criteria for nonlinear problems," Section 7.2.3 of the Abaqus Analysis User's Guide

2.2.3 DIRECT CYCLIC ALGORITHM

Product: Abaqus/Standard

The classical approach in Abaqus/Standard to obtain the stabilized response of an elastic-plastic structure subjected to cyclic loading is to apply the periodic loading cycles repetitively to the unstressed structure until a stabilized state is obtained. At each instant in time it typically involves using Newton's method to solve the nonlinear equilibrium equations

$$R(t) = F(t) - I(t) = 0,$$

where $F(t)$ is the discretized form of a cyclic load that has the characteristic $F(t+T) = F(t)$ at all times t during a load cycle with period T, $I(t)$ represents the internal force vector generated by the stress, and $R(t)$ is the residual vector. As the problem size increases, the solution of these nonlinear equations can dominate the entire computational effort. This approach, therefore, can be quite expensive, since the application of many loading cycles may be required before the stabilized response is obtained. To avoid the considerable numerical expense associated with such a transient analysis, a direct cyclic algorithm is implemented in Abaqus/Standard and is described in this section.

The direct cyclic algorithm uses a modified Newton method in conjunction with a Fourier representation of the solution and the residual vector to obtain the stabilized cyclic response directly. The basic formalism of this method is as follows. We are looking for a displacement function that describes the response of the structure at all times t during a load cycle with period T and has the characteristic $u(t+T) = u(t)$. We use a truncated Fourier series for this purpose:

$$u(t) = u_0 + \sum_{k=1}^{n} \left[u_k^s \sin k\omega t + u_k^c \cos k\omega t \right],$$

where n stands for the number of terms in the Fourier series; $\omega = 2\pi/T$ is the angular frequency; and u_0, u_k^s, and u_k^c are unknown displacement coefficients. Consider c_k the corrections to the coefficients of the displacement solution. The equilibrium equations can then be written as the following linear system of equations:

$$K c_k^{(i+1)} = R_k^{(i)},$$

where K is the elastic stiffness matrix and i stands for the iteration number. Because the elastic stiffness serves as the Jacobian matrix throughout the analysis, the equation system is solved only once. Therefore, the direct cyclic algorithm is likely to be less expensive to use than the full Newton approach to the solution of the nonlinear equations, especially when the problem is large.

We also expand the residual vector in a truncated Fourier series in the same form as the displacement solution:

$$\bar{R}(t) = R_0 + \sum_{k=1}^{n} \left[R_k^s \sin k\omega t + R_k^c \cos k\omega t \right],$$

where each residual vector coefficient R_0, R_k^s, and R_k^c in the Fourier series corresponds to a displacement coefficient. The conversion of $R(t)$ into Fourier terms is done incrementally on an element-by-element basis:

$$R_0 = \frac{1}{T} \int_0^T R(t) dt,$$

$$R_k^s = \frac{2}{T} \int_0^T R(t) \sin k\omega t \, dt,$$

$$R_k^c = \frac{2}{T} \int_0^T R(t) \cos k\omega t \, dt.$$

At the end of each loading cycle, we solve for the corrections to the displacement Fourier coefficients—c_0, c_k^s, and c_k^c. The next displacement coefficients are then

$$u_0^{(i+1)} = u_0^{(i)} + c_0^{(i+1)},$$

$$u_k^{c(i+1)} = u_k^{c(i)} + c_k^{c(i+1)},$$

$$u_k^{s(i+1)} = u_k^{s(i)} + c_k^{s(i+1)}.$$

The updated displacement coefficients are used in the next iteration to obtain displacements at each instant in time. This process is repeated until convergence is obtained. Each pass through the complete load cycle can therefore be thought of as a single iteration of the solution to the nonlinear problem.

Convergence of the direct cyclic method is best measured by ensuring that all the entries in R_k and c_k are sufficiently small. By default, both these criteria are checked in an Abaqus/Standard solution.

There are two accuracy aspects to this algorithm: the number of Fourier terms and the number of iterations to obtain convergence. The number of Fourier terms needed to obtain a solution depends on the time variation of the cyclic load as well as the variation of the structure response. In determining the number of terms, keep in mind that the objective of this kind of analysis is to make low-cycle fatigue life predictions. Hence, the goal is to obtain a good approximation of the plastic strain cycle at each point; local inaccuracies in the stress are less important. More Fourier terms usually provide a more accurate solution but at the expense of additional data storage and computational time. Abaqus/Standard uses an adaptive algorithm to determine the number of Fourier terms during the analysis. Both “automatic” time incrementation and direct user control of the time incrementation can be used in the direct cyclic method.

Since the direct cyclic algorithm uses the modified Newton method, in which a constant elastic stiffness matrix serves as the Jacobian throughout the analysis, interface nonlinearities such as contact and friction are

not taken into account. These nonlinearities are severe and would probably lead to convergence difficulties if they were included in the direct cyclic algorithm.

By default, the periodicity condition, in which the solution of an iteration starts with the solution at the end of the previous iteration, is always imposed from the beginning of an analysis. However, in cases where the periodic solution is not easily found (for example, when the loading is close to causing ratchetting), the state around which the periodic solution is obtained may show considerably more "drift" than would be obtained in a transient analysis. In such cases the user may wish to delay the application of the periodicity condition as an artificial method to reduce this drift. Abaqus/Standard allows the user to choose when to impose the periodicity condition. By delaying the application of the periodicity condition, the user can influence the mean stress and strain level, without affecting the shape of the stress-strain curves or the amount of energy dissipated during the cycle. Therefore, this is rarely necessary since the average stress and strain levels are usually not needed for low-cycle fatigue life predictions.

Reference

- "Direct cyclic analysis," Section 6.2.6 of the Abaqus Analysis User's Guide

2.3 Buckling and postbuckling

- "Eigenvalue buckling prediction," Section 2.3.1
- "Modified Riks algorithm," Section 2.3.2

2.3.1 EIGENVALUE BUCKLING PREDICTION

Product: Abaqus/Standard

Abaqus/Standard contains a capability for estimating elastic buckling by eigenvalue extraction. This estimation is typically useful for "stiff" structures, where the prebuckling response is almost linear. The buckling load estimate is obtained as a multiplier of the pattern of perturbation loads, which are added to a set of base state loads. The base state of the structure may have resulted from any type of response history, including nonlinear effects. It represents the initial state to which the perturbation loads are added. The response to the perturbation loads must be elastic up to the estimated buckling load values for the eigenvalue estimates to be reasonable.

The following physical problem is addressed in eigenvalue buckling analysis: from an arbitrarily achieved base configuration with stresses $\boldsymbol{\sigma}^B$ in equilibrium with surface traction $\mathbf{t}^B$ and body forces $\mathbf{q}^B$, we consider an elastic deformation with "small" displacement gradients under additional surface tractions $\Delta\mathbf{t}$, body forces $\Delta\mathbf{b}$, and boundary displacements $\Delta\mathbf{u}$, where the additional tractions and displacements are applied on mutually complementary parts of the boundary. Such a deformation is a linear perturbation on a predeformed state. A consistent application of the small-displacement gradient assumption to the kinematics and the constitutive equation from an initially stressed state leads to the solution of a linear problem as the response to the additional loading. Since the problem is linear, if $\Delta\boldsymbol{\sigma}$ is the stress response to the loads $\Delta\mathbf{t}$, $\Delta\mathbf{q}$, and $\Delta\mathbf{u}$, then for loads $\lambda\Delta\mathbf{t}$, $\lambda\Delta\mathbf{q}$, and $\lambda\Delta\mathbf{u}$ the stress response will be $\lambda\Delta\boldsymbol{\sigma}$.

Each distinct value of λ corresponds to a linear perturbation of the base state. Among these perturbed states we seek special values of λ that allow for the existence of nontrivial incremental displacement fields with arbitrary magnitudes as valid solutions to the problem. Such nontrivial incremental displacement fields are referred to as buckling modes. In the buckling analysis procedure in Abaqus we do not distinguish between the geometry of the base state and the linearly perturbed configurations. As a result of this assumption we can seek the buckling modes as incremental displacements out of the base state geometry with stresses $\boldsymbol{\sigma}^B + \lambda\Delta\boldsymbol{\sigma}$, applied tractions $\mathbf{t}^B + \lambda\Delta\mathbf{t}$, and applied body forces $\mathbf{q}^B + \lambda\Delta\mathbf{q}$.

The equations of equilibrium for an arbitrarily chosen configuration during buckling, referred to as the current configuration, are written in terms of the nominal stress $\mathbf{P}$ in the base state. If $\mathbf{X}$ represents the position of a material point in the base state, the equilibrium equations can be expressed as

$$\int_{V^B} \mathbf{P} : \frac{\partial \bar{\mathbf{v}}}{\partial \mathbf{X}} \, dV^B = \int_{S^B} \mathbf{p} \cdot \bar{\mathbf{v}} \, dS^B + \int_{V^B} \mathbf{b} \cdot \bar{\mathbf{v}} \, dV^B,$$

where $\bar{\mathbf{v}}$ is an arbitrary virtual velocity field, $\mathbf{p}$ is the nominal traction on the boundary S^B of the body in the base state, $\mathbf{b}$ represents the body force per unit volume in the base state, and V^B is the volume that the body occupies in the base state. The corresponding rate form is given by

$$\int_{V^B} \dot{\mathbf{P}} : \frac{\partial \bar{\mathbf{v}}}{\partial \mathbf{X}} \, dV^B = \int_{S^B} \dot{\mathbf{p}} \cdot \bar{\mathbf{v}} \, dS^B + \int_{V^B} \dot{\mathbf{b}} \cdot \bar{\mathbf{v}} \, dV^B. \qquad (2.3.1\text{–}1)$$

Since we have assumed that the base state and the current state are indistinguishable, we now proceed to express the left-hand side in terms of the rate of Kirchhoff stress $\dot{\boldsymbol{\tau}}$, the velocity gradient $\mathbf{L}$, the virtual velocity

gradient $\bar{\mathbf{L}}$, and the deformation gradient $\mathbf{F}$. Using the relations $\mathbf{P} = \boldsymbol{\tau} \cdot \mathbf{F}^{-T}$, where $\boldsymbol{\tau}$ is the Kirchhoff stress based on the base state as the reference configuration, and $\dot{\bar{\mathbf{F}}} = \bar{\mathbf{L}} \cdot \mathbf{F}$, Equation 2.3.1–1 takes the form

$$\int_{V^B} \dot{\mathbf{P}} : \frac{\partial \bar{\mathbf{v}}}{\partial \mathbf{X}} \, dV^B \;=\; \int_{V^B} [\dot{\boldsymbol{\tau}} : \bar{\mathbf{L}} - (\boldsymbol{\tau} \cdot \mathbf{L}^T) : \bar{\mathbf{L}}] \, dV^B.$$

We now use the relation between the rate of Kirchhoff stress $\dot{\boldsymbol{\tau}}$, the material spin $\boldsymbol{\omega} = \frac{1}{2}(\mathbf{L} - \mathbf{L}^T)$, and the Jaumann rate of Kirchhoff stress $\boldsymbol{\tau}^{\nabla}$ to transform this expression into

$$\int_{V^B} \dot{\mathbf{P}} : \frac{\partial \bar{\mathbf{v}}}{\partial \mathbf{X}} \, dV^B \;=\; \int_{V^B} [\boldsymbol{\tau}^{\nabla} : \bar{\mathbf{D}} \;+\; \boldsymbol{\tau} : (\bar{\mathbf{L}}^T \cdot \mathbf{L} - 2\mathbf{D} \cdot \bar{\mathbf{D}})] \, dV^B.$$

In addition, we can replace the Kirchhoff stress $\boldsymbol{\tau}$ with the Cauchy stress $\boldsymbol{\sigma}$ since it is assumed that the current and reference configurations are indistinguishable.

For the right-hand side of Equation 2.3.1–1 we note that the nominal tractions $\mathbf{p}$ and body forces $\mathbf{b}$ are given by $\mathbf{p} = \mathbf{t} \; dS/dS^B$ and $\mathbf{b} = \mathbf{q} \; dV/dV^B$, where dS and dV are the elements of surface area and volume in the current configuration. For any material point the changes in $\mathbf{t}$ and $\mathbf{q}$ during buckling are completely characterized by the change of the deformation gradient at that point; loosely speaking, the magnitude of the applied forces at any material point is kept fixed, and the change in the applied tractions and body force intensities arises due to the change in geometry. For example, for a pressure load the magnitude of the pressure remains constant but the surface normal changes—a change that is completely characterized by the change in the deformation gradient. Since the ratios of the surface area and volume measures between the reference and current configurations can be viewed as functions of the deformation gradient $\mathbf{F}$ only, it follows that $\mathbf{p}$ and $\mathbf{b}$ at any given material point also change only through their dependence on the deformation gradient; hence, their rates of change can be written as

$$\dot{\mathbf{p}} = \frac{\partial \mathbf{p}}{\partial \mathbf{F}} : \dot{\mathbf{F}} \qquad \text{and} \qquad \dot{\mathbf{b}} = \frac{\partial \mathbf{b}}{\partial \mathbf{F}} : \dot{\mathbf{F}},$$

or when the current and reference configurations are indistinguishable,

$$\dot{\mathbf{p}} = \frac{\partial \mathbf{p}}{\partial \mathbf{F}} : \mathbf{L} \qquad \text{and} \qquad \dot{\mathbf{b}} = \frac{\partial \mathbf{b}}{\partial \mathbf{F}} : \mathbf{L}.$$

Assuming a hypoelastic constitutive law,

$$\boldsymbol{\tau}^{\nabla} = \mathbf{C}(\boldsymbol{\sigma}) : \mathbf{D},$$

where $\mathbf{C}(\boldsymbol{\sigma})$ is a fourth-order tensor that can depend on the current stress, the governing equation for the buckling analysis becomes

$$\int_{V^B} \bar{\mathbf{D}} : \mathbf{C}(\boldsymbol{\sigma}^B) : \mathbf{D}\, dV^B + \int_{V^B} (\boldsymbol{\sigma}^B + \lambda\Delta\boldsymbol{\sigma}) : (\bar{\mathbf{L}}^T \cdot \mathbf{L} - 2\mathbf{D} \cdot \bar{\mathbf{D}})\, dV^B -$$
$$\int_{S^B} \bar{\mathbf{v}} \cdot \left(\frac{\partial \mathbf{p}^B}{\partial \mathbf{F}} : \mathbf{L} \right) dS^B - \int_{S^B} \bar{\mathbf{v}} \cdot \left(\frac{\partial \lambda\Delta\mathbf{p}}{\partial \mathbf{F}} : \mathbf{L} \right) dS^B - \qquad (2.3.1\text{–}2)$$
$$\int_{V^B} \bar{\mathbf{v}} \cdot \left(\frac{\partial \mathbf{b}^B}{\partial \mathbf{F}} : \mathbf{L} \right) dV^B - \int_{V^B} \bar{\mathbf{v}} \cdot \left(\frac{\partial \lambda\Delta\mathbf{b}}{\partial \mathbf{F}} : \mathbf{L} \right) dV^B = \mathbf{0},$$

where $\mathbf{p}^B$ and $\lambda\Delta\mathbf{p}$ are the nominal tractions generated during buckling corresponding to the base state tractions $\mathbf{t}^B$ and the linear perturbation tractions $\Delta\mathbf{t}$, respectively; similar definitions apply for the nominal body force terms. The constitutive relation can represent elasticity, hypoelasticity, and hyperelasticity; rate effects and plasticity are ignored. The effective moduli are evaluated for the value of the stress and deformation in the base state.

To derive the finite element discretization for the expression above, we introduce the interpolated velocity field

$$\mathbf{v} = v^N \mathbf{N}^N(\mathbf{X}),$$

where $\mathbf{X}$ represents the position in the base state. Using the standard finite element approach, the governing equations for buckling then take the form of the standard eigenvalue problem:

$$(K_0^{NM} + \lambda K_\Delta^{NM}) v^M = 0,$$

where K_0^{NM} is the base state stiffness and K_Δ^{NM} is the differential stiffness. The base state stiffness is the sum of the hypoelastic tangent stiffness, the initial stress stiffness, and the load stiffness:

$$K_0^{NM} = \int_{V^B} \left(\frac{\partial \mathbf{N}^N}{\partial \mathbf{x}} \right)_{sym} : \mathbf{C}(\boldsymbol{\sigma}^B) : \left(\frac{\partial \mathbf{N}^M}{\partial \mathbf{x}} \right)_{sym} dV^B +$$
$$\int_{V^B} \boldsymbol{\sigma}^B : \left[\left(\frac{\partial \mathbf{N}^N}{\partial \mathbf{x}} \right)^T \cdot \frac{\partial \mathbf{N}^M}{\partial \mathbf{x}} - 2 \left(\frac{\partial \mathbf{N}^M}{\partial \mathbf{x}} \right)_{sym} \cdot \left(\frac{\partial \mathbf{N}^N}{\partial \mathbf{x}} \right)_{sym} \right] dV^B -$$
$$\int_{S^B} \mathbf{N}^N \cdot \frac{\partial \mathbf{p}^B}{\partial u^M}\, dS^B - \int_{V^B} \mathbf{N}^N \cdot \frac{\partial \mathbf{b}^B}{\partial u^M}\, dV^B,$$

where $\partial \mathbf{p}^B / \partial u^M$ and $\partial \mathbf{b}^B / \partial u^M$ are the derivatives of the nominal surface tractions and body forces with respect to the nodal displacements. Since we do not distinguish between the current configuration and the reference configuration, the partial derivatives appearing in the load stiffness terms are all evaluated at $u^M = 0$ corresponding to $\mathbf{F} = \mathbf{I}$. For example, the load stiffness term for the surface tractions appearing in Equation 2.3.1–2,

$$\bar{\mathbf{v}} \cdot \left(\frac{\partial \mathbf{p}^B}{\partial \mathbf{F}} : \mathbf{L} \right),$$

transforms into the finite element expression

$$\mathbf{N}^N \cdot \left(\frac{\partial \mathbf{p}^B}{\partial \mathbf{F}} : \frac{\partial \mathbf{N}^M}{\partial \mathbf{X}} \right) v^M = \mathbf{N}^N \cdot \frac{\partial \mathbf{p}^B}{\partial u^M} v^M,$$

with

$$\mathbf{F} = \mathbf{I} + \frac{\partial \mathbf{N}^K}{\partial \mathbf{X}} u^K.$$

The differential stiffness consists of the sum of the initial stress stiffness due to the perturbation stresses and the load stiffness due to the perturbation loads:

$$K_\Delta^{NM} = \int_{V^B} \Delta\boldsymbol{\sigma} : \left[\left(\frac{\partial \mathbf{N}^N}{\partial \mathbf{x}} \right)^T \cdot \frac{\partial \mathbf{N}^M}{\partial \mathbf{x}} - 2 \left(\frac{\partial \mathbf{N}^M}{\partial \mathbf{x}} \right)_{sym} \cdot \left(\frac{\partial \mathbf{N}^N}{\partial \mathbf{x}} \right)_{sym} \right] dV^B - \int_{S^B} \mathbf{N}^N \cdot \frac{\partial \Delta \mathbf{p}}{\partial u^M} dS^B - \int_{V^B} \mathbf{N}^N \cdot \frac{\partial \Delta \mathbf{b}}{\partial u^M} dV^B.$$

The contribution in this expression that is derived from the stress is symmetric; however, the contribution derived from the applied loads (the load stiffness) is symmetric only if the applied loading is conservative—that is, if the loads can be derived from an energy potential. If the load stiffness is nonsymmetric, the contribution will be symmetrized since Abaqus can solve eigenvalue problems only with symmetric matrices.

If the generalized nodal "loads" resulting from both applied forces $\mathbf{t}^B$ and $\mathbf{q}^B$ as well as prescribed displacements $\mathbf{u}^B$ are denoted by P^N and those due to $\Delta\mathbf{t}$, $\Delta\mathbf{q}$, and $\Delta\mathbf{u}$ are denoted by Q^N, the eigenvalues λ_i represent the multipliers that provide the estimated generalized buckling load as $P^N + \lambda_i Q^N$, while the corresponding eigenvectors v_i^N give the associated buckling modes. Although in most analyses the lowest mode is the only one of interest, Abaqus is able to extract several modes simultaneously. It is also worth noting that the common case of an antisymmetric buckling mode on a symmetric base state and buckling load is easily done with Abaqus.

If the tangent stiffness is predicted poorly by $K_0^{NM} + \lambda K_\Delta^{NM}$ (that is, the structure is not "stiff" in the sense that the response is nonlinear prior to buckling), a nonlinear analysis using the Riks method is required to obtain a reliable estimate for the load carrying capacity of the structure.

Reference

- "Eigenvalue buckling prediction," Section 6.2.3 of the Abaqus Analysis User's Guide

2.3.2 MODIFIED RIKS ALGORITHM

Product: Abaqus/Standard

It is often necessary to obtain nonlinear static equilibrium solutions for unstable problems, where the load-displacement response can exhibit the type of behavior sketched in Figure 2.3.2–1—that is, during periods of the response, the load and/or the displacement may decrease as the solution evolves. The modified Riks method is an algorithm that allows effective solution of such cases.

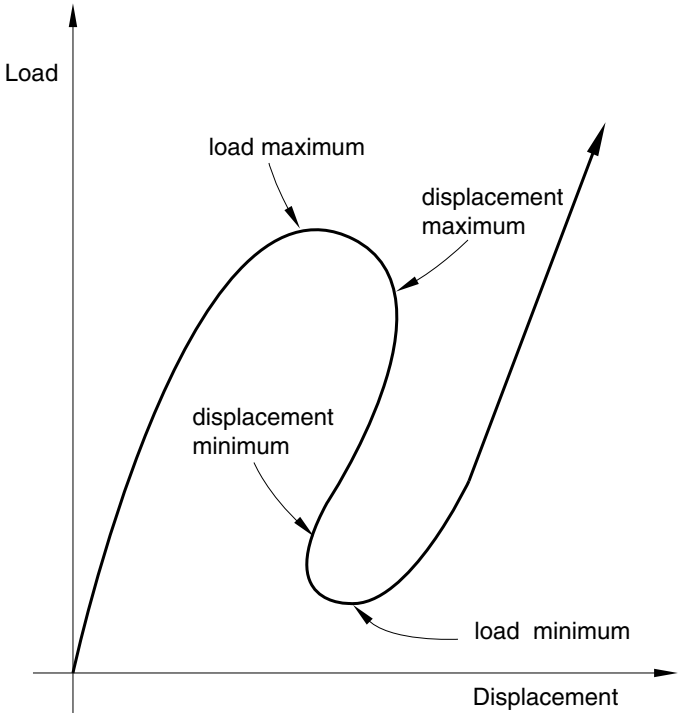


Figure 2.3.2–1 Typical unstable static response.

It is assumed that the loading is proportional—that is, that all load magnitudes vary with a single scalar parameter. In addition, we assume that the response is reasonably smooth—that sudden bifurcations do not occur. Several methods have been proposed and applied to such problems. Of these, the most successful seems to be the modified Riks method—see, for example, Crisfield (1981), Ramm (1981), and Powell and Simons (1981)—and a version of this method has been implemented in Abaqus. The essence of the method is that the solution is viewed as the discovery of a single equilibrium path in a space defined by the nodal variables and the loading parameter. Development of the solution requires that we traverse this path as far as required. The basic algorithm remains the Newton method; therefore, at any time there will be a finite radius

of convergence. Further, many of the materials (and possibly loadings) of interest will have path-dependent response. For these reasons, it is essential to limit the increment size. In the modified Riks algorithm, as it is implemented in Abaqus, the increment size is limited by moving a given distance (determined by the standard, convergence rate-dependent, automatic incrementation algorithm for static case in Abaqus/Standard) along the tangent line to the current solution point and then searching for equilibrium in the plane that passes through the point thus obtained and that is orthogonal to the same tangent line. Here the geometry referred to is the space of displacements, rotations, and the load parameter mentioned above.

Basic variable definitions

Let P^N ($N = 1, 2, \ldots$ = the degrees of freedom of the model) be the loading pattern, as defined with one or more of the loading options in Abaqus. Let λ be the load magnitude parameter, so at any time the actual load state is λP^N, and let u^N be the displacements at that time.

The solution space is scaled to make the dimensions approximately the same magnitude on each axis. In Abaqus this is done by measuring the maximum absolute value of all displacement variables, $\overline{u}$, in the initial (linear) iteration. We also define $\overline{P} = (P^N\ P^N)^{\frac{1}{2}}$. The scaled space is then spanned by

$$\text{load} = \lambda \tilde{P}^N,\ \tilde{P}^N = P^N/\overline{P},$$

$$\text{displacements} = \tilde{u}^N = (u^N/\overline{u});$$

and the solution path is then the continuous set of equilibrium points described by the vector $(\tilde{u}^N;\ \lambda)$ in this scaled space. All components of this vector will be of order unity. The algorithm is shown in Figure 2.3.2–2 and is described below.

Suppose the solution has been developed to the point $A^o = (\tilde{u}_o^N;\ \lambda_o)$. The tangent stiffness, K_o^{NM}, is formed, and we solve

$$K_o^{NM} v_o^M = P^N.$$

The increment size (A^o to A^1 in Figure 2.3.2–2) is chosen from a specified path length, Δl, in the solution space, so that

$$\Delta\lambda_o^2(\tilde{v}_o^N;\ 1) : (\tilde{v}_o^N;\ 1) = \Delta l^2,$$

and, hence,

$$\Delta\lambda_o = \frac{\pm\Delta l}{\left(\tilde{v}_o^N \tilde{v}_o^N + 1\right)^{\frac{1}{2}}}$$

(here $\tilde{v}_o^N$ is v_o^N scaled by $\overline{u}$). The value Δl is initially suggested by the user and is adjusted by the Abaqus/Standard automatic load incrementation algorithm for static problems, based on the convergence rate. The sign of $\Delta\lambda_o$—the direction of response along the tangent line—is chosen so that the dot product of $\Delta\lambda_o(\tilde{v}_o^N;\ 1)$ on the solution to the previous increment, $(\Delta\tilde{u}_{-1}^N;\ \Delta\lambda_{-1})$, is positive:

$$\Delta\lambda_o(\tilde{v}_o^N;\ 1) : (\Delta\tilde{u}_{-1}^N;\ \Delta\lambda_{-1}) > 0,$$

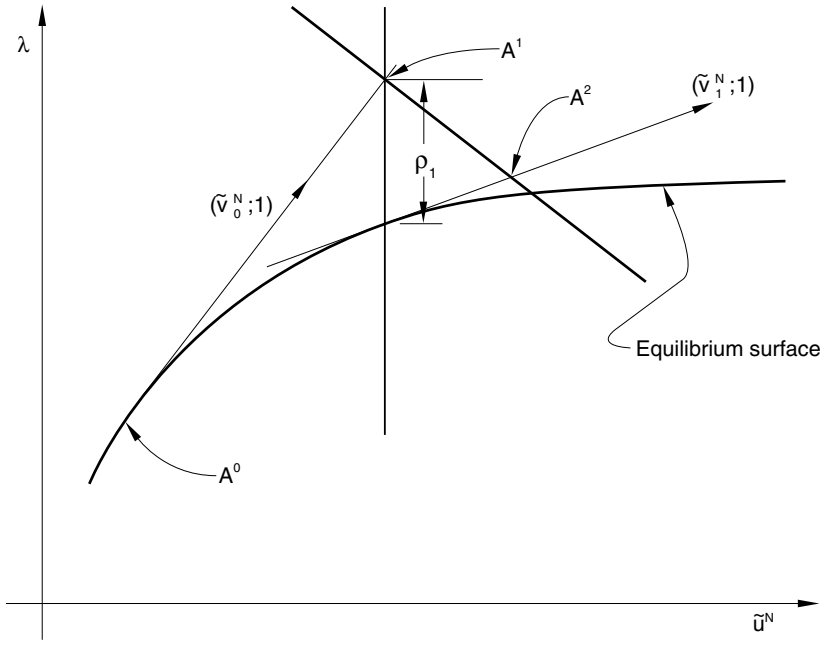


Figure 2.3.2–2 Modified Riks algorithm.

that is

$$\Delta\lambda_o(\tilde{v}_o^N \Delta\tilde{u}_{-1}^N + \Delta\lambda_{-1}) > 0.$$

It is possible that in some cases, where the response shows very high curvature in the $(\tilde{u}^N; \lambda)$ space, this criterion will cause the wrong sign to be chosen—see, for example, Figure 2.3.2–3. The wrong sign is rarely chosen in practical cases, unless the increment size is too large or the solution bifurcates sharply. To check for such cases is computationally expensive: one approach would be for the solution to be found at $\lambda_o - \varepsilon\Delta\lambda_{-1},\ 0 < \varepsilon << 1$, so that we obtain a vector that gives a close approximation of the directed tangent at A^o. Because the case is so rare, such a check is not included, and the simple dot product given above is used alone to determine the sign of $\Delta\lambda^o$. Thus, we have now found the point $A^1\ (\tilde{u}_o^N + \Delta\lambda_o\tilde{v}_o^N; \lambda_o + \Delta\lambda_o)$ in Figure 2.3.2–2. The solution is now corrected onto the equilibrium path in the plane passing through A^1 and orthogonal to $(\tilde{v}_o^N; 1)$, by the following iterative algorithm.

Initialize: $\Delta\lambda_i = \Delta\lambda_o,\ \Delta u_i^N = \Delta\lambda_0 v_o^N$

For $i =$ iteration ($i = 1,\ 2,\ 3,$ etc.):

a. Form $I^N,\ K^{NM}$, the internal (stress) forces at the nodes,

$$I^N = \int_V \boldsymbol{\beta}^N : \boldsymbol{\sigma} dV, \text{ and } K^{NM} = \frac{\partial I^N}{\partial u^M},$$

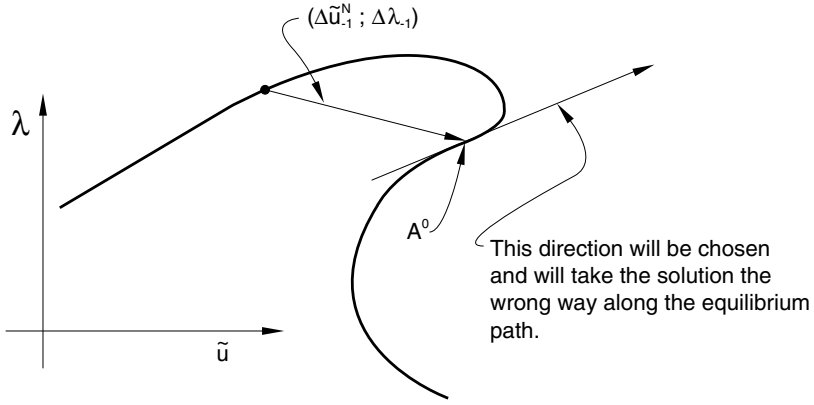


Figure 2.3.2–3 Example of incorrect choice of sign for $\Delta\lambda$.

at the state $(u_o^N + \Delta u_i^N;\ \lambda_o + \Delta\lambda_i)$—that is, at A^i in Figure 2.3.2–2.

b. Check equilibrium:

$$R_i^N = (\lambda_o + \Delta\lambda_i)P^N - I^N.$$

If all the entries in R_i^N are sufficiently small, the increment has converged. If not, we proceed.

c. Solve:

$$K^{NM}\{v_i^M; c_i^M\} = \{P^N; R_i^N\}.$$

That is, we solve simultaneously with two load vectors, P^N and R^N, and obtain two displacement vectors, v_i^N and c_i^N.

d. Now scale the vector $(\tilde{v}_i^N; 1)$, and add it to $(\tilde{c}_i^N; \rho_i)$ where $\rho_i = R_i^N P^N / \overline{P}^2$ is the projection of the scaled residuals onto $\tilde{P}^N$ so that we move from A^i to A^{i+1} in the plane orthogonal to $(\tilde{v}_o^N; 1)$—see Figure 2.3.2–2. This gives the equation

$$\{(0; -\rho_i) + (\tilde{c}_i^N; \rho_i) + \mu(\tilde{v}_i^N; 1)\} : (\tilde{v}_o^N; 1) = 0,$$

which simplifies to give

$$\mu = -\frac{\tilde{c}_i^N \tilde{v}_o^N}{\tilde{v}_i^N \tilde{v}_o^N + 1},$$

and the solution point is now A^i:

$$(u_o^N + \Delta u_i^N + c_i^N + \mu v_i^N; \lambda_o + \Delta\lambda_i + \mu).$$

e. Update for the next iteration,

$$\Delta u_{i+1}^N = \Delta u_i^N + c_i^N + \mu v_i^N$$
$$\Delta \lambda_{i+1} = \Delta \lambda_i + \mu$$
$$i = i + 1,$$

and return to (a) above for the next iteration.

The implementation in Abaqus/Standard includes the additional update after each iteration:

$$v_o^N = v_i^N.$$

This causes the equilibrium search to be orthogonal to the last tangent, rather than to the tangent at the beginning of the increment. The main motivation for this additional modification comes from the use of the method in plasticity problems, where the first iteration of each increment uses the elastic material stiffness to establish the direction of straining and so provides a stiffness that is not representative of the tangent to the equilibrium path if active plasticity is occurring.

The total path length traversed is determined by the load magnitudes supplied by the user on the loading options; while the number of increments is determined by the user-specified time increment data, assisted by Abaqus/Standard's automatic incrementation scheme if that is chosen.

Reference

- "Unstable collapse and postbuckling analysis," Section 6.2.4 of the Abaqus Analysis User's Guide

2.4 Nonlinear dynamics

- "Implicit dynamic analysis," Section 2.4.1
- "Intermittent contact/impact," Section 2.4.2
- "Subspace dynamics," Section 2.4.3
- "Equivalent rigid body dynamic motion," Section 2.4.4
- "Explicit dynamic analysis," Section 2.4.5

2.4.1 IMPLICIT DYNAMIC ANALYSIS

Product: Abaqus/Standard

Abaqus offers dynamic analysis options for both linear and nonlinear problems. In the case of purely linear systems, methods based on the eigenmodes of the system are almost always chosen because they can provide insight into the structure's behavior that is not otherwise available and because they are usually significantly more cost-effective than the direct integration methods that are usually used for nonlinear problems. The linear dynamic analysis methods provided in Abaqus/Standard are discussed in "Modal dynamics," Section 2.5. For mildly nonlinear dynamic analysis problems the "modal projection method" is provided. The basis of that method is to use the eigenmodes of the linear system (extracted from an eigenfrequency analysis) as a set of global Ritz functions—a set of global interpolation functions, in the terminology of the finite element method—whose amplitudes define the response. Abaqus/Standard provides direct time integration using the explicit, central difference operator for this option. For any more severely nonlinear case the dynamic response is obtained by direct time integration of all of the degrees of freedom of the finite element model. The methods provided for this type of analysis are discussed in this section.

The choice of operator used to integrate the equations of motion in a dynamic analysis is influenced by many factors. Abaqus/Standard is designed to analyze structural components, by which we mean that the overall dynamic response of a structure is sought, in contrast to wave propagation solutions associated with relatively local response in continua. Belytschko (1976) labels these "inertial problems" and classifies them by stating that "wave effects such as focusing, reflection, and diffraction are not important." Structural problems are considered "inertial" because the response time sought is long compared to the time required for waves to traverse the structure.

Dynamic integration operators are broadly characterized as implicit or explicit. Explicit schemes, as used in Abaqus/Explicit, obtain values for dynamic quantities at $t + \Delta t$ based entirely on available values at time t. The central difference operator, which is the most commonly used explicit operator for stress analysis applications, is only conditionally stable, the stability limit being approximately equal to the time for an elastic wave to cross the smallest element dimension in the model. Implicit schemes remove this upper bound on time step size by solving for dynamic quantities at time $t + \Delta t$ based not only on values at t, but also on these same quantities at $t + \Delta t$. But because they are implicit, nonlinear equations must be solved. In structural problems implicit integration schemes usually give acceptable solutions with time steps typically one or two orders of magnitude larger than the stability limit of simple explicit schemes, but the response prediction will deteriorate as the time step size, Δt, increases relative to the period, T, of typical modes of response. See, for example, Hilber, Hughes, and Taylor (1977) and Hilber and Hughes (1978) for a discussion of such errors. Three factors should be considered when selecting the maximum allowable time step size: the rate of variation of the applied loading, the complexity of the nonlinear damping and stiffness properties, and the typical period of vibration of the structure. In general, a maximum increment versus period ratio $\Delta t/T < 1/10$ is a good rule of thumb for obtaining reliable results. Thus, the relative economy of the two techniques of integration depends on the stability limit of the explicit scheme, the ease with which the nonlinear equations can be solved for the implicit operator, the relative size of time increments that can provide acceptable accuracy with the implicit scheme compared to the stability limit of the explicit scheme, and the size of the model.

In Abaqus/Standard the time step for implicit integration can be chosen automatically on the basis of the "half-increment residual," a concept introduced in Hibbitt and Karlsson (1979). By monitoring the values of equilibrium residuals at $t + \Delta t/2$ once the solution at $t + \Delta t$ has been obtained, the accuracy of the solution can be assessed and the time step adjusted appropriately.

To discuss the dynamic procedures further, we write out the d'Alembert force in the overall equilibrium equation, Equation 2.1.1–1. The body force at a point, $\mathbf{f}$, can be written as an externally prescribed body force, $\mathbf{F}$, and a d'Alembert force:

$$\mathbf{f} = \mathbf{F} - \rho\ddot{\mathbf{u}},$$

where ρ is the current density of the material at this point and $\mathbf{u}$ is the displacement of the point. The body force term in the virtual work equation is

$$\int_V \mathbf{f} \cdot \delta\mathbf{v}\, dV = \int_V \mathbf{F} \cdot \delta\mathbf{v}\, dV - \int_V \rho\, \ddot{\mathbf{u}} \cdot \delta\mathbf{v}\, dV.$$

The d'Alembert term can be written more conveniently in terms of the reference volume and reference density, ρ_0, as

$$\int_{V_0} \rho_0 \ddot{\mathbf{u}} \cdot \delta\mathbf{v}\, dV_0,$$

where $\ddot{\mathbf{u}}$ is the acceleration field. When implicit integration is used, the equilibrium equations are written at the end of a time step (at time $t + \Delta t$), and $\ddot{\mathbf{u}}$ is calculated from the time integration operator. The interpolator approximates the displacement at a point as

$$\mathbf{u} = \mathbf{N}^N u^N,$$

so that

$$\ddot{\mathbf{u}} = \mathbf{N}^N \ddot{u}^N,$$

provided that $\mathbf{N}^N$ are not displacement dependent. This is the case for most of the elements in Abaqus; the form taken for the d'Alembert force terms in those instances where it is not true (the Hermite cubic beams, B23 and B33) is discussed in detail in Chapter 3, "Elements." With this interpolation assumption, the d'Alembert force term is

$$-\left(\int_{V_0} \rho_0 \mathbf{N}^N \cdot \mathbf{N}^M\, dV_0\right) \ddot{u}^M;$$

that is, the consistent mass matrix times the accelerations of the nodal variables. The finite element approximation to equilibrium, Equation 2.1.1–2, is

$$M^{NM}\ddot{u}^M + I^N - P^N = 0, \tag{2.4.1–1}$$

where

$$M^{NM} = \int_{V_0} \rho_0 \mathbf{N}^N \cdot \mathbf{N}^M \, dV_0$$

is the consistent mass matrix,

$$I^N = \int_{V_0} \boldsymbol{\beta}^N : \boldsymbol{\sigma} \, dV_0$$

is the internal force vector, and

$$P^N = \int_S \mathbf{N}^N \cdot \mathbf{t} \, dS + \int_V \mathbf{N}^N \cdot \mathbf{F} \, dV$$

is the external force vector. In this context the terms "matrix" and "vector" refer to matrices and vectors in the space of the nodal variables u^N.

The definition of the mass matrix introduced above is the "consistent" mass: the mass matrix obtained by consistent use of the interpolation. The first-order elements in Abaqus all use "lumped" mass, where the mass matrix is a diagonal matrix. The lumped matrix is obtained by adding each row of the consistent matrix onto the diagonal. For these first-order elements the lumped mass matrix gives more accurate results in numerical experiments that calculate the natural frequencies of simple models.

Implicit operators available in Abaqus/Standard for time integration of the dynamic problem include the operator defined by Hilber, Hughes, and Taylor (1977) and the backward Euler operator. The Hilber-Hughes-Taylor operator is a generalization of the Newmark operator with controllable numerical damping—the damping being most valuable in the automatic time stepping scheme, because the slight high-frequency numerical noise inevitably introduced when the time step is changed is removed rapidly by a small amount of numerical damping. The operator replaces the actual equilibrium equation (Equation 2.4.1–1) with a balance of d'Alembert forces at the end of the time step and a weighted average of the static forces at the beginning and end of the time step:

$$M^{NM}\ddot{u}^M|_{t+\Delta t} + (1+\alpha)(I^N|_{t+\Delta t} - P^N|_{t+\Delta t}) - \alpha(I^N|_t - P^N|_t) + L^N|_{t+\Delta t} = 0, \qquad (2.4.1\text{–}2)$$

where $L^N|_{t+\Delta t}$ is the sum of all Lagrange multiplier forces associated with degree of freedom N. The operator definition is completed by the Newmark formulae for displacement and velocity integration:

$$u|_{t+\Delta t} = u|_t + \Delta t\, \dot{u}|_t + \Delta t^2\big((\frac{1}{2} - \beta)\ddot{u}|_t + \beta\ddot{u}|_{t+\Delta t}\big) \qquad (2.4.1\text{–}3)$$

and

$$\dot{u}|_{t+\Delta t} = \dot{u}|_t + \Delta t\big((1-\gamma)\ddot{u}|_t + \gamma\ddot{u}|_{t+\Delta t}\big), \qquad (2.4.1\text{–}4)$$

with

$$\beta = \frac{1}{4}(1-\alpha)^2, \quad \gamma = \frac{1}{2} - \alpha \quad \text{and} \quad -\frac{1}{2} \le \alpha \le 0.$$

Hilber, Hughes, and Taylor (1977) present cogent arguments for the use of Equation 2.4.1–2–Equation 2.4.1–4 for integrating structural dynamics problems. The main appeal of the operator is its controllable numerical damping and the form this damping takes, slowly growing at low frequencies, with more rapid growth in damping at high frequencies. Control over the amount of numerical damping is provided by the parameter α: with $\alpha = 0$, there is no damping and the operator is the trapezoidal rule (Newmark, $\beta = 1/4$); while with $\alpha = -1/3$, significant damping is available. This operator is used primarily because the slight numerical damping it offers is needed in the automatic time stepping scheme. Each time step change introduces some slight noise or "ringing" into the solution; a little numerical damping ($\alpha = -0.05$ seems a good choice) quickly removes this high-frequency noise without having any significant effect on the meaningful, lower frequency response. An energy content output is available and should be printed to monitor the overall energy balance. This has been done in dynamic examples in the Abaqus Example Problems Guide and shows that the numerical dissipation is always quite small (less than 1% of the total energy).

The integration of rotations during implicit dynamic calculations is done to preserve accuracy in cases where the rotary inertia is different in different directions in a body. For this purpose the accelerations are integrated in the body axis system, so that Newmark's formula gives the change in velocity as

$$\dot{\phi}^\alpha|_{t+\Delta t} = \dot{\phi}^\alpha|_t + \Delta t \left[\gamma \ddot{\phi}^\alpha|_{t+\Delta t} + (1-\gamma)\ddot{\phi}^\alpha|_t\right],$$

where $\dot{\phi}^\alpha$ is the angular velocity of the node and $\ddot{\phi}^\alpha$ is its angular acceleration, both taken in the current direction of the α body axis at time t or at time $t + \Delta t$; Δt is the time increment.

In the global system this is

$$\dot{\boldsymbol{\phi}}|_{t+\Delta t} = \Delta t \gamma \ddot{\boldsymbol{\phi}}|_{t+\Delta t} + [\mathbf{e}^\alpha|_{t+\Delta t}\, \mathbf{e}^\alpha|_t] \cdot \left[\dot{\boldsymbol{\phi}}|_t + \Delta t(1-\gamma)\ddot{\boldsymbol{\phi}}|_t\right],$$

where $\mathbf{e}^\alpha = \mathbf{e}^\alpha(\boldsymbol{\phi})$ are orthonormal base vectors defining the body axis system. Since these are orthonormal vectors, this can be rewritten as

$$\dot{\boldsymbol{\phi}}|_{t+\Delta t} = \Delta t \gamma \ddot{\boldsymbol{\phi}}|_{t+\Delta t} + \Delta \mathbf{C} \cdot \left[\dot{\boldsymbol{\phi}}|_t + \Delta t(1-\gamma)\ddot{\boldsymbol{\phi}}|_t\right], \tag{2.4.1–5}$$

where $\Delta \mathbf{C}$ is the incremental rotation matrix. Let $\Delta \boldsymbol{\theta}$ be the increment in rotation from time t to $t + \Delta t$. Then $\Delta \mathbf{C}$ is

$$\Delta \mathbf{C} = \exp[\Delta \hat{\boldsymbol{\theta}}],$$

where $\Delta \hat{\boldsymbol{\theta}}$ is the skew-symmetric matrix with axial vector $\Delta \boldsymbol{\theta}$. See "Rotation variables," Section 1.3.1, for a discussion of rotation variables, rotation matrices, and the exponential mapping of a skew-symmetric matrix.

Newmark's formula for the time integration of the rotation increment in the body axis system gives

$$\Delta\theta^\alpha = \Delta t\dot{\phi}^\alpha|_t + \Delta t^2\left[\left(\frac{1}{2}-\beta\right)\ddot{\phi}^\alpha|_t + \beta\ddot{\phi}^\alpha|_{t+\Delta t}\right].$$

Since $\Delta\mathbf{C}\cdot\Delta\boldsymbol{\theta} = \exp[\Delta\hat{\boldsymbol{\theta}}]\cdot\Delta\boldsymbol{\theta} = \Delta\boldsymbol{\theta}$, the components $\Delta\theta^\alpha$ are the same relative to the body axes at time $\boldsymbol{t}$ or $t+\Delta t$. In the global system this is

$$\Delta\boldsymbol{\theta} = \Delta t^2\beta\ddot{\boldsymbol{\phi}}|_{t+\Delta t} + \Delta\mathbf{C}\cdot\left[\Delta t\dot{\boldsymbol{\phi}}|_t + \Delta t^2\left(\frac{1}{2}-\beta\right)\ddot{\boldsymbol{\phi}}|_t\right].$$

Solving for the unknown velocity and acceleration at time $t+\Delta t$, the velocity update equation is

$$\dot{\boldsymbol{\phi}}|_{t+\Delta t} = \frac{\gamma}{\Delta t\beta}\Delta\boldsymbol{\theta} + \Delta\mathbf{C}\cdot\left[\left(1-\frac{\gamma}{\beta}\right)\dot{\boldsymbol{\phi}}|_t + \Delta t\left(1-\frac{\gamma}{2\beta}\right)\ddot{\boldsymbol{\phi}}|_t\right],$$

and the acceleration update equation becomes

$$\ddot{\boldsymbol{\phi}}|_{t+\Delta t} = \frac{1}{\Delta t\gamma}\dot{\boldsymbol{\phi}}|_{t+\Delta t} + \Delta\mathbf{C}\cdot\left[\left(1-\frac{1}{\gamma}\right)\ddot{\boldsymbol{\phi}}|_t - \frac{1}{\Delta t\gamma}\dot{\boldsymbol{\phi}}|_t\right].$$

The backward Euler operator solves the equilibrium equation (Equation 2.4.1–1) at the end of the time step and updates the displacement and velocity using

$$u|_{t+\Delta t} = u|_t + \Delta t\,\dot{u}|_{t+\Delta t}$$

and

$$\dot{u}|_{t+\Delta t} = \dot{u}|_t + \Delta t\ddot{u}|_{t+\Delta t}.$$

The automatic time stepping for dynamic problems in Abaqus is based on the half-increment residual first proposed in Hibbitt and Karlsson (1979). The concept is quite appealing intuitively. Satisfaction of Equation 2.4.1–1 at the end of each time step (or actually of the Hilber-Hughes-Taylor form, Equation 2.4.1–2) ensures equilibrium in the discrete sense of the finite element model at these points in time but does not say anything about the quality of equilibrium at intermediate time points. The idea of the half-increment residual is to calculate the equilibrium residual error (the left-hand side of Equation 2.4.1–1) at some intermediate time point (chosen as $t+\Delta t/2$) and to assess the error in the dynamic response prediction by the magnitude of that error.

The half-increment residual is based on the assumption that the accelerations vary linearly over the time interval (this is the basis of Newmark's formulae) so that, for any nodal displacement or rotation component, $\boldsymbol{u}$:

$$\ddot{u}|_\tau = (1-\tau)\ddot{u}|_t + \tau\ddot{u}|_{t+\Delta t}, \quad 0\le\tau\le 1.$$

Having already solved for the state at $t+\Delta t$, this equation, together with Newmark's formulae now written for the time interval from $\boldsymbol{t}$ to $t+\Delta t$, requires that

$$\Delta u|_\tau = \tau^3 \Delta u|_{t+\Delta t} + \tau(1-\tau^2)\Delta t \dot{u}|_t + \tau^2(1-\tau)\frac{\Delta t^2}{2}\ddot{u}|_t,$$
$$\dot{u}|_\tau = \frac{\gamma}{\beta\tau\Delta t}\Delta u|_\tau + \left(1-\frac{\gamma}{\beta}\right)\dot{u}|_t + \left(1-\frac{\gamma}{2\beta}\right)\tau\Delta t\ddot{u}|_t, \quad \text{and}$$
$$\ddot{u}|_\tau = \frac{1}{\beta\tau^2\Delta t^2}\Delta u|_\tau - \frac{1}{\beta\tau\Delta t}\dot{u}|_t + \left(1-\frac{1}{2\beta}\right)\ddot{u}|_t,$$

where

$$\Delta u|_{t+\Delta t} = u|_{t+\Delta t} - u|_t$$

is the increment in displacement obtained for the time step, Δt.

With these equations it is possible to evaluate the equilibrium residual at any time within the step. Presumably, if the solution is accurate, this residual will be small compared to significant forces in the problem. The residual at the end of the time step is

$$R^N|_{t+\Delta t} = M^{NM}\ddot{u}^M|_{t+\Delta t} + (1+\alpha)(I^N|_{t+\Delta t} - P^N|_{t+\Delta t}) - \alpha(I^N|_t - P^N|_t) + L^N|_{t+\Delta t}.$$

The residual at the start of the step is

$$R^N|_t = M^{NM}\ddot{u}^M|_t + (1+\alpha)(I^N|_t - P^N|_t) - \alpha(I^N|_{t^-} - P^N|_{t^-}) + L^N|_t,$$

where t^- is the time at the start of the previous time step during normal time stepping analysis or $t^- = t$ if this is the first increment after an initial acceleration or impact calculation. Then the residual at $t + \Delta t/2$ is defined as

$$R^N|_{t+\Delta t/2} \stackrel{\text{def}}{=} M^{NM}\ddot{u}^M|_{t+\Delta t/2} + (1+\alpha)(I^N|_{t+\Delta t/2} - P^N|_{t+\Delta t/2}) - \frac{1}{2}\alpha(I^N|_t - P^N|_t + I^N|_{t^-} - P^N|_{t^-}) + L^N|_{t+\Delta t/2},$$

where the $I^N|_{t+\Delta t/2}$, etc. are computed for conditions at time $(t + \Delta t/2)$ and

$$L^N|_{t+\Delta t/2} \stackrel{\text{def}}{=} \frac{1}{2}(L^N|_{t+\Delta t} + L^N|_t).$$

The "half-increment residual," $R|_{t+\Delta t/2}$, is defined as the magnitude of the largest entry in $R^N|_{t+\Delta t/2}$ and provides a measure of accuracy of the time-stepping solution.

The motivation behind the calculation of the half-increment residual is to provide a measure of the accuracy of the solution for a given time step. Numerical tests show that it provides a sensitive accuracy check on dynamic solutions and suggest that, if $\boldsymbol{P}$ is a typical magnitude of real forces in an undamped elastic system (for which the high-frequency response must be modeled reasonably accurately), then

if $R|_{t+\Delta t/2} \approx 0.1P$ consistently, the time stepping solution has high accuracy;
if $R|_{t+\Delta t/2} \approx P$ consistently, the time stepping solution has moderately good accuracy;
if $R|_{t+\Delta t/2} \approx 10P$ consistently, the time stepping solution is rather coarse.

Problems with large amounts of natural dissipation of energy, such as elastic-plastic systems, are usually less sensitive to time step choice than purely elastic problems, because the energy that appears in higher frequency modes is quickly dissipated. In such cases maximum half-increment residuals in the range of 1–10 times typical forces indicate quite acceptable accuracy for most studies, and even values of 10–100 times typical forces can give useful results for primary effects, such as overall deformation. Thus, the method can offer relatively cost-effective solutions for highly dissipative systems for which we require only moderately accurate prediction of the overall response.

The half-increment residual is the basis of the adaptive time incrementation scheme. If the half-increment residual is small, it indicates that the accuracy of the solution is high and the time step can be increased safely; conversely, if the half-increment calculation shows the solution is coarse, the time step used in the solution should be reduced. The algorithm is described in detail in "Time integration accuracy in transient problems," Section 7.2.4 of the Abaqus Analysis User's Guide. The algorithm is purely empirical, but experience shows that it works quite well (Hibbitt and Karlsson, 1979), most especially in initially excited problems with high dissipation, such as impulsively loaded problems (or problems with short duration forcing) with extensive plasticity. In these cases the scheme is economic because the time step naturally increases as the solution progresses and the high-frequency response is dissipated.

The above observations are based on using the Hilber-Hughes-Taylor operator with $\alpha = -0.05$. The slight numerical damping that the operator introduces removes the noise that inevitably enters the solution when the time step is changed. Even in quite lengthy problems the overall energy totals, computed on the basis of physical mechanisms in the model, balance well. The energy balance calculation is useful in assessing a solution—for example, the extent to which energy is dissipated by plasticity can be measured—and it is recommended that the user request occasional printout of the energy balance calculation when doing any analysis with Abaqus.

Reference

- "Implicit dynamic analysis using direct integration," Section 6.3.2 of the Abaqus Analysis User's Guide

2.4.2 INTERMITTENT CONTACT/IMPACT

Product: Abaqus/Standard

Many dynamic problems involve intermittent contact, often with severe impact occurring when the structures hit. The contact algorithms in Abaqus/Standard are designed to handle such cases. An example is the pipe whip problem, in which the fluid escaping from a ruptured pipe causes pipe motion and possible impact with restraints along the pipe. Depending on the geometry of the pipe and the position of the postulated break, severe impact can occur since the pipe may acquire high velocity before hitting a motionless restraint.

Impact conditions with hard contact are modeled in Abaqus/Standard with the assumption that, at the time of impact, the two impacting surfaces instantaneously acquire the same velocity in the direction of the impact. This "fully plastic" impact concept is essential to any discrete model, the accuracy of the physical representation of local effects being dependent on the spatial model adopted. It should be emphasized that the plastic impact assumption is local: it is assumed that, at impact, energy is dissipated by some mechanism that is not modeled and whose spatial and temporal scale is infinitesimal compared to the discrete model. In the case of a beam model of a pipe hitting some other structure, this would presumably be the local plastic deformation of the pipe wall where it impacts the other structure. The points that acquire the same velocity sometimes separate just after the impact: part of the dynamic contact algorithm is designed to handle this case.

The local plastic impact concept is quite different from the simple "assumed coefficient of restitution" methods sometimes used in scoping analyses in place of detailed analysis of the history of contact forces. The method discussed here is specifically designed to allow as much—or as little—of this local detail to be obtained, as required. The usual time integration procedures in Abaqus maintain the energy balance in terms of the energy mechanisms of the discrete model, so the instantaneous jumps that occur in the velocity and acceleration at impact imply that some other system of equations must govern the solution during impact. Thus, we view the impact events as separate from the usual time stepping and develop a set of impulse equations that allow the propagation of the solution over these instants of time and, hence, provide initial conditions from which the normal time stepping solution can continue to the next such event.

To develop the governing equations for impact, assume that, at some time t_0, a part I of the surface of two bodies, A and B, comes into contact. Denote the velocities and accelerations of corresponding parts just before impact as

$$\dot{u}_{AI}^{-},\ \dot{u}_{BI}^{-},\ \ddot{u}_{AI}^{-},\ \ddot{u}_{BI}^{-}$$

and

$$\dot{u}_{AI}^{+}, \dot{u}_{BI}^{+}, \ddot{u}_{AI}^{+}, \ddot{u}_{BI}^{+}$$

just after impact.

The fully plastic impact assumption requires that corresponding points acquire the same velocity and acceleration in the direction of impact immediately after impact so that, at time t_0^+:

$$\begin{aligned} \mathbf{n}\cdot\dot{\mathbf{u}}_{AI} &= \mathbf{n}\cdot\dot{\mathbf{u}}_{BI}, \\ \mathbf{n}\cdot\ddot{\mathbf{u}}_{AI} &= \mathbf{n}\cdot\ddot{\mathbf{u}}_{BI}. \end{aligned} \tag{2.4.2–1}$$

Here $\mathbf{n}$ is the current normal to the interface surface $\boldsymbol{I}$.

Writing $\Delta\dot{\mathbf{u}} = \dot{\mathbf{u}}^{+} - \dot{\mathbf{u}}^{-}$ to describe the velocity "jump" at t_0 at a point, we have

$$\mathbf{n}\cdot\left(\dot{\mathbf{u}}^{-}_{AI} + \Delta\dot{\mathbf{u}}_{AI}\right) = \mathbf{n}\cdot\left(\dot{\mathbf{u}}^{-}_{BI} + \Delta\dot{\mathbf{u}}_{BI}\right). \tag{2.4.2–2}$$

The component of the force per unit area between the bodies across $\boldsymbol{I}$ in the normal direction $\mathbf{n}$, N^I, must satisfy

$$\begin{aligned} N^I_A &= -N^I_B \quad \text{for} \quad t \geq t_0 \\ N^I_A &= N^I_B = 0 \quad \text{for} \quad t < t_0. \end{aligned} \tag{2.4.2–3}$$

Since finite velocity jumps are occurring at the time of impact in infinitesimal time (compared to the time scale of the simulation), during the infinitesimal interval t_0^- to t_0^+, N^I dominates all other forces in the system except for the d'Alembert forces. This simplifies the virtual work equation during t_0^- to t_0^+ to

$$\sum_{A,B}\Big[\int_V \rho\,\ddot{\mathbf{u}}\cdot\delta\mathbf{u}\,dV\Big] + \int_I N_A\,\mathbf{n}\cdot\delta\mathbf{u}_A\,dS + \int_I N_B\,\mathbf{n}\cdot\delta\mathbf{u}_B\,dS = 0.$$

Integrating from t_0^- to t_0^+,

$$\sum_{A,B}\Big[\int_V \int_{t_0^-}^{t_0^+} \rho\,\ddot{\mathbf{u}}\cdot\delta\mathbf{u}\,dV\Big] + \int_I \int_{t_0^-}^{t_0^+} \Big(N_A\,\mathbf{n}\cdot\delta\mathbf{u}_A + N_B\,\mathbf{n}\cdot\delta\mathbf{u}_B\Big)\,dS = 0. \tag{2.4.2–4}$$

But $\mathbf{n}\cdot\delta\mathbf{u}_A = \mathbf{n}\cdot\delta\mathbf{u}_B$ at t_0, and $N_B = -N_A$, so the second term in Equation 2.4.2–4 is zero. However, the constraint (Equation 2.4.2–2) must be satisfied by augmenting Equation 2.4.2–4 with a Lagrange multiplier term with $\boldsymbol{H}$ as the multiplier:

$$\sum_{A,B}\Big[\int_V \rho\Delta\dot{\mathbf{u}}\cdot\delta\mathbf{u}\,dV\Big] + \int_I \delta\Big[H\,\mathbf{n}\cdot\left(\dot{\mathbf{u}}^{-}_{AI} + \Delta\dot{\mathbf{u}}_{AI} - \dot{\mathbf{u}}^{-}_{BI} - \Delta\dot{\mathbf{u}}_{BI}\right)\Big]\,dS.$$

The first term has been integrated over t_0^- to t_0^+ to give the velocity jump term. Taking the variation in the second term and noting that $\mathbf{n}$ cannot rotate in the infinitesimal time interval because there is no discontinuity in displacement, we obtain

$$\sum_{A,B}\left[\int_V \rho\Delta\dot{\mathbf{u}}\cdot\delta\mathbf{u}\,dV\right]+\int_I H\mathbf{n}\cdot\left(\delta\mathbf{u}_{AI}-\delta\mathbf{u}_{BI}\right)dS$$
$$+\int_I\left(\Delta\dot{\mathbf{u}}_{AI}-\Delta\dot{\mathbf{u}}_{BI}\right)\cdot\mathbf{n}\,\delta H\,dS \qquad (2.4.2\text{–}5)$$
$$=-\int_I\left(\dot{\mathbf{u}}^-_{AI}-\dot{\mathbf{u}}^-_{BI}\right)\cdot\mathbf{n}\,\delta H\,dS.$$

This equation is the impulse condition, which can be solved for the velocity jump $\Delta\dot{\mathbf{u}}$ at all nodes. The solution also provides the impulse per unit area, H: the time integral of the pressure between the surfaces over the infinitesimal time of impact. The equilibrium equation written at time t_0^+ and including the constraint that $\mathbf{n}\cdot(\ddot{\mathbf{u}}_{AI}-\ddot{\mathbf{u}}_{BI})=0$ can then be used to obtain the initial accelerations immediately after impact.

The two sets of equations (for the velocity jumps and the initial accelerations at t_0^+) require solution of the mass matrix, augmented by the constraint, with different right-hand sides for the initial acceleration and the velocity jump equations. When the elements attached to the nodes that impact have consistent mass matrices, the equations will give velocity and acceleration jumps throughout that part of the model and not just at the impacting nodes themselves. From these initial conditions at t_0^+ the usual time stepping equations can continue, augmented by the constraint that $\int_I \mathbf{n}\cdot(\dot{\mathbf{u}}_A-\dot{\mathbf{u}}_A)\,dS=0$, which is imposed by a Lagrange multiplier. Since this multiplier represents the interface pressure, its value is monitored for possible separation: if a negative value is seen (that is, if a state of tension exists between A and B across I), the constraint is removed: this requires another solution for the initial accelerations just after removal of the constraint, although there will be no velocity jump at that time. Separation can occur immediately after impact. If so, the equilibrium equations must be solved again without the constraint to find the corresponding accelerations.

In any real problem, impact and separation will occur at some intermediate point in a time step. To accommodate this, Abaqus/Standard first solves the time step by ignoring impact, then estimates (by linear interpolation) the average time of impact or separation of all points that change in the increment, t_I, and again solves the increment to that time. All surface contact changes are assumed to take place at that interpolated time point, with the surface clearances adjusted as necessary. For reasonable time steps this geometric adjustment is slight.

Typically, the time step used in the solution following a severe impact event is one or two orders of magnitude smaller than that preceding the event. As the high frequency noise generated by impact dissipates (through plasticity and the artificial damping introduced by the parameter α in the time integration operator), the time step is expanded by the automatic time stepping algorithm and the solution proceeds accordingly.

Soft contact conditions treat impact as an "elastic" event that does not destroy any kinetic energy. The change in velocity is determined by the amount of interpenetration. Thus, with soft contact the standard implicit integration procedure is used and no impulse equations need be solved. Under truly high velocity impact conditions, this may result in nodes bouncing back immediately after impact. This may lead to excessive contact *chattering*, resulting in convergence problems and small time increments. For this reason, softened contact is not recommended in "true" impact calculations. However, in certain dynamic calculations where impact effects are not critical—such as sheet forming, drop forging, or rolling operations—soft contact can work well because cutbacks due to impact calculations are avoided. The soft contact constraint

is enforced via Lagrange multipliers. If the soft contact constraint compatibility is not satisfied within the given tolerance, a severe discontinuity iteration is performed.

Reference

- "Implicit dynamic analysis using direct integration," Section 6.3.2 of the Abaqus Analysis User's Guide

2.4.3 SUBSPACE DYNAMICS

Product: Abaqus/Standard

An alternative approach provided in Abaqus/Standard for analysis of nonlinear dynamic problems is the "subspace projection" method. This method uses direct, explicit integration of the dynamic equations of equilibrium written in terms of a vector space spanned by a number of eigenvectors. The eigenmodes of the system, extracted in an eigenfrequency step prior to the dynamic analysis, are used as the global basis vectors. This method can be very effective for systems with mild nonlinearities that do not substantially change the mode shapes. The method cannot be used for contact problems. As with the other direct integration methods, it is more expensive in terms of computer time than the modal methods of purely linear dynamic analysis, but it is often significantly less expensive than the direct integration of all of the equations of motion of the model. This method is implemented in Abaqus/Standard using the explicit (central difference) operator to integrate the equations of motion projected onto the eigenmodes. In this procedure the explicit integration method is particularly effective because the eigenmodes are orthogonal with respect to the mass matrix so that the projected system always has a diagonal mass matrix and because the stability limit is determined by the highest eigenfrequency associated with the modes used in the analysis and not by the highest eigenfrequency of the structure.

Let us write the finite element approximation to the virtual work equations as

$$\delta u^N \left(M^{NM} \ddot{u}^M + I^N - P^N \right) = 0, \tag{2.4.3–1}$$

where

$$M^{NM} = \int_{V_0} \rho_0 \, \mathbf{N}^N \cdot \mathbf{N}^M \, dV_0$$

is the consistent mass matrix,

$$I^N = \int_{V_0} \boldsymbol{\beta}^N : \boldsymbol{\sigma} \, dV_0$$

is the internal force vector, and

$$P^N = \int_S \mathbf{N}^N \cdot \mathbf{t} \, dS + \int_V \mathbf{N}^N \cdot \mathbf{F} \, dV$$

is the external force vector.

The nodal displacements, velocities, accelerations, and the variations in displacement are expressed in terms of eigenmodes:

$$
\begin{aligned}
u^N &= \sum_\alpha \Phi_\alpha^N q_\alpha, \qquad & \dot{u}^N &= \sum_\alpha \Phi_\alpha^N \dot{q}_\alpha, \\
\ddot{u}^N &= \sum_\alpha \Phi_\alpha^N \ddot{q}_\alpha, \qquad & \delta u^N &= \sum_\alpha \Phi_\alpha^N \delta q_\alpha,
\end{aligned}
\tag{2.4.3–2}
$$

where q_α, $\dot{q}_\alpha$, $\ddot{q}_\alpha$, and δq_α represent generalized displacement, velocity, acceleration, and displacement variation, respectively. Substitution into the virtual work expression yields the formula for the acceleration associated with mode α as

$$
\ddot{q}_\alpha = \frac{1}{m_\alpha} \Phi_\alpha^N \left(P^N - I^N \right) \qquad \text{(no sum on } \alpha\text{)}, \tag{2.4.3–3}
$$

where m_α is the generalized mass associated with mode α:

$$
m_\alpha = \Phi_\alpha^N M^{NM} \Phi_\alpha^M .
$$

From Equation 2.4.3–3 it is seen that the element residuals are projected onto the vector space spanned by the chosen number of eigenmodes. Having calculated the generalized acceleration for each mode, the generalized displacement and velocity are calculated with the central difference operator

$$
\begin{aligned}
q_{t+\Delta t} &= q_t + \ddot{q}_t \Delta t^2, \\
\dot{q}_{t+\Delta t} &= \Delta q_t / \Delta t + \ddot{q}_t \Delta t / 2.
\end{aligned}
\tag{2.4.3–4}
$$

The nodal values for all kinematic variables are obtained using the formulæ in Equation 2.4.3–2.

When initial velocities are applied, specified either explicitly by the user or implicitly by continuation of the previous dynamic step, the initial velocity vector $\dot{u}_0$ has to be projected into the eigenspace. This leads to an initial generalized velocity for the mode α in the form

$$
\dot{q}_\alpha^0 = \frac{1}{m_\alpha} \Phi_\alpha^N M^{NM} \dot{u}_0^M .
$$

As in standard explicit dynamic integration, the method is conditionally stable. The stability limit is determined by the highest eigenfrequency of the modes used in the analysis:

$$
\Delta t_{stable} = \frac{2}{\omega_{\max}},
$$

where $\omega_{\max}$ is the highest circular frequency of the eigenmodes that are used as the basis of the solution. Since we will generally use a relatively small number of modes, this stability limit is usually much less restrictive than the stability limit for standard explicit integration.

Throughout the procedure a fixed time increment is used: the value is chosen as the smaller of the time increment specified by the user, or 80% of the stable time increment. The 80% factor is intended as a safety factor, so a small increase in the highest eigenfrequency caused by nonlinear effects will not cause the integration to become unstable.

The number of eigenvectors spanning the solution space for a subspace dynamic analysis can be specified by the user. The default number of vectors will be equal to the number of eigenvectors extracted in the eigenfrequency calculation.

It is possible to perform a subspace dynamic simulation for some time and then reextract the modes based on the current, stressed geometry (by using another eigenfrequency extraction step), followed by continuation of the analysis with the new modes as the subspace basis system. This can improve the accuracy of the method in certain cases.

Note that the method is noniterative; hence, there are no tolerances required for the procedure.

Reference

- "Implicit dynamic analysis using direct integration," Section 6.3.2 of the Abaqus Analysis User's Guide

2.4.4 EQUIVALENT RIGID BODY DYNAMIC MOTION

Product: Abaqus/Standard

It is often useful to obtain the equivalent rigid body motion of part of a model (or of the whole model): the position and translational velocity of the part's center of mass and its angular rotation and velocity about the same center of mass. Abaqus/Standard provides such output, based on equivalent momentum. This section defines how these values are calculated.

Let V be the volume of a part for which the equivalent rigid body motion values are requested. The density of the part in its initial configuration is $\rho_0(S_i)$, where S_i, $i = 1, 2, 3$, are material coordinates in the part. The spatial position of a material particle in its initial configuration is $\mathbf{X}(S_i)$ and in the current configuration is $\mathbf{x}(S_i)$, resulting in a displacement of $\mathbf{u}(S_i)$. We wish to compute the current spatial position of the center of mass of the part, $\mathbf{x}_\circ$; the translational velocity of an equivalent rigid body, $\dot{\mathbf{x}}_\circ$; the angular velocity of this equivalent rigid body, $\boldsymbol{\omega}_\circ$; the translational displacement of an equivalent rigid body motion, $\mathbf{u}_\circ$; and the rotation of an equivalent rigid body motion around the center of mass, $\boldsymbol{\phi}$. In these definitions an "equivalent rigid body" means a rigid body with the same mass distribution and the same translational and angular momentum as the actual deforming part in the current configuration.

For simplicity of notation we define some quantities. The mass of the part is

$$\mathcal{M} \stackrel{\text{def}}{=} \int_{V^0} \rho_0 \, dV^0.$$

Its first mass moment about the origin is

$$\mathcal{J}_\circ \stackrel{\text{def}}{=} \int_{V^0} \rho_0 \, \mathbf{x} \, dV^0.$$

Its second mass moment about the origin is

$$\mathcal{I}_\circ \stackrel{\text{def}}{=} \int_{V^0} \rho_0 (\mathbf{x} \cdot \mathbf{x} \, \mathbf{I} - \mathbf{x} \, \mathbf{x}) \, dV^0,$$

where $\mathbf{I}$ is a unit matrix.

It is convenient to invoke the relation $\mathbf{x} = N^N \mathbf{x}^N$ (the summation convention is assumed), where N^N are the finite element interpolation functions associated with each degree of freedom and $\mathbf{x}^N$ is the vector of current nodal positions. We can now write

$$\mathcal{I}_\circ \stackrel{\text{def}}{=} \int_{V^0} \rho_0 (N^N N^M \mathbf{x}^N \cdot \mathbf{x}^M \, \mathbf{I} - N^N N^M \mathbf{x}^N \, \mathbf{x}^M) \, dV^0.$$

Recognizing that the primitive mass matrix is

$$M^{NM} \stackrel{\text{def}}{=} \int_{V^0} \rho_0 \, N^N N^M \, dV^0,$$

we have

$$\mathcal{I}_\circ \stackrel{\text{def}}{=} M^{NM}(\mathbf{x}^N \cdot \mathbf{x}^M \, \mathbf{I} - \mathbf{x}^N \, \mathbf{x}^M).$$

We can immediately obtain

$$\mathbf{x}_\circ = \mathcal{J}_\circ / \mathcal{M}$$

and, by equating the translational momentum of the equivalent and the actual body,

$$\dot{\mathbf{x}}_\circ = \int_{V^0} \rho_0 \, \dot{\mathbf{x}} \, dV^0 / \mathcal{M}.$$

The angular velocity of the part is defined by equating the angular momentum of the part and of the equivalent rigid body about the center of mass:

$$\int_{V^0} \rho_0 (\mathbf{x} - \mathbf{x}_\circ) \times \dot{\mathbf{x}} \, dV^0 = \mathcal{I} \cdot \boldsymbol{\omega}_\circ,$$

where

$$\mathcal{I} = \mathcal{I}_\circ + \mathcal{M}(\mathbf{x}_\circ \, \mathbf{x}_\circ - \mathbf{x}_\circ \cdot \mathbf{x}_\circ \, \mathbf{I})$$

is the second mass moment of the part about its center of mass.

Abaqus uses the lumped mass formulation for low-order elements. As a consequence, the second mass moments of inertia can deviate from the theoretical values, especially for coarse meshes. Certain Abaqus elements produce lumped or structural contributions to this second mass moment (rotary inertias) not shown in these equations.

This provides

$$\boldsymbol{\omega}_\circ = \mathcal{I}^{-1} \cdot [\mathcal{H}_\circ - \mathcal{J}_\circ \times \dot{\mathbf{x}}_\circ],$$

where

$$\mathcal{H}_\circ \stackrel{\text{def}}{=} \int_{V^0} \rho_0 \, \mathbf{x} \times \dot{\mathbf{x}} \, dV^0 = M^{NM} \mathbf{x}^N \times \dot{\mathbf{x}}^M$$

is the angular momentum of the part about the origin.

The perceived translational motion of the center of mass in an equivalent rigid body motion is calculated as

$$\mathbf{u}_\circ = \int_{V^0} \rho_0 \, \mathbf{u} \, dV^0 / \mathcal{M}.$$

The equivalent rigid body rotation of the part with respect to its center of mass requires some conceptual approximations as follows. Denote the relative positions of a material particle with respect to the center of

mass in the undeformed configuration and in the deformed configuration $\overline{\mathbf{X}}$ and $\overline{\mathbf{x}}$, respectively. Consider that the configurations are known and that the axis of rotation of the body is denoted by the unit vector $\mathbf{p}$. A material particle sees such rotation relative to the center of mass as

$$\boldsymbol{\phi} = \phi\mathbf{p} = \frac{\overline{\mathbf{X}}_p \times \overline{\mathbf{x}}_p}{|\overline{\mathbf{X}}_p \times \overline{\mathbf{x}}_p|} \tan^{-1}\left(\frac{|\overline{\mathbf{X}}_p \times \overline{\mathbf{x}}_p|}{\overline{\mathbf{X}}_p \cdot \overline{\mathbf{x}}_p}\right),$$

where the subscript $\boldsymbol{p}$ denotes the projection of a vector into a plane normal to $\mathbf{p}$. We now generalize this concept by integration of the constituent parts. Define

$$\mathbf{a} = \int_V \rho\, \overline{\mathbf{X}}_p \times \overline{\mathbf{x}}_p \, dV, \qquad b = \int_V \rho\, \overline{\mathbf{X}}_p \cdot \overline{\mathbf{x}}_p \, dV.$$

The average Euler rotation then follows with the equation

$$\boldsymbol{\phi}_{av} = \frac{\mathbf{a}}{|\mathbf{a}|} \tan^{-1}\left(\frac{|\mathbf{a}|}{b}\right).$$

These integrals are not easily calculated, but with the modifications below they can be expanded in such a way that the (initially unknown) current center of mass, $\mathbf{x}_\circ$, only appears in products with the position of particles in the known current configuration.

A necessary condition for the validity of the intuitive generalization above is that if the part undergoes an arbitrary rigid body rotation, the formula returns the rotation. That can easily be proved as follows. Every material particle rotates exactly an angle $\boldsymbol{\phi} = \boldsymbol{\phi}_{av}$ in such a way that

$$\mathbf{a} = \mathbf{p}\, \sin\phi \int_V \rho\, |\overline{\mathbf{X}}_p||\overline{\mathbf{x}}_p| \, dV, \qquad b = \cos\phi \int_V \rho\, |\overline{\mathbf{X}}_p||\overline{\mathbf{x}}_p| \, dV,$$

and, therefore,

$$\frac{\mathbf{a}}{|\mathbf{a}|} = \mathbf{p}, \qquad \frac{|\mathbf{a}|}{b} = \tan\phi.$$

In all of these equations the direction vector $\mathbf{p}$ is unknown. To determine $\mathbf{p}$ we consider the characteristics of the displacement field of a rigid body rotation. For such a field, 1) the displacement of a particle is orthogonal to the rotation vector, and 2) the displacement is orthogonal to the position vector at half the motion. In a deformable body context we try to determine $\mathbf{p}$ by forcing these two statements to be true in an average sense. Considering that we are looking strictly at a rotation with respect to the center of mass, its definition automatically ensures that the first statement is satisfied. The second condition can be written in the form

$$\int_V \rho\, \left[\mathbf{p} \times (\overline{\mathbf{x}} + \overline{\mathbf{X}})\right] \times [\overline{\mathbf{x}} - \overline{\mathbf{X}}] \, dV = 0,$$

which is a homogeneous set of equations in the components of $\mathbf{p}$ with coefficients made out of integrals of known quantities, from which $\mathbf{p}$ can be solved. We can then calculate the projection of the old and new position onto the plane normal to $\mathbf{p}$ with

$$\overline{\mathbf{X}}_p = \overline{\mathbf{X}} - \mathbf{p}\,\mathbf{p}\cdot\overline{\mathbf{X}}, \qquad \overline{\mathbf{x}}_p = \overline{\mathbf{x}} - \mathbf{p}\,\mathbf{p}\cdot\overline{\mathbf{x}},$$

and by simple substitution one then obtains

$$\mathbf{a} = \int_V \rho\,\left[\overline{\mathbf{X}} - \left(\overline{\mathbf{X}}\cdot\mathbf{p}\right)\mathbf{p}\right] \times \left[\overline{\mathbf{x}} - \left(\overline{\mathbf{x}}\cdot\mathbf{p}\right)\mathbf{p}\right]\,dV = \mathbf{p}\,\mathbf{p}\cdot\int_V \rho\,\overline{\mathbf{X}}\times\overline{\mathbf{x}}\,dV,$$

where the vector $\int_V \rho\,\overline{\mathbf{X}}\times\overline{\mathbf{x}}\,dV$ is easily calculated from available quantities. With $\mathbf{p}$ known, $\mathbf{a}$ becomes determined. The quantity $\boldsymbol{b}$ can be calculated using with the same expressions, which yields

$$b = \int_V \rho\,\left[\overline{\mathbf{X}} - (\overline{\mathbf{X}}\cdot\mathbf{p})\mathbf{p}\right]\cdot\left[\overline{\mathbf{x}} - (\overline{\mathbf{x}}\cdot\mathbf{p})\mathbf{p}\right]\,dV = (\mathbf{I} - \mathbf{p}\,\mathbf{p}) : \int_V \rho\,\overline{\mathbf{X}}\,\overline{\mathbf{x}}\,dV.$$

Once $\mathbf{a}$ and $\boldsymbol{b}$ are known, ϕ is readily determined.

The determination of the equivalent rigid body rotation is based on average particle translations. Rotational degrees of freedom are ignored in the calculation of this variable; it is assumed that such rotations will produce motions of points that will measurably contribute to the calculation. However, it is possible to find pathological cases in which that would not be the case; for instance, if the part of the model considered consists of rotary inertia elements only, the calculated average rigid body rotation will be calculated as zero, even if the elements have indeed rotated.

Reference

- "Implicit dynamic analysis using direct integration," Section 6.3.2 of the Abaqus Analysis User's Guide

2.4.5 EXPLICIT DYNAMIC ANALYSIS

Product: Abaqus/Explicit

The explicit dynamics analysis procedure in Abaqus/Explicit is based upon the implementation of an explicit integration rule together with the use of diagonal or "lumped" element mass matrices. The equations of motion for the body are integrated using the explicit central difference integration rule

$$\dot{\mathbf{u}}^{(i+\frac{1}{2})} = \dot{\mathbf{u}}^{(i-\frac{1}{2})} + \frac{\Delta t^{(i+1)} + \Delta t^{(i)}}{2}\ddot{\mathbf{u}}^{(i)},$$

$$\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + \Delta t^{(i+1)}\dot{\mathbf{u}}^{(i+\frac{1}{2})},$$

where $\dot{\mathbf{u}}$ is velocity and $\ddot{\mathbf{u}}$ is acceleration. The superscript (i) refers to the increment number and $i - \frac{1}{2}$ and $i + \frac{1}{2}$ refer to midincrement values. The central difference integration operator is explicit in that the kinematic state can be advanced using known values of $\dot{\mathbf{u}}^{(i-\frac{1}{2})}$ and $\ddot{\mathbf{u}}^{(i)}$ from the previous increment. The explicit integration rule is quite simple but by itself does not provide the computational efficiency associated with the explicit dynamics procedure. The key to the computational efficiency of the explicit procedure is the use of diagonal element mass matrices because the inversion of the mass matrix that is used in the computation for the accelerations at the beginning of the increment is triaxial:

$$\ddot{\mathbf{u}}^{(i)} = \mathbf{M}^{-1} \cdot (\mathbf{F}^{(i)} - \mathbf{I}^{(i)}),$$

where $\mathbf{M}$ is the diagonal lumped mass matrix, $\mathbf{F}$ is the applied load vector, and $\mathbf{I}$ is the internal force vector. The explicit procedure requires no iterations and no tangent stiffness matrix.

Special treatment of the mean velocities $\dot{\mathbf{u}}^{(i+\frac{1}{2})}$, $\dot{\mathbf{u}}^{(i-\frac{1}{2})}$ etc. is required for initial conditions, certain constraints, and presentation of results. For presentation of results, the state velocities are stored as a linear interpolation of the mean velocities:

$$\dot{\mathbf{u}}^{(i+1)} = \dot{\mathbf{u}}^{(i+\frac{1}{2})} + \frac{1}{2}\Delta t^{(i+1)}\ddot{\mathbf{u}}^{(i+1)}.$$

The central difference operator is not self-starting because the value of the mean velocity $\dot{\mathbf{u}}^{(-\frac{1}{2})}$ needs to be defined. The initial values (at time $t = 0$) of velocity and acceleration are set to zero unless they are specified by the user. We assert the following condition:

$$\dot{\mathbf{u}}^{(+\frac{1}{2})} = \dot{\mathbf{u}}^{(0)} + \frac{\Delta t^{(1)}}{2}\ddot{\mathbf{u}}^{(0)}.$$

Substituting this expression into the update expression for $\dot{\mathbf{u}}^{(i+\frac{1}{2})}$ yields the following definition of $\dot{\mathbf{u}}^{(-\frac{1}{2})}$:

$$\dot{\mathbf{u}}^{(-\frac{1}{2})} = \dot{\mathbf{u}}^{(0)} - \frac{\Delta t^{(0)}}{2}\ddot{\mathbf{u}}^{(0)}.$$

Stability

The explicit procedure integrates through time by using many small time increments. The central difference operator is conditionally stable, and the stability limit for the operator (with no damping) is given in terms of the highest eigenvalue in the system as

$$\Delta t \leq \frac{2}{\omega_{max}}.$$

In Abaqus/Explicit a small amount of damping is introduced to control high frequency oscillations. With damping the stable time increment is given by

$$\Delta t \leq \frac{2}{\omega_{max}}(\sqrt{1+\xi^2}-\xi),$$

where ξ is the fraction of critical damping in the highest mode. Contrary to our usual engineering intuition, introducing damping to the solution reduces the stable time increment.

The time incrementation scheme in Abaqus/Explicit is fully automatic and requires no user intervention. Abaqus/Explicit uses an adaptive algorithm to determine conservative bounds for the highest element frequency. An estimate of the highest eigenvalue in the system can be obtained by determining the maximum element dilatational mode of the mesh. The stability limit based upon this highest element frequency is conservative in that it will give a smaller stable time increment than the true stability limit that is based upon the maximum frequency of the entire model. In general, constraints such as boundary conditions and contact have the effect of compressing the eigenvalue spectrum, which the element by element estimates do not take into account. Abaqus/Explicit contains a global estimation algorithm, which determines the maximum frequency of the entire model. This algorithm continuously updates the estimate for the maximum frequency. Abaqus/Explicit initially uses the element by element estimates. As the step proceeds, the stability limit will be determined from the global estimator once the algorithm determines that the accuracy of the global estimation is acceptable. The global estimation algorithm is not used when any of the following capabilities are included in the model: fluid elements, JWL equation of state, infinite elements, material damping, dashpots, thick shells (thickness to characteristic length ratio larger than 0.92) or thick beams (thickness to length ratio larger than 1.0), and nonisotropic elastic materials with temperature and field variable dependency.

A trial stable time increment is computed for each element in the mesh using the following expression:

$$\Delta t = \frac{2}{\omega_{max}^{element}},$$

where $\omega_{max}^{element}$ is the element maximum eigenvalue. A conservative estimate of the stable time increment is given by the minimum taken over all the elements. The above stability limit can be rewritten as

$$\Delta t = min(\frac{L_e}{c_d}),$$

where L_e is the characteristic element dimension and c_d is the current effective, dilatational wave speed of the material. The characteristic element dimension is derived from an analytic upper bound expression for the maximum element eigenvalue. Considering the 4-node uniform strain quadrilateral (CPE4R), the characteristic element dimension is

$$L_e = \frac{A}{\sqrt{B_{iI}B_{iI}}},$$

where A is the element area and B_{iI} is the element gradient operator (see "Solid isoparametric quadrilaterals and hexahedra," Section 3.2.4). Similar characteristic element dimensions are derived for all element types found in Abaqus/Explicit.

The current dilatational wave speed is determined in Abaqus/Explicit by calculating the effective hypoelastic material moduli from the material's constitutive response. Effective Lamé's constants, $\hat{\lambda}$ and $\hat{\mu}$, are determined in the following manner. We define Δp as the increment in the equivalent pressure stress, $p = -\frac{1}{3}\mathrm{trace}(\boldsymbol{\sigma})$, $\Delta\mathbf{S}$ as the increment in the deviatoric stress, $\Delta\epsilon_{vol}$ as the increment of volumetric strain, and $\Delta\mathbf{e}$ as the deviatoric strain increment. We assume a hypoelastic stress-strain rule of the form

$$\Delta p = -\hat{K}\Delta\epsilon_{vol},$$

$$\Delta\mathbf{S} = 2\hat{\mu}\Delta\mathbf{e}\ ,$$

where $\hat{K}$ is the effective bulk modulus. The effective moduli can then be computed as

$$\hat{K} = \frac{-\Delta p}{\Delta\epsilon_{vol}},$$

$$\hat{\mu} = \frac{1}{2}\frac{\Delta\mathbf{S} : \Delta\mathbf{e}}{\Delta\mathbf{e} : \Delta\mathbf{e}},$$

$$\hat{\lambda} = \hat{K} - \frac{2}{3}\hat{\mu}\ .$$

If the strain increments are insignificant, these relationships will not yield numerically meaningful results. In this circumstance Abaqus/Explicit sets the effective Lamé's constants to the initial values for the material, λ_o and $2\mu_o$. In the case where the volumetric strain increment is significant but the deviatoric stress increment is not, the effective shear modulus is estimated to be

$$2\hat{\mu} = \frac{1}{2}(3(\lambda_o + 2\mu_o) - 3\hat{K})\ .$$

These effective moduli represent the element stiffness and determine the current dilatational wave speed in the element as

$$c_d = \sqrt{\frac{(\hat{\lambda} + 2\hat{\mu})}{\rho}} \ .$$

Bulk viscosity

Bulk viscosity introduces damping associated with the volumetric straining. Its purpose is to improve the modeling of high-speed dynamic events.

There are two forms of bulk viscosity in Abaqus/Explicit. The first is found in all elements and is introduced to damp the "ringing" in the highest element frequency. This damping is sometimes referred to as truncation frequency damping. It generates a bulk viscosity pressure, which is linear in the volumetric strain:

$$p_{bv1} = b_1 \rho c_d L_e \dot{\epsilon}_{vol},$$

where b_1 is a damping coefficient (default=.06), ρ is the current material density, c_d is the current dilatational wave speed, L_e is an element characteristic length, and $\dot{\epsilon}_{vol}$ is the volumetric strain rate.

The second form of bulk viscosity pressure is found only in solid continuum elements (except CPS4R). This form is quadratic in the volumetric strain rate:

$$p_{bv2} = \rho(b_2 L_e \dot{\epsilon}_{vol})^2,$$

where b_2 is a damping coefficient (default=1.2) and all other quantities are as defined for the linear bulk viscosity. The quadratic bulk viscosity is applied only if the volumetric strain rate is compressive.

The quadratic bulk viscosity pressure will smear a shock front across several elements and is introduced to prevent elements from collapsing under extremely high velocity gradients. Consider a simple one element problem in which the nodes on one side of the element are fixed and the nodes on the other side have an initial velocity in the direction of the fixed nodes. If the initial velocity is equal to the dilatational wave speed of the material, the element—without the quadratic bulk viscosity—would collapse to zero volume in one time increment (because the stable time increment size is precisely the transit time of a dilatational wave across the element). The quadratic bulk viscosity pressure will introduce a resisting pressure that will prevent the element from collapsing.

The bulk viscosity pressure is not included in the material point stresses because it is intended as a numerical effect only—it is not considered to be part of the material's constitutive response. The bulk viscosity pressures are based upon the dilatational mode of each element. The fraction of critical damping in the dilatational mode of each element is given by

$$\xi = b_1 - b_2^2 \frac{L_e}{c_d} \min(0, \dot{\epsilon}_{vol}) \ .$$

Linear bulk viscosity is always included in Abaqus/Explicit. The parameters b_1 and b_2 can be redefined by the user. The default values are $b_1 = 0.06$ and $b_2 = 1.2$. The bulk viscosity parameters can

be changed from step to step. If the default values are changed in a step, the new values will be used in any subsequent steps unless they are redefined.

Rotational bulk viscosity for shell elements

For the displacement degrees of freedom, bulk viscosity introduces damping associated with volumetric straining. Linear bulk viscosity or truncation frequency damping is used to damp the high frequency ringing that leads to unwanted noise in the solution or spurious overshoot in the response amplitude. For the same reason, in shells the high frequency ringing in the rotational degrees of freedom is damped with linear bulk viscosity acting on the mean curvature strain rate. This damping generates a bulk viscosity "pressure moment," m, which is linear in the mean curvature strain rate:

$$m = b_1 \frac{h_0^2}{12} \rho c_d L \frac{\Delta\kappa}{\Delta t} ,$$

where b_1 is a damping coefficient (default = 0.06), h_0 is the original thickness, ρ is the mass density, c_d is the current dilatational wave speed, L is the characteristic length used for rotary inertia and transverse shear stiffness scaling, and $\Delta\kappa = \Delta\kappa_{11} + \Delta\kappa_{22}$ is twice the increment in mean curvature. The dilatational wave speed is given in terms of the effective Lamé constants as

$$c_d = \sqrt{\frac{(\hat{\lambda} + 2\hat{\mu})}{\rho}} .$$

The resultant pressure moment mh, where h is the current thickness, is added to the direct components of the moment resultant.

Reference

- "Explicit dynamic analysis," Section 6.3.3 of the Abaqus Analysis User's Guide

2.5 Modal dynamics

- "Eigenvalue extraction," Section 2.5.1
- "Variables associated with the natural modes of a model," Section 2.5.2
- "Linear dynamic analysis using modal superposition," Section 2.5.3
- "Damping options for modal dynamics," Section 2.5.4
- "Modal dynamic analysis," Section 2.5.5
- "Response spectrum analysis," Section 2.5.6
- "Steady-state linear dynamic analysis," Section 2.5.7
- "Random response analysis," Section 2.5.8
- "Base motions in modal-based procedures," Section 2.5.9

2.5.1 EIGENVALUE EXTRACTION

Product: Abaqus/Standard

There are many important areas of structural analysis in which it is essential to be able to extract the eigenvalues of the system and, hence, obtain its natural frequencies of vibration or investigate possible bifurcations that may be associated with kinematic instabilities. For example, structural evaluation for seismic events is often based on linear analysis, using the structure's modes up to a limiting cutoff frequency, which is usually taken as 33 Hz (cycles/second). Once the modes are available, their orthogonality property allows the linear response of the structure to be constructed as the response of a number of single degree of freedom systems. This opens the way to several response evaluation methods that are computationally inexpensive and provide useful insight into the structure's dynamic behavior. Several such methods are provided in Abaqus/Standard and are described in the following sections.

The mathematical eigenvalue problem is a classical field of study, and much work has been devoted to providing eigenvalue extraction methods. Wilkinson's (1965) book provides an excellent compendium on the problem. The eigenvalue problems arising out of finite element models are a particular case: they involve large but usually narrowly banded matrices, and only a small number of eigenpairs are usually required. For many important cases the matrices are symmetric. The eigenvalue problem for natural modes of small vibration of a finite element model is

$$\left(\mu^2 M^{MN} + \mu C^{MN} + K^{MN}\right) \phi^N = 0$$

or, in classical matrix notation,

$$\left(\mu^2 [M] + \mu [C] + [K]\right) \{\phi\} = 0, \tag{2.5.1–1}$$

where $[M]$ is the mass matrix, which is symmetric and positive definite in the problems of interest here; $[C]$ is the damping matrix; $[K]$ is the stiffness matrix, which may include large-displacement effects, such as "stress stiffening" (initial stress terms), and, therefore, may not be positive definite or symmetric; μ is the eigenvalue; and $\{\phi\}$ is the eigenvector—the mode of vibration. This equation is available immediately from a linear perturbation of the equilibrium equation of the system.

The eigensystem (Equation 2.5.1–1) in general will have complex eigenvalues and eigenvectors. This system can be symmetrized by assuming that $[K]$ is symmetric and by neglecting $[C]$ during eigenvalue extraction. The symmetrized system has real squared eigenvalues, μ^2, and real eigenvectors only.

Typically, for symmetric eigenproblems we will also assume that $[K]$ is positive semidefinite. In this case μ becomes an imaginary eigenvalue, $\mu = i\omega$, where ω is the circular frequency, and the eigenvalue problem can be written as

$$\left(-\omega^2 [M] + [K]\right) \{\phi\} = 0. \tag{2.5.1–2}$$

If the model contains hybrid elements, contact pairs, or contact elements, the system of equations contains Lagrange multipliers and the stiffness matrix $[K]$ becomes indefinite. However, all the terms of the mass matrix corresponding to the Lagrange multipliers are equal to zero. Therefore, all the eigenvalues are imaginary, and the eigenvalue problem can still be written as Equation 2.5.1–2.

Abaqus provides eigenvalue extraction procedures for both symmetric and complex eigenproblems. For symmetrized eigenproblems Abaqus/Standard offers two approaches: Lanczos and subspace iteration methods. For complex eigenproblems the subspace projection method is used.

Eigenvalue extraction for symmetric systems

The structural eigenvalue problem has received considerable attention since the advent of finite element models. Ramaswami (1979) summarizes available methods for the problem: the most attractive appear to be the Lanczos method (see, for example, Newman and Pipano, 1973; Parlett, 1980) and the subspace iteration method, a classical method that was introduced into finite element applications by Bathe and Wilson (1972).

Both the subspace iteration and the Lanczos methods, using the Householder and Q-R algorithm for the reduced eigenproblem, have been implemented in Abaqus/Standard. The various parts of these algorithms are discussed in the remainder of this section.

Subspace iteration—the basic algorithm

The application of the subspace iteration method to eigenproblems arising from finite element models of the dynamic behavior of structures has been discussed by a number of authors—see Ramaswami (1979) for references. The basic idea is a simultaneous inverse power iteration. A small set of base vectors is created, thus defining a "subspace": this "subspace" is then transformed, by iteration, into the space containing the lowest few eigenvectors of the overall system.

Assume that, at the i-th iteration, a set of m vectors, $\{V\}^n_{(i)}$, $n = 1, \ldots, m$, exists, where m is less than the number of variables in the finite element model. These are considered as the "base vectors" that define the m-dimensional subspace out of the n dimensions defined by the variables in the finite element model. We arrange these vectors as the columns in the matrix $[V]_{(i)}$. The number of rows of $[V]_{(i)}$ is, thus, the size of the complete set of equations (the number of variables in the finite element model, N), and the number of columns is the dimension of the subspace, m.

The first step in the algorithm is to define a new set of base vectors by solving

$$[K][\hat{V}]_{(i+1)} = [M][V]_{(i)}. \tag{2.5.1–3}$$

This operation, a generalized "inverse power sweep" with m vectors, involves the solution of the complete set of linear stiffness equations for several right-hand-side vectors (with, after the first iteration of the method, a previously decomposed matrix).

The stiffness and mass matrices of the structure are then projected onto the subspace by

$$[K^*] = [\hat{V}]^T_{(i+1)}[K][\hat{V}]_{(i+1)},$$
$$[M^*] = [\hat{V}]^T_{(i+1)}[M][\hat{V}]_{(i+1)},$$

thus defining a mass matrix, $[M^*]$, and a stiffness matrix, $[K^*]$, in the subspace. These matrices are of dimension m by m. The eigenproblem

$$(-\omega^2[M^*] + [K^*])\{\phi\} = 0 \tag{2.5.1–4}$$

is now solved completely in the subspace using the Householder and Q-R methods, which are discussed below.

The eigenvectors have now been defined in the reduced space and can be transformed back to the full space of the structural problem to define $[V]$ for the next iteration:

$$[V]_{(i+1)} = [\hat{V}]_{(i+1)}[\phi],$$

where $[\phi] = [\{\phi\}_1|\{\phi\}_2|\cdots|\{\phi\}_m]$. This completes an iteration.

With $m = 1$ (a one-dimensional subspace), the method reduces to the simple "inverse power sweep" method (see Wilkinson, 1965, or Strang, 1976).

The advantage of the subspace method is the extraction of the eigenvalues in reduced space, which will cause a rapid convergence to the eigenvectors in full space. The number of base vectors carried in the iterations and the choice of initial base vectors are, therefore, important for an economical solution. The convergence rate of a particular eigenvalue is proportional to $\omega_i^2/\omega_{(m+1)}^2$, where $\omega_{(m+1)}^2$ is the eigenvalue corresponding to the next vector, $m + 1$ beyond the m vectors used to define the subspace. The default value of m used in Abaqus is $m = \min(2p, p + 8)$, where p is the number of eigenvectors requested. If more vectors are used, the number of required iterations is reduced, but each iteration takes longer because of the greater number of right-hand sides. Increasing m can sometimes improve the performance of the algorithm significantly.

The choice of starting vectors is also important. There is no requirement that these should be close to the eigenvectors for rapid convergence. The Householder and Q-R steps rotate the vectors into the eigenvectors as long as the base vectors span the space of the eigenvectors. The approach used is to choose initial vectors that span this space as completely as possible. The algorithm used in Abaqus/Standard for this purpose is that of Bathe and Wilson (1972). They recommend using the diagonal mass terms as one vector: the other vectors are unit vectors, providing a set of single unit entries at the nodes and the degrees of freedom with the largest mass terms. A check is made to ensure that all degrees of freedom are represented in these additional vectors.

This completes the description of the basic subspace iteration algorithm as it is implemented in Abaqus/Standard. The small eigenproblem, for which all eigenpairs must be found, is solved by the Householder and Q-R algorithms. These algorithms are described at the end of this section.

Lanczos eigensolver

The implementation of the Lanczos eigensolver as a powerful tool for extraction of the extreme eigenvalues and the corresponding eigenvectors of a sparse symmetric generalized eigenproblem has been discussed by a number of authors; see, for example, Parlett (1980), Parlett and Nour-Omid (1989), Simon (1984), and Ericsson and Ruhe (1980). A shifted block Lanczos algorithm was developed and described in detail by Grimes, Lewis, and Simon (1994).

The Lanczos procedure in Abaqus/Standard consists of a set of Lanczos "runs," in each of which a set of iterations called steps is performed. For each Lanczos run the following spectral transformation is applied:

$$[M]\left([K]-\sigma[M]\right)^{-1}[M]\{\phi\}=\theta[M]\{\phi\}, \tag{2.5.1–5}$$

where σ is the shift, θ is the eigenvalue, and $\{\phi\}$ is the eigenvector. This transformation allows rapid convergence to the desired eigenvalues. The eigenvectors of the symmetrized problem (Equation 2.5.1–2) and the transformed problem (Equation 2.5.1–5) are identical, while the eigenvalues of the original problem and the transformed problem are related in the following manner:

$$\omega^2=\frac{1}{\theta}+\sigma.$$

A Lanczos run will be terminated when its continuation is estimated to be inefficient. In general, only tens of eigenvalues (closest to the shift value) are computed in a single Lanczos run. The possibility of computing many eigenmodes by carrying out several runs with different shift values is an important feature of the Lanczos eigensolver.

Within each run a sequence of Krylov subspaces is created, and the best possible approximation of the eigenvectors on each subspace is computed in a series of "steps." In each Lanczos step the dimension of the subspace grows, allowing better approximation of the desired eigenvectors. This is in contrast to the subspace iteration method, in which the dimension of the subspace used to approximate the eigenvectors is fixed.

In theory the basic Lanczos process (in the assumption of "exact" computations without taking into account round-off errors) is able to determine only simple eigenvalues. The shifting strategy (and the Sturm sequence check as a part of it) detects missing modes and enforces computation of all the modes during the subsequent Lanczos runs. However, this strategy is expensive if the multiplicity of certain eigenvalues is high. Therefore, a "blocked" version of the Lanczos algorithm is implemented in Abaqus/Standard. The idea is to start with a block of orthogonal vectors and to increase the dimension of the Krylov subspaces by the block size at each Lanczos step. This approach allows automatic computation of all multiple eigenvalues if the largest multiplicity does not exceed the block size. Another important advantage of the blocked Lanczos method is that it allows efficient implementation of expensive computational kernels such as matrix-blocked vector multiplications, blocked back substitutions, and blocked vector products.

As discussed above, the Lanczos process consists of several Lanczos runs. Each Lanczos run is associated with some shift value that remains constant during the run. The initial shift value, σ_1, is determined (by a heuristic approach) using the geometric mean of the centers of the Gershgorin circles, termed the "problem scale." At the beginning of each Lanczos run a factorization of the shifted matrix $[K]-\sigma_p[M]$ (where p is the run number) is carried out, after which a sequence of Lanczos steps is performed.

Each block Lanczos step i is implemented in Abaqus/Standard in the following manner:

1. Solve the system of linear equations with b right-hand sides (b is the Lanczos block size) using the factorized shifted matrix:

$$([K] - \sigma_p[M])\,[\hat{V}]_{(i+1)} = [M][V]_{(i)},$$

where $[V]_{(i)}$ is a block of Lanczos vectors. The number of rows of $[V]_{(i)}$ is the number of variables in the finite element model, and the number of columns is the Lanczos block size b. The initial block of vectors $[V]_{(1)}$ is a set of random vectors orthonormalized using the procedure: $[I] = [V]_{(1)}^T[M][V]_{(1)}$, where $[I]$ is the identity matrix.

2. Set the auxiliary block of vectors:

$$[U]_{(i+1)} = [\hat{V}]_{(i+1)} - [V]_{(i-1)}\beta_i,$$

where β_i is a $b \times b$ upper triangular matrix. $\beta_1 = [0]$.

3. Compute the symmetric matrix α_i of size b:

$$\alpha_i = [U]_{(i+1)}^T[M][V]_{(i)}.$$

4. Formulate the Lanczos residual:

$$[R]_{(i+1)} = [U]_{(i+1)} - [V]_{(i)}\alpha_i.$$

5. Normalize the residual. Compute a block of vectors, $[V]_{(i+1)}$, such that

$$[V]_{(i+1)}\beta_{i+1} = [R]_{(i+1)}$$

and

$$[I] = [V]_{(i+1)}^T[M][V]_{(i+1)},$$

where β_{i+1} is a $b \times b$ upper triangular matrix.

6. Estimate the loss of orthogonality between $[V]_{(i+1)}$ and $[V]_{(j)}$, for $j = 1, 2, ..., i-1$, using the partial reorthogonalization technique; and perform reorthogonalization if necessary (see Simon, 1984; Grimes, Lewis, and Simon, 1994).

7. Perform "local reorthogonalization": recompute $[V]_{(i+1)}$ to provide

$$[0] = [V]_{(i+1)}^T[M][V]_{(i)}.$$

8. Solve the following reduced eigenvalue problem for the band matrix $[T]$:

$$[T][S] = [S]\,[\theta]\,,$$

where

$$[T] = \begin{bmatrix} \alpha_1 & \beta_2^T & & & & \\ \beta_2 & \alpha_2 & \beta_3^T & & & \\ & \beta_3 & \alpha_3 & \beta_4^T & & \\ & & \beta_4 & \alpha_4 & \ldots & \\ & & & \ldots & \ldots & \beta_i^T \\ & & & & \beta_i & \alpha_i \end{bmatrix}$$

and $[S]$ and $[\theta]$ are, respectively, the matrices containing the eigenvectors and eigenvalues of the reduced eigenproblem. This problem is solved by the Householder and Q-R algorithms, which are discussed below.

9. Determine the error bounds in eigenvalue approximation for the symmetrized eigenvalue problem (Equation 2.5.1–2) (see Parlett, 1980; Grimes, Lewis, and Simon, 1994). Check the termination conditions of the Lanczos run.

The Lanczos run terminates if one of the following conditions is satisfied:

- All the eigenvalues required for the current run are extracted.
- The triangular matrix β_{i+1} is singular or ill-conditioned.
- The number of Lanczos steps exceeds the maximum number allowed.
- Continuation of the current run is evaluated to be inefficient. This decision is based on the estimation of the "cost per eigenvalue" over the next few steps (the Lanczos run is continued as long as the cost per eigenvalue is decreasing).

After termination of each Lanczos run, the converged eigenvectors of the symmetrized problem (Equation 2.5.1–2) are recovered using the blocks of vectors $[V]_{(j)}, j = 1, 2, ..., i$ and the eigenvectors $[S]$.

Once the Lanczos run for the shift σ_p is completed, the shift value σ_{p+1} is computed using the results of all the previous Lanczos runs. To describe the shifting strategy, the following concepts are introduced:

The "computational interval" is the interval between the minimum and maximum eigenvalues of interest. This interval can be finite (both ends are finite), semi-infinite (only one end is finite), or infinite (both ends are infinite). The "center point" is the point in the computational interval nearest to the desired eigenvalues.

The Sturm sequence number, $\nu([A])$, of a real nonsingular symmetric matrix $[A]$ is the number of negative eigenvalues. This number is equal to the number of negative terms in the diagonal matrix $[D]$ of the Cholesky decomposition $[A] = [L][D][L]^T$ and is, therefore, available after the Cholesky decomposition is completed.

Let σ_1 and σ_2 be two shift values such that $(\sigma_1 < \sigma_2)$. Then the number of eigenvalues of the symmetrized problem (Equation 2.5.1–2, assuming that $[M]$ is positive definite or positive semidefinite) in the interval $[\sigma_1, \sigma_2]$ is equal to $\nu([K] - \sigma_2[M]) - \nu([K] - \sigma_1[M])$. If this number is equal to the number of eigenvalues actually determined by the Lanczos algorithm, the interval $[\sigma_1, \sigma_2]$ is called "a trust interval." The trust interval containing the center point is referred to as "a primary trust interval" (denoted by the key "+" in the trust intervals list printed in the message file). The shifting strategy is

aimed at constructing the primary trust interval inside the computational interval containing the required number of eigenvalues closest to the center point.

One of the most important features in the formulation of the shift update strategy is the concept of a "sentinel." The sentinels are the endpoints of the intervals containing exclusively converged eigenvalues closest to the current shift value (in each direction). The sentinels are computed during each Lanczos run and are updated at the end of each step after the eigenvalue analysis of the reduced matrix $[T]$ is completed. The basic assumption is that there are no missing eigenvalues inside the sentinels; therefore, they are excluded from the computation intervals for the upcoming Lanczos runs. This assumption is later verified on the basis of the Sturm sequence check, and a special procedure is activated if some eigenvalues are missing. If no missing eigenvalues are detected, the sentinels transform directly into the corresponding trust intervals.

The new shift values are selected on the basis of the nonconverged eigenvalue approximations after the Lanczos run is terminated. Assuming the same convergence properties for the upcoming runs, the new shift values are selected in such a way that the number of eigenvalues expected to be found in the upcoming runs will be close to the number of eigenvalues found inside the corresponding sentinels on the previous run.

The Householder method with quarter rotation

The Householder method is used to reduce a general matrix to a symmetric tridiagonal form. A tridiagonal matrix is one whose only nonzero entries are on or immediately adjacent to the diagonal. The first step is to transform Equation 2.5.1–4 to the form

$$\left([A] - \omega^2[I]\right)\{\phi\} = 0,$$

where $[I]$ is the identity matrix.

This is done by using a Cholesky decomposition of $[M^*]$ and then premultiplying and postmultiplying $[K^*]$ by the inverse of the lower and upper triangular matrices that are, thus, obtained:

$$[A] = [L]^{-1}[K^*][L^T]^{-1},$$

where

$$[L][L]^T = [M^*].$$

To preserve symmetry in $[A]$, it is necessary that the decomposition of $[M^*]$ result in two matrices that are the transpose of each other. A Cholesky decomposition produces the desired result but adds the requirement that the matrix $[M^*]$ be positive definite. $[M^*]$ should always be positive definite (because $[M]$ is positive definite in all of the problems considered here) if the base vectors defining the subspace are not linearly dependent across $[M]$. In practice the choices made for starting values of the base vectors usually satisfy this requirement, although cases can arise when this is not true. Then Abaqus/Standard will reduce the dimensionality of the subspace to obtain a positive definite $[M^*]$.

The Householder transformation starts with the matrix $[A]$ and, proceeding one column at a time, reduces all the entries outside the tridiagonal part of the matrix to zero by the transformation

$$[A]_2 = [U]_1^{-1}[A]_1[U]_1. \tag{2.5.1–6}$$

The matrix $[U]$ is of the form

$$[U] = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & & & \\ 0 & 0 & & [H] & \\ 0 & 0 & & & \end{pmatrix}.$$

For the first transformation $[H]$ is of the same order as $[A]$ and $[U]$; that is, $m \times m$, the dimension of the subspace. However, each transformation reduces the order of $[H]$ by one, the leading parts of $[U]$ being 1 on the diagonal and 0 outside the diagonal, as illustrated above.

The matrix $[H]$ is obtained by the following algorithm:

1. If this is the kth iteration, write the kth column of $[A]_k$ below the diagonal as

$$\{X\} = \lfloor a_{k+1,k}, a_{k+2,k} \ldots a_{n,k} \rfloor^T.$$

2. Calculate the norm $X| = \sqrt{\lfloor X \rfloor \{X\}}$.
3. Define a vector,

$$\lfloor V \rfloor = \lfloor a_{k+1,k} \rfloor + |X| \lfloor a_{k+1,k} \ldots a_{n,k} \rfloor^T.$$

4. Then,

$$[H] = [I] - 2\frac{1}{\lfloor V \rfloor \{V\}} \{V\} \lfloor V \rfloor.$$

This algorithm is developed in detail in Strang's (1976) book.

The Householder algorithm produces a symmetric tridiagonal matrix, which has the same eigenvalues as the original matrix, because the transformation (Equation 2.5.1–6) does not alter the eigenvalues.

The next step is to calculate the eigenvalues of the tridiagonal matrix. This is done by the Q-R algorithm. In this method the matrix $[A]$, which is now tridiagonal (although this is not a requirement for the method to work), is factored into $[Q][R]$, where $[Q]$ is an orthonormal matrix (that is, $[Q]^T[Q] = [I]$) and $[R]$ is an upper-triangular matrix (that is, all the terms in $[R]$ below the diagonal are zero).

The next matrix in the iteration is obtained by premultiplying and postmultiplying $[A]$ by $[Q]$:

$$[A]_{i+1} = [Q]_i^T[A]_i[Q]_i = [Q]_i^T[Q]_i[R]_i[Q]_i = [R]_i[Q]_i.$$

Because $[Q]$ is orthonormal, this will not affect the eigenvalues. This process gradually reduces the off-diagonal terms of $[A]$ so that the diagonal terms approach the eigenvalues. To speed up convergence, a method of shifting is introduced, leading to the following iterative loop:

$$[B]_i = [A]_i - \alpha_i[I]$$

$$[Q]_i[R]_i = [B]_i$$

$$[B]_{i+1} = [R]_i[Q]_i$$

$$[A]_{i+1} = [B]_{i+1} + \alpha_i[I].$$

The shift cannot be used until the iteration is converging toward an eigenvalue; but once that is obvious, the shift value α_i can be set to the expected eigenvalue. This will lead to a significant improvement in the convergence rate. If the shift is done too early, with a value that is not close to an eigenvalue, it is quite possible that the process will converge to an incorrect number. The Q-R process will converge to the eigenvalues in ascending order; and as soon as an eigenvalue is obtained, the order of the matrix can be reduced by one.

The final step after the eigenvalues have been obtained is to calculate the eigenvectors by using the inverse power method to solve for an eigenvector, given any right-hand side:

$$\left[[K^*] - \lambda[M^*]\right]\{\phi\} = \{V\},$$

where λ is the eigenvalue just obtained and $\{V\}$ is any vector. Because the left-hand-side matrix is singular in the direction of the eigenvector $\{\phi\}$, this vector will be obtained regardless of the right-hand-side vector $\{V\}$, as long as $\{V\}$ is not orthogonal to the eigenvector. To ensure that consecutive vectors are orthogonal, especially in the case of multiple eigenvalues, $\{V\}$ is always chosen to be orthogonal to the previously extracted eigenvectors. Since $\left[[K^*] - \lambda[M^*]\right]$ is singular, a slight numerical shift must be included to decompose it and, thus, solve for $\{\phi\}$. In the standard notation used in this guide, the matrix of eigenvectors $[\phi]$ is written as ϕ^N_α, where N refers to the nodal variable in the problem and $\alpha = 1, 2, \ldots, p$ is the mode number.

Complex eigenvalue extraction

Abaqus/Standard offers the subspace projection method to solve for complex eigenvalues and right eigenvectors of the original eigenproblem (Equation 2.5.1–1).

Subspace projection method

In the subspace projection method the original eigensystem (Equation 2.5.1–1) is projected onto a subspace spanned by the eigenvectors of the undamped, symmetric system (Equation 2.5.1–2). Thus, the symmetrized eigenproblem must be solved prior to the complex eigenvalue extraction procedure to create the subspace onto which the original system will be projected. Next, the original mass, damping, and stiffness matrices are projected onto the subspace of N eigenvectors:

$$[M^*] = [\phi_1, \ldots, \phi_N]^T [M][\phi_1, \ldots, \phi_N],$$

$$[C^*] = [\phi_1, \ldots, \phi_N]^T [C][\phi_1, \ldots, \phi_N],$$

$$[K^*] = [\phi_1, \ldots, \phi_N]^T [K][\phi_1, \ldots, \phi_N].$$

Then, we arrive at the following eigenproblem:

$$\left(\mu^2[M^*] + \mu[C^*] + [K^*]\right)\{\phi^*\} = 0.$$

Typically, the number of eigenvectors is relatively small; a few hundred is common. This small complex eigenvalue system is solved using the standard QZ method for generalized nonsymmetric eigenproblems. Complex eigenvalues of the projected system are the approximation of the eigenvalues of the original system (Equation 2.5.1–1), but the right eigenvectors of the original system need to be recovered:

$$\{\phi\}^k = [\phi_1, \ldots, \phi_N]\{\phi^*\}^k,$$

where $\{\phi\}^k$ is the approximation of k-th right eigenvector of the original system.

The only energy density calculated in the eigenvalue extraction procedures is the elastic strain energy density, which for the complex eigenvalue extraction is defined as follows:

$$E_S = \frac{1}{4}[(\boldsymbol{\sigma}_r : \boldsymbol{\varepsilon}_r + \boldsymbol{\sigma}_i : \boldsymbol{\varepsilon}_i) + \sqrt{(\boldsymbol{\sigma}_r : \boldsymbol{\varepsilon}_r - \boldsymbol{\sigma}_i : \boldsymbol{\varepsilon}_i)^2 + (\boldsymbol{\sigma}_r : \boldsymbol{\varepsilon}_i + \boldsymbol{\sigma}_i : \boldsymbol{\varepsilon}_r)^2}],$$

where $\boldsymbol{\sigma}_r, \boldsymbol{\sigma}_i$ are real and imaginary components of the eigenstress tensor and $\boldsymbol{\varepsilon}_r, \boldsymbol{\varepsilon}_i$ are real and imaginary components of the eigenstrain tensor.

References

- "Eigenvalue buckling prediction," Section 6.2.3 of the Abaqus Analysis User's Guide
- "Natural frequency extraction," Section 6.3.5 of the Abaqus Analysis User's Guide
- "Complex eigenvalue extraction," Section 6.3.6 of the Abaqus Analysis User's Guide

2.5.2 VARIABLES ASSOCIATED WITH THE NATURAL MODES OF A MODEL

After an eigenfrequency step has been used to find the eigenvalues of a model, Abaqus/Standard automatically calculates the participation factor, the effective mass, and the composite modal damping for each mode so that these variables are available for use in subsequent linear dynamic analysis.

Generalized mass

The "generalized mass" associated with mode α is

$$m_\alpha = \phi_\alpha^N M^{NM} \phi_\alpha^M \quad \text{(no sum over } \alpha\text{)},$$

where M^{NM} is the structure's mass matrix and ϕ_α^N is the eigenvector for mode α. The superscripts N and M refer to degrees of freedom of the finite element model.

Abaqus/Standard allows the user to choose between two types of eigenvector normalization: the eigenvectors can be scaled so that the largest entry in each vector is unity, or they can be normalized so that the generalized mass for each vector is unity ($m_\alpha = 1$). The choice of eigenvector normalization type does not influence the results of subsequent modal dynamic steps. The normalization type determines only the manner in which the eigenvectors are represented.

Modal participation factors

The participation factor for mode α in direction i, $\Gamma_{\alpha i}$, is a variable that indicates how strongly motion in the global x-, y- or z-direction or rotation about one of these axes (indicated by i, $i = 1, 2, \ldots, 6$) is represented in the eigenvector of that mode. It is defined as

$$\Gamma_{\alpha i} = \frac{1}{m_\alpha} \phi_\alpha^N M^{NM} T_i^M \quad \text{(no sum over } \alpha\text{)},$$

where T_i^M defines the magnitude of the rigid body response of a degree of freedom in the model (M) to imposed rigid body motion (displacement or infinitesimal rotation) in the i-direction. For example, at a node with the usual three displacement and three rotation components, T_i^M is

$$\begin{pmatrix} 1 & 0 & 0 & 0 & (z-z_0) & -(y-y_0) \\ 0 & 1 & 0 & -(z-z_0) & 0 & (x-x_0) \\ 0 & 0 & 1 & (y-y_0) & -(x-x_0) & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{Bmatrix} \hat{e}_1 \\ \hat{e}_2 \\ \hat{e}_3 \\ \hat{e}_4 \\ \hat{e}_5 \\ \hat{e}_6 \end{Bmatrix},$$

where $\hat{e}_i$ is unity; and all other $\hat{e}_p$ zero, x, y, and z are the coordinates of the node; and x_0, y_0, and z_0 represent the coordinates of the center of rotation. The participation factors are, thus, defined for the translational degrees of freedom and for rotation around the center of rotation.

For coupled acoustic-structural eigenfrequency analyses, an additional acoustic participation factor is computed for each mode as outlined in "Coupled acoustic-structural medium analysis," Section 2.9.1.

Modal effective mass

The effective mass for mode α associated with kinematic direction i, $i = 1, 2, \ldots, 6$) is defined as

$$m_{\alpha i}^{\text{eff}} = (\Gamma_{\alpha i})^2 \, m_\alpha \quad \text{(no sum over } \alpha\text{)}.$$

If the effective masses of all modes are added in any particular direction, the sum should give the total mass of the model, except for mass at kinematically restrained degrees of freedom. Thus, if the effective masses of the modes used in the analysis add up to a value that is significantly less than the model's total mass, this suggests that modes that have significant participation in excitation in that direction have not been extracted.

For coupled acoustic-structural eigenfrequency analyses, an additional acoustic effective mass is computed for each mode as outlined in "Coupled acoustic-structural medium analysis," Section 2.9.1.

Composite modal damping

Abaqus/Standard provides an option to define a composite damping factor for each material. These are assembled into fractions of critical damping values for each mode, $\left(\xi_\alpha\right)_c$, according to

$$\left(\xi_\alpha\right)_c = \frac{1}{m_\alpha} \phi_\alpha^N \left(\sum_a \xi_a M_a^{NM} \right) \phi_\alpha^M \quad \text{(no sum over } \alpha\text{)},$$

where ξ_a is the critical damping fraction given for material a and M_a^{NM} is the part of the structure's mass matrix made up of material a.

References

- "Natural frequency extraction," Section 6.3.5 of the Abaqus Analysis User's Guide
- "Material damping," Section 26.1.1 of the Abaqus Analysis User's Guide

2.5.3 LINEAR DYNAMIC ANALYSIS USING MODAL SUPERPOSITION

Product: Abaqus/Standard

Linear dynamic analysis using modal superposition is computationally inexpensive and can provide useful insight into the dynamic behavior of a system. With modern eigenvalue/eigenvector extraction techniques—such as the subspace method available in Abaqus/Standard—the cost of obtaining a sufficient basis of eigensolutions is not excessive; and the subsequent computational effort involved in obtaining the dynamic response by modal superposition methods is relatively small, especially when compared to the cost of the direct integration methods used for general nonlinear analysis ("Implicit dynamic analysis," Section 2.4.1).

The basic concept of modal superposition is that the response of the structure is expressed in terms of a relatively small number of eigenmodes of the system. The orthogonality of the eigenmodes uncouples this system. Furthermore, only eigenmodes that are close to the frequencies of interest are usually needed; for example, only the lowest few frequencies are usually required to obtain an accurate estimate of a structure's linear dynamic response to relatively long-term loading (for example, its steady-state response to low frequency excitation). The technique can be extended in a limited way into the nonlinear régime, but the superposition and orthogonality principles apply only to purely linear systems: for this reason the methods described in this section are implemented only for linear analysis.

Abaqus/Standard has two "subspace" procedures—one for nonlinear dynamic and the other one for steady-state dynamic analysis—that use some of the eigenmodes of the system on which the equilibrium equations are projected. In both cases the system's eigenmodes are used as a set of global basis vectors for computing the dynamic response, even though the system exhibits nonlinear or frequency-dependent effects during the dynamic response. These methods are cost-effective compared to fully nonlinear dynamic response analysis developed in terms of all the system's degrees of freedom. The subspace projection method for steady-state response is described in "Subspace-based steady-state dynamic analysis," Section 2.6.2. The time-domain subspace projection method is described in "Subspace dynamics," Section 2.4.3.

The procedures provided for modal dynamic analysis of linear systems are summarized below:

a. Modal dynamic time history analysis (see "Modal dynamic analysis," Section 2.5.5).

   This procedure can be used to obtain the time history response of a system to loading conditions that are given as functions of time. The response is integrated through time: the integration method used is exact for loadings that vary piecewise linearly with time. Thus, the only approximations in this analysis procedure are the linearization of the problem, the spatial modeling (that is, the choice of the finite element model), the loading definitions, and the choice of the number of eigenmodes used to represent the system.

b. Response spectrum analysis (see "Response spectrum analysis," Section 2.5.6).

   Response spectrum analysis is often used to obtain an approximate upper bound to the peak significant response of a system to an input spectrum as a function of frequency: it gives the maximum response of a one degree of freedom system as a function of its fundamental frequency of vibration and of its damping ratio. The method has very low computational cost and gives useful information about the spectral behavior of a system with respect to frequency.

c. Steady-state harmonic response analysis (see "Steady-state linear dynamic analysis," Section 2.5.7, and "Subspace-based steady-state dynamic analysis," Section 2.6.2).

This procedure is used when the steady-state response of a system to harmonic excitation is required. The solution is given as the peak amplitudes and phase relationships of the solution variables (stress, displacement, etc.) as functions of frequency: postprocessing options are provided to display such results conveniently.

A similar option is provided for direct harmonic response analysis without using the eigenmodes as a basis. The direct method is significantly more expensive computationally than the modal method: it is needed if the system is nonsymmetric (because Abaqus presently does not have a nonsymmetric eigenvalue extraction capability) or if the system's behavior includes frequency-dependent parameters.

The "subspace" method is typically less expensive than the direct method. It is generally used for nonsymmetric systems or when the system's behavior includes frequency-dependent parameters or discrete damping.

d. Random response analysis (see "Random response analysis," Section 2.5.8).

This procedure is used when the structure is excited continuously and the loading can be expressed statistically in terms of a "Power Spectral Density Function." The response is calculated in terms of statistical quantities, such as the mean value and the standard deviation of nodal and element variables.

Reference

- "Dynamic analysis procedures: overview," Section 6.3.1 of the Abaqus Analysis User's Guide

2.5.4 DAMPING OPTIONS FOR MODAL DYNAMICS

Product: Abaqus/Standard

For linear dynamic analysis based on modal superposition, several options are provided in Abaqus/Standard to introduce damping, as follows:

Critical damping factors

The damping in each eigenmode can be given as a fraction of the critical damping for that mode.

The equation of motion for a one degree of freedom system (one of the eigenmodes of the system) is

$$m\ddot{q} + c\dot{q} + kq = 0,$$

where m is the mass, c the damping, k the stiffness, and q the modal amplitude.

The solution is of the form

$$q = A \exp \lambda t,$$

where A is a constant, and

$$\lambda = \frac{-c}{2m} \pm \sqrt{\frac{c^2}{4m^2} - \frac{k}{m}}.$$

The solution will be oscillatory if the expression under the root sign is negative. Critical damping is defined as the damping that makes this expression zero:

$$c_{cr} = 2\sqrt{mk}.$$

If the system is critically damped, after any disturbance the system will return to a static equilibrium state as rapidly as possibly without any oscillation.

Typically, when damping is given as a fraction of critical damping associated with each mode, the values used are in the range of 1% to 10% of critical damping. This method of introducing damping has no physical basis in the finite element model: it is a purely mathematical concept introduced in association with the eigenmodes of the system. Thus, the concept cannot be extended to nonlinear applications where the equations of motion of the system are integrated directly and where the natural frequencies of the system are constantly changing because of nonlinearities.

Rayleigh damping

Rayleigh damping is defined by a damping matrix formed as a linear combination of the mass and the stiffness matrices:

$$C^{MN} = \alpha M^{MN} + \beta K^{MN}.$$

With Rayleigh damping the eigenvectors of the damped system are the same as the eigenvectors of the undamped system. Rayleigh damping can, therefore, be converted into critical damping fractions for each mode: this is the way Rayleigh damping is handled in Abaqus/Standard.

A form of Rayleigh damping is also provided in Abaqus for nonlinear analysis. When the problem is nonlinear the mass damping factor can be used directly: the stiffness damping factor is interpreted as creating viscoelastic behavior in which the viscosity is proportional to the elasticity, which gives exactly the stiffness proportional damping effect defined above for the linear case.

Composite modal damping

When composite modal damping is used, a damping value is defined for each material as a fraction of critical damping to be associated with that material. These values are converted into a weighted average for each eigenmode, weighted by the mass matrix according to the equation

$$\xi_\alpha = \frac{1}{m_\alpha} \phi_\alpha^M \xi_m M_m^{MN} \phi_\alpha^N \quad \text{(no sum over } \alpha\text{)},$$

where ξ_α is the critical damping ratio used in mode α; ξ_m is the critical damping fraction defined for material m; M_m^{MN} is the mass matrix associated with material m; ϕ_α^M is the eigenvector of the αth mode; and m_α is the generalized mass associated with the αth mode ($m_\alpha = \phi_\alpha^M M^{MN} \phi_\alpha^N$, no sum over α).

Structural damping

Structural damping assumes that the damping forces are proportional to the forces caused by stressing of the structure and are opposed to the velocity. This form of damping can be used only if the displacement and velocity are exactly 90° out of phase, which is the case when the excitation is sinusoidal, so structural damping can be used only in steady-state and random response analysis. The damping forces are then

$$F_D^N = isI^N,$$

where I^N are the forces caused by stressing of the structure, F_D^N are the damping forces, s is the structural damping factor, and $i = \sqrt{-1}$.

Any combination of damping options can be used in an analysis: the effects will be added if several damping definitions are chosen.

Reference

- "Material damping," Section 26.1.1 of the Abaqus Analysis User's Guide

2.5.5 MODAL DYNAMIC ANALYSIS

Product: Abaqus/Standard

The modal dynamic procedure provides time history analysis of linear systems. The excitation is given as a function of time: it is assumed that the amplitude curve is specified so that the magnitude of the excitation varies linearly within each increment. When the model is projected onto the eigenmodes used for its dynamic representation, we obtain the following set of equations at time t:

$$\ddot{q}_\beta + C_{\beta\alpha}\dot{q}_\alpha + \omega_\beta^2 q_\beta = (f_t)_\beta = f_{t-\Delta t} + \frac{\Delta f}{\Delta t}\Delta t, \tag{2.5.5–1}$$

where the α and β indices span the eigenspace; $C_{\beta\alpha}$ is the projected viscous damping matix; q_β is the "generalized coordinate" of the mode β (the amplitude of the response in this mode); $\omega_\beta = \sqrt{k_\beta/m_\beta}$ is the natural frequency of the undamped mode β (obtained as the square root of the eigenvalue in the eigenfrequency step that precedes the modal dynamic time history analysis); $(f_t)_\beta$ is the magnitude of the loading projected onto this mode (the "generalized load" for the mode); and Δf is the change in f over the time increment, which is Δt.

If the projected damping matrix is diagonal, Equation 2.5.5–1 becomes the following uncoupled set of equations:

$$\ddot{q}_\beta + 2\xi_\beta\omega_\beta\dot{q}_\beta + \omega_\beta^2 q_\beta = (f_t)_\beta, \tag{2.5.5–2}$$

where ξ_β is the critical damping ratio given by the relation

$$2\xi_\beta\omega_\beta = \frac{c_\beta}{m_\beta},$$

where c_β is the modal viscous damping coefficient and m_β is the modal mass in mode β.

Solution to the uncoupled system

The solution to the uncoupled equations is obtained readily as a particular integral for the loading and a solution to the homogeneous equation (with no right-hand side). These solutions can be combined and written in the general form

$$\begin{Bmatrix} q_{t+\Delta t} \\ \dot{q}_{t+\Delta t} \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{Bmatrix} q_t \\ \dot{q}_t \end{Bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{Bmatrix} f_t \\ f_{t+\Delta t} \end{Bmatrix}, \tag{2.5.5–3}$$

where a_{ij} and b_{ij}, $i, j = 1, 2$ are constants, since we have assumed that the loading only varies linearly over the time increment (that is, $\Delta f/\Delta t$ is constant).

There are three cases of this solution for nonrigid body motion ($\omega \neq 0$), depending on whether the damping in the modal equilibrium equation is greater than, equal to, or less than critical damping (that is, depending on whether $(\xi^2 - 1)$ is positive, zero, or negative).

Solution to the coupled system

Equation 2.5.5–3 can be generalized to address full coupling in the projected damping matrix. Let the matrix $\mathbf{C}$ be split into its diagonal and off-diagonal parts so that

$$\mathbf{C} = \mathbf{C}_{diag} + \mathbf{C}_{off}.$$

Then, with the additional assumption that the off-diagonal damping force varies linearly over a time increment, the equation for the uncoupled system can be rewritten as

$$\begin{Bmatrix} q_{t+\Delta t} \\ \bar{\mathbf{C}}\dot{q}_{t+\Delta t} \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} - b_{11}\mathbf{C}_{off} \\ a_{21} & a_{22} - b_{21}\mathbf{C}_{off} \end{bmatrix} \begin{Bmatrix} q_t \\ \dot{q}_t \end{Bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{Bmatrix} f_t \\ f_{t+\Delta t} \end{Bmatrix} + \begin{bmatrix} 0 & -b_{12}\mathbf{C}_{off} \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} q_{t+\Delta t} \\ \dot{q}_{t+\Delta t} \end{Bmatrix}, \tag{2.5.5–4}$$

where $\bar{\mathbf{C}}$ is given by

$$\bar{\mathbf{C}} = I + b_{22}\mathbf{C}_{off}.$$

$\bar{\mathbf{C}}$ is fully populated but is a function only of Δt, so it needs to be factored only a single time for a given analysis.

Integral coefficients

There are three cases of this solution for nonrigid body motion ($\omega \neq 0$), depending on whether the damping in the modal equilibrium equation is greater than, equal to, or less than critical damping (that is, depending on whether $(\xi^2 - 1)$ is positive, zero, or negative).

Damping less than critical

This case is the most common. With

$$\bar{\omega} = \omega\sqrt{1 - \xi^2},$$

we have

$$\begin{aligned} a_{11} &= \exp(-\xi\omega\Delta t)\left(\xi\frac{\omega}{\bar{\omega}}\sin\bar{\omega}\Delta t + \cos\bar{\omega}\Delta t\right), \\ a_{12} &= \exp(-\xi\omega\Delta t)\frac{1}{\bar{\omega}}\sin\bar{\omega}\Delta t, \\ a_{21} &= -\exp(-\xi\omega\Delta t)\frac{\omega}{\sqrt{1-\xi^2}}\sin\bar{\omega}\Delta t, \\ a_{22} &= \exp(-\xi\omega\Delta t)\left(\cos\bar{\omega}\Delta t - \frac{\xi\omega}{\bar{\omega}}\sin\bar{\omega}\Delta t\right), \end{aligned}$$

$$
\begin{aligned}
b_{11} &= -\exp(-\xi\omega\Delta t)\left\{\left(\frac{\xi}{\omega\bar{\omega}}+\frac{2\xi^2-1}{\omega^2\bar{\omega}\Delta t}\right)\sin\bar{\omega}\Delta t+\left(\frac{1}{\omega^2}+\frac{2\xi}{\omega^3\Delta t}\right)\cos\bar{\omega}\Delta t\right\}+\frac{2\xi}{\omega^3\Delta t},\\
b_{12} &= \exp(-\xi\omega\Delta t)\left\{\frac{2\xi^2-1}{\omega^2\bar{\omega}\Delta t}\sin\bar{\omega}\Delta t+\frac{2\xi}{\omega^3\Delta t}\cos\bar{\omega}\Delta t\right\}+\frac{1}{\omega^2}-\frac{2\xi}{\omega^3\Delta t},\\
b_{21} &= -\exp(-\xi\omega\Delta t)\left\{(\bar{\omega}\cos\bar{\omega}\Delta t-\xi\omega\sin\bar{\omega}\Delta t)\left(\frac{2\xi^2-1}{\omega^2\bar{\omega}\Delta t}+\frac{\xi}{\bar{\omega}\omega}\right)\right.\\
&\quad\left.-(\bar{\omega}\sin\bar{\omega}\Delta t+\xi\omega\cos\bar{\omega}\Delta t)\left(\frac{1}{\omega^2}+\frac{2\xi}{\omega^3\Delta t}\right)\right\}-\frac{1}{\omega^2\Delta t},\\
b_{22} &= -\exp(-\xi\omega\Delta t)\left\{-(\bar{\omega}\cos\bar{\omega}\Delta t-\xi\omega\sin\bar{\omega}\Delta t)\frac{2\xi^2-1}{\bar{\omega}\omega^2\Delta t}+(\bar{\omega}\sin\bar{\omega}\Delta t+\xi\omega\cos\bar{\omega}\Delta t)\frac{2\xi}{\omega^3\Delta t}\right\}\\
&\quad+\frac{1}{\omega^2\Delta t}.
\end{aligned}
$$

Damping equal to critical

In this case

$$
\begin{aligned}
a_{11} &= \exp(-\xi\omega\Delta t)(1+\xi\omega\Delta t),\\
a_{12} &= \exp(-\xi\omega\Delta t)\,\Delta t,\\
a_{21} &= -\exp(-\xi\omega\Delta t)(\xi\omega)^2\Delta t,\\
a_{22} &= \exp(-\xi\omega\Delta t)(1-\xi\omega\Delta t),
\end{aligned}
$$

$$
\begin{aligned}
b_{11} &= -\exp(-\xi\omega\Delta t)\left\{-(1+\xi\omega\Delta t)\left(\frac{1}{\omega^2}+\frac{2\xi}{\omega^3\Delta t}\right)+\frac{1}{\omega^2}\right\},\\
b_{12} &= \exp(-\xi\omega\Delta t)\left\{(1+\xi\omega\Delta t)\left(\frac{2\xi}{\omega^3\Delta t}-\frac{1}{\omega^2}\right)+\frac{1}{\omega^2}\right\}-\frac{2\xi}{\omega^3\Delta t},\\
b_{21} &= -\exp(-\xi\omega\Delta t)\left\{(\xi\omega)^2\Delta t\left(\frac{1}{\omega^2}+\frac{2\xi}{\omega^3\Delta t}\right)+\frac{1-\xi\omega\Delta t}{\omega^2\Delta t}\right\}-\frac{1}{\Delta t\omega^2},\\
b_{22} &= -\exp(-\xi\omega\Delta t)\left\{-(\xi\omega)^2\Delta t\frac{2\xi}{\omega^3\Delta t}-\frac{1-\xi\omega\Delta t}{\Delta t\omega^2}\right\}+\frac{1}{\omega^2\Delta t}.
\end{aligned}
$$

Damping higher than critical

In this case, with

$$
\bar{\omega}=\omega\sqrt{\xi^2-1},
$$

we have

$$
\begin{aligned}
a_{11} &= \exp\left(-\xi\omega\Delta t\right)\left(\xi\frac{\omega}{\bar{\omega}}\sinh\bar{\omega}\Delta t + \cosh\bar{\omega}\Delta t\right),\\
a_{12} &= \exp\left(-\xi\omega\Delta t\right)\frac{1}{\bar{\omega}}\sinh\bar{\omega}\Delta t,\\
a_{21} &= -\exp\left(-\xi\omega\Delta t\right)\frac{\omega}{\sqrt{\xi^2-1}}\sinh\bar{\omega}\Delta t,\\
a_{22} &= \exp\left(-\xi\omega\Delta t\right)\left(\cosh\bar{\omega}\Delta t - \frac{\xi\omega}{\bar{\omega}}\sinh\bar{\omega}\Delta t\right),
\end{aligned}
$$

$$
\begin{aligned}
b_{11} &= -\exp\left(-\xi\omega\Delta t\right)\left\{\left(\frac{\xi}{\omega\bar{\omega}} + \frac{2\xi^2-1}{\omega^2\bar{\omega}\Delta t}\right)\sinh\bar{\omega}\Delta t + \left(\frac{1}{\omega^2} + \frac{2\xi}{\omega^3\Delta t}\right)\cosh\bar{\omega}\Delta t\right\} + \frac{2\xi}{\omega^3\Delta t},\\
b_{12} &= \exp\left(-\xi\omega\Delta t\right)\left\{\frac{2\xi^2-1}{\omega^2\bar{\omega}\Delta t}\sinh\bar{\omega}\Delta t + \frac{2\xi}{\omega^3\Delta t}\cosh\bar{\omega}\Delta t\right\} + \frac{1}{\omega^2} - \frac{2\xi}{\omega^3\Delta t},
\end{aligned}
$$

$$
\begin{aligned}
b_{21} = &-\exp\left(-\xi\omega\Delta t\right)\left\{\left(\bar{\omega}\cosh\bar{\omega}\Delta t - \xi\omega\sinh\bar{\omega}\Delta t\right)\left(\frac{2\xi^2-1}{\omega^2\bar{\omega}\Delta t} + \frac{\xi}{\bar{\omega}\omega}\right)\right.\\
&\left. - \left(-\bar{\omega}\sinh\bar{\omega}\Delta t + \xi\omega\cosh\bar{\omega}\Delta t\right)\left(\frac{1}{\omega^2} + \frac{2\xi}{\omega^3\Delta t}\right)\right\} - \frac{1}{\omega^2\Delta t},\\
b_{22} = &-\exp\left(-\xi\omega\Delta t\right)\left\{-\left(\bar{\omega}\cosh\bar{\omega}\Delta t - \xi\omega\sinh\bar{\omega}\Delta t\right)\frac{2\xi^2-1}{\bar{\omega}\omega^2\Delta t}\right.\\
&\left. + \left(-\bar{\omega}\sinh\bar{\omega}\Delta t + \xi\omega\cosh\bar{\omega}\Delta t\right)\frac{2\xi}{\omega^3\Delta t}\right\} + \frac{1}{\omega^2\Delta t}.
\end{aligned}
$$

Rigid body mode with damping

If there are rigid body modes in the finite element model, there will be one or several eigenvalues that are zero. The equation of motion (Equation 2.5.5–1) is reduced to

$$\ddot{q} + \alpha\dot{q} = f_{t-\Delta t} + \frac{\Delta f}{\Delta t}\Delta t. \tag{2.5.5–5}$$

Only Rayleigh damping can be specified for a rigid body mode, since the critical damping is zero. Furthermore, since it is a rigid body mode, only the mass damping factor, α, appears (stiffness damping requires that there be straining of the body). For this case

$$
\begin{aligned}
a_{11} &= 1,\\
a_{12} &= \frac{1}{\alpha}\left(1 - \exp\left(-\alpha\Delta t\right)\right),\\
a_{21} &= 0,\\
a_{22} &= \exp\left(-\alpha\Delta t\right),
\end{aligned}
$$

$$
\begin{aligned}
b_{11} &= -\frac{1}{\alpha^2}\left(1-\exp\left(-\alpha\Delta t\right)\right)\left(1+\frac{1}{\alpha\Delta t}\right)+\Delta t\left(1+\frac{1}{\alpha\Delta t}\right)-\frac{\Delta t}{2\alpha},\\
b_{12} &= \frac{1}{\alpha}\left(1-\exp\left(-\alpha\Delta t\right)\right)\frac{1}{\alpha^2\Delta t}-\frac{1}{\alpha^2}+\frac{\Delta t}{2\alpha},\\
b_{21} &= \frac{1}{\alpha}\left(1+\frac{1}{\alpha\Delta t}\right)\left(1-\exp\left(-\alpha\Delta t\right)\right)-\frac{1}{\alpha},\\
b_{22} &= -\frac{1}{\alpha^2\Delta t}\left(1-\exp\left(-\alpha\Delta t\right)\right)+\frac{1}{\alpha}.
\end{aligned}
$$

Rigid body mode without damping

For the particular case of a rigid body mode without damping, the equation of motion (Equation 2.5.5–1) is reduced to

$$\ddot{q} = f_{t-\Delta t} + \frac{\Delta f}{\Delta t}\Delta t. \tag{2.5.5–6}$$

For this case

$$
\begin{aligned}
a_{11} &= 1,\\
a_{12} &= \Delta t,\\
a_{21} &= 0,\\
a_{22} &= 1,\\
b_{11} &= \frac{\Delta t^2}{3},\\
b_{12} &= \frac{\Delta t^2}{6},\\
b_{21} &= \frac{\Delta t}{2},\\
b_{22} &= \frac{\Delta t}{2}.
\end{aligned}
$$

Structural damping in modal dynamic analysis

Structural damping is a commonly used damping model that represents damping as complex stiffness. This representation causes no difficulty for frequency domain analysis such as steady-state dynamics for which the solution is already complex. However, in the time domain, the solution must remain real-valued. To allow users to apply their structural damping model in the time domain, a method has been developed to convert structural damping to an equivalent viscous damping. This technique was designed so that, in the frequency domain, the viscous damping applied is identical to structural damping if the projected damping matrix is diagonal.

We start with the equation of a single degree-of-freedom oscillator,

$$m\ddot{x} + c\dot{x} + k(1 + i\gamma)x = f,$$

where $\boldsymbol{m}$ is the mass, $\boldsymbol{c}$ is the viscous damping factor, $\boldsymbol{k}$ is the stiffness, $\boldsymbol{x}$ is the response, $\boldsymbol{f}$ is the force, and γ is the structural damping factor. If we normalize by the mass and assume harmonic input, we obtain the relation

$$(\omega^2(1 + i\gamma) - \Omega^2 + i2\xi\omega\Omega)x_0 = f_0,$$

where ω is the natural frequency, Ω is the excitation frequency, ξ is the coefficient of the harmonic response, and f_0 is the coefficient of the harmonic input. The relation between the viscous damping factor $\boldsymbol{c}$ and the critical damping factor ξ is such that $c = 2\xi\omega m$.

Based on the two previous equations, if we want the viscous damping factor to have the same effect as the structural damping factor, we must have

$$\xi = \frac{\omega\gamma}{2\Omega}.$$

If we further assume that $\omega = \Omega$, the relation simplifies to

$$\xi = \frac{\gamma}{2}.$$

Now we consider the finite element system of equations,

$$M^{NM}\ddot{x}^M + C^{NM}\dot{x}^M + (K^{NM} + iK_s^{NM})x^M = f^N,$$

where C^{NM} is the finite element viscous damping matrix and K_s^{NM} is the finite element structural damping matrix. If we project these equations onto the eigenmodes, the damping matrices become

$$\hat{C} = \Phi^N C^{NM} \Phi^M \qquad \text{and} \qquad \hat{K}_s = \Phi^N K_s^{NM} \Phi^M,$$

which could be fully populated. If we take the case that C^{NM} and K_s^{NM} are diagonal, their diagonal elements can be written

$$\hat{c}_\alpha = 2\xi_\alpha\omega_\alpha \qquad \text{and} \qquad \hat{k}_{s\alpha} = \gamma_\alpha\omega_\alpha^2,$$

where α is a particular mode. Maintaining the relation developed for the single degree-of-freedom system $\xi = \frac{\gamma}{2}$ for this diagonal case would require

$$(\hat{c}_{equivalent})_\alpha = \frac{\hat{k}_{s\alpha}}{\omega_\alpha}.$$

For nondiagonal matrices the expression for equivalent viscous damping becomes

$$\hat{C}_{equivalent} = D\hat{K}_s D,$$

where D is a diagonal matrix whose entries are given by

$$d_\alpha = \frac{1}{\sqrt{\omega_\alpha}}.$$

Response of nodal and element variables

The time integration is done in terms of the generalized coordinates, and the response of the physical variables is then immediately available by summation:

$$u = \sum_\alpha \phi_\alpha q_\alpha,$$

$$\varepsilon = \sum_\alpha \varepsilon_\alpha q_\alpha,$$

$$\sigma = \sum_\alpha \sigma_\alpha q_\alpha,$$

$$R = \sum_\alpha R_\alpha q_\alpha,$$

where ϕ_α are the modes, ε_α are the modal strain amplitudes, σ_α are the modal stress amplitudes, and R_α are the modal reaction force amplitudes corresponding to each eigenvector α.

Initial conditions

At the beginning of the step initial displacements and initial velocities must be converted to equivalent values of the generalized coordinates, which can be done exactly only if the number of eigenvectors equals the number of degrees of freedom. Since this is usually not the case, the initial values of the generalized coordinate displacement and velocity are calculated as

$$q_\alpha = \frac{1}{m_\alpha}\phi_\alpha^M M^{MN} x_0^N,$$

where m_α is the generalized mass for the eigenvector α, ϕ_α^M is the eigenvector, M^{MN} is the mass matrix, and x_0^N are the initial displacements.

Similarly, for the initial velocities

$$\dot{q}_\alpha = \frac{1}{m_\alpha}\phi_\alpha^M M^{MN} \dot{x}_0^N.$$

For the case where the initial conditions are given by a previous modal dynamic analysis, the generalized displacement, velocity, and acceleration are simply taken from the previous analysis.

Base motion definition

Many linear dynamic problems involve finding the response of a structure to a "base motion": a time history of displacement, velocity, or acceleration given for the points where the displacements of the structure are prescribed. In all cases these base motions are converted into an acceleration history. If the displacement or velocity history has nonzero values at time $t = 0$, corrections to the acceleration histories are made at times $t = 0$ and $t = \Delta t$. If velocities are given, the acceleration at $t = 0$ is

$$\ddot{z}_0 = \frac{\dot{z}^*_{\Delta t} - \dot{z}^*_0}{\Delta t}.$$

If displacements are given, the accelerations at $t = 0$ and $t = \Delta t$ are

$$\ddot{z}_0 = \frac{3z^*_{\Delta t}}{\Delta t^2} \quad \text{and} \quad \ddot{z}_{\Delta t} = \frac{z^*_{2\Delta t} - \frac{5}{2}z^*_{\Delta t}}{\Delta t^2}.$$

In the above expressions and in the expressions to follow, a superscript * indicates user-defined amplitude data.

If velocities are given for tabular or equally spaced amplitude curve definitions, the acceleration is defined by the central difference rule

$$\ddot{z}_{t+\Delta t} = \frac{\dot{z}^*_{t+2\Delta t} - \dot{z}^*_t}{2\Delta t}.$$

If displacements are given, the acceleration is defined by the central difference rule

$$\ddot{z}_{t+\Delta t} = \frac{z^*_{t+2\Delta t} - 2z^*_{t+\Delta t} + z^*_t}{\Delta t^2}.$$

The response is calculated relative to the base. If total values of nodal variables are required, the motion at the base is added to the relative values:

$$\begin{aligned} \ddot{u}_{\text{abs}} &= \ddot{z} + \ddot{x}, \\ \dot{u}_{\text{abs}} &= \dot{z} + \dot{x}, \\ u_{\text{abs}} &= z + x, \end{aligned}$$

where

$$\begin{aligned} \dot{z}_{t+\Delta t} &= \dot{z}_t + \frac{\Delta t}{2}\left(\ddot{z}_t + \ddot{z}_{t+\Delta t}\right), \\ z_{t+\Delta t} &= z_t + \dot{z}_t \Delta t + \frac{\Delta t^2}{3}\left(\ddot{z}_t + \frac{1}{2}\ddot{z}_{t+\Delta t}\right). \end{aligned}$$

Reference

- "Transient modal dynamic analysis," Section 6.3.7 of the Abaqus Analysis User's Guide

2.5.6 RESPONSE SPECTRUM ANALYSIS

Product: Abaqus/Standard

Response spectrum analysis is intended to provide an inexpensive approach to estimating the peak response of a model (usually a model of a structure) subjected to "base motion": the simultaneous motion of all nodes fixed with boundary conditions. The approach assumes that the system's response is linear, so it can be analyzed in the frequency domain using its lowest eigenmodes—Φ_α—and eigenfrequencies ω_α—extracted in a previous eigenfrequency step. The method is typically used to estimate the response of a building or of a piping system in a building to an earthquake. The method is not appropriate if the excitation is so severe that nonlinear effects in the system are important. In such a case the time history of the base excitation must be known and used with a dynamic analysis step to obtain the system's response.

Even for a linear system the response spectrum method provides only estimates of the peak response. If more precise values are required, a modal dynamic analysis step can be used to integrate the system through time and, thus, develop its response to the given base excitation.

In response spectrum analysis the estimates of peak values are obtained by combining the peak responses of the participating modes corresponding to user-specified spectra definitions. Several approximations are introduced by response spectrum analysis. The conversion from a time history of excitation into an equivalent frequency domain spectrum is based on the behavior of a single degree of freedom system. Different spectra are often applied in different excitation directions. Once the spectra are known, the peak modal responses can be calculated. The manner in which these peak modal responses are combined to estimate the peak physical response, together with the manner in which multidirectional excitations are combined, introduces approximations. Since no one method gives good approximations for all cases, several methods are offered. These methods are discussed in the Regulatory Guide 1.92 (1976) of the U.S. Nuclear Regulatory Commission, in the papers by Anagnastopoulos (1981), Der Kiureghian (1981), and Smeby (1984) and in the book by A. K. Gupta (1990). The choice of the summation rules depends on the particular case and is a matter of the user's judgment.

Since response spectrum analysis is commonly used as a basic design tool, spectra are defined in many design codes for such applications as seismic analysis of buildings. In such cases the user works from the given spectra. In other cases the time history of a known base excitation must first be converted into a response spectrum by considering the response of a single degree of freedom system excited by the known base motion. For this purpose the single degree of freedom system is characterized by its undamped natural frequency, ω, and the fraction of critical damping present in the system, ξ. The equation of motion of the system is integrated through time to find peak values of relative displacement, relative velocity, and absolute acceleration. The integration described in "Modal dynamic analysis," Section 2.5.5, can be used for this purpose, since it is exact when the base motion record varies linearly with time. Thus, the maximum values of displacement, velocity, and acceleration are found for the linear, one degree of freedom system. This process is repeated for all frequency and damping values in the range of interest to construct displacement, velocity, and acceleration spectra, $S^D(\omega, \xi)$, $S^V(\omega, \xi)$, and $S^A(\omega, \xi)$. A FORTRAN program to build spectra in this way from an acceleration record is given in "Analysis of a cantilever subject to earthquake motion," Section 1.4.13 of the Abaqus Benchmarks Guide (file cantilever_spectradata.f.)

If there is no damping, the relationship between S^D, S^V, and S^A is given by

$$S^D = \frac{1}{\omega} S^V = \frac{1}{\omega^2} S^A.$$

In Abaqus/Standard it is assumed that the damping is always small, so these relationships are used whenever a conversion is needed.

A response spectrum is defined by giving a table of values of S at increasing values of frequency, ω, for increasing values of damping, ξ. Linear interpolation on a logarithmic scale is used to compute the response for any required frequency and damping factor. Any number of spectra can be defined.

The response spectrum procedure allows up to three spectra, which we denote by k, $k = 1, 2, 3$, to be applied to the model in orthogonal physical directions defined by their direction cosines, $\mathbf{t}^k$. These spectra can come from different excitations (with a certain level of correlation between them), or they can be components of a single base excitation acting in an arbitrary direction.

When modal methods are used to define a model's response, the value of any physical variable is defined from the amplitudes of the modal responses (the "generalized coordinates"), q_α. The first stage in the response spectrum procedure is to estimate the peak values of these modal responses. For mode α and spectrum k this is

$$(q_\alpha^{\max})_k = c_k S_k^D \sum_j t_j^k \Gamma_{\alpha j},$$

where c_k is a user-defined scaling parameter, $S_k^D(\omega, \xi)$ is the kth displacement spectrum, t_j^k is the jth direction cosine for the kth spectrum, and $\Gamma_{\alpha j}$ is the participation factor for mode α in direction j (see "Variables associated with the natural modes of a model," Section 2.5.2, for the definition of $\Gamma_{\alpha j}$). Similar expressions for $(\dot{q}_\alpha^{\max})_k$ and $(\ddot{q}_\alpha^{\max})_k$ are obtained by using velocity or acceleration spectra in the above formula.

We now have estimates of the peak responses of the "generalized coordinates"—the amplitudes of the responses of the natural modes of the system for excitation in each direction. If the input spectra in the different directions are components of a single base excitation acting in an arbitrary direction, for each mode we can combine these peak responses into a single value by specifying algebraic summation of the values for the different spatial directions:

$$q_\alpha^{\max} = \sum_k (q_\alpha^{\max})_k.$$

In this case the modal combinations discussed below still apply, but the subscript k is no longer relevant and should be ignored.

Let us denote by $\left(R_\alpha^i(\omega, \xi)\right)_k^{\max}$ the peak response of some physical variable R^i (a component of displacement, stress, section force, reaction force, etc.) caused by motion in the natural mode α excited by the response spectrum in excitation direction k at frequency ω and with damping ξ. Denote the component of the eigenvector α associated with R_α^i by Φ_α^i. Then

$$(R_\alpha^i)_k^{\max} = \Phi_\alpha^i (q_\alpha^{\max})_k,$$

$$(\dot{R}^i_\alpha)^{\max}_k = \Phi^i_\alpha(\dot{q}^{\max}_\alpha)_k,$$

and

$$(\ddot{R}^i_\alpha)^{\max}_k = \Phi^i_\alpha(\ddot{q}^{\max}_\alpha)_k.$$

We need to combine these estimates of the peak physical responses in the individual modes into estimates of the total peak response of the particular physical variable to the given spectrum, $\left(R^i(\omega,\xi)\right)^{\max}_k$. Since the peak responses in the different modes will not in general occur at the same time, this combination is only an estimate, so several formulæ are offered, as follows:

Summation of the absolute values of the modal peak responses estimates

$$(R^i)^{\max}_k = \sum_\alpha \left|(R^i_\alpha)^{\max}_k\right|.$$

This provides the most conservative estimate of the peak response, since it assumes that all modes provide peak response in phase at the same time.

Square root of the summation of the squares (SRSS) estimates

$$(R^i)^{\max}_k = \sqrt{\sum_\alpha \left((R^i_\alpha)^{\max}_k\right)^2}.$$

This summation usually provides a reasonable estimate if the natural frequencies of the modes are well separated.

The Naval Research Laboratory method distinguishes the mode, β, in which the physical variable has its maximum response, and adds the square root of the sum of squares of the peak responses in all other modes to the absolute value of the peak response of that mode. This gives the estimate

$$(R^i)^{\max}_k = \left|(R^i_\beta)^{\max}_k\right| + \sqrt{\sum_{\alpha\neq\beta} \left((R^i_\alpha)^{\max}_k\right)^2}.$$

Again, the modes must be reasonably well-spaced in the frequency domain to obtain an accurate estimate with this method.

A variety of methods are available that aim to improve the estimation for structures with closely spaced frequencies. Abaqus/Standard provides two of them: the Ten Percent Method recommended by Regulatory Guide 1.92 (1976) of the U.S. Nuclear Regulatory Commission and the Complete Quadratic Combination Method, which was first introduced by Der Kiureghian (1981) and developed by Smeby and Der Kiureghian (1984). Both methods reduce to the SRSS method if the modes are well separated with no coupling among them.

Two additional methods, the Grouping Method and the Double Sum Combination Method, are also available to satisfy Regulatory Guide 1.92.

Ten Percent Method

The Ten Percent Method described in Regulatory Guide 1.92 modifies the SRSS Method by adding a contribution from all pairs of modes α and β whose frequencies are within 10% of each other, giving the estimate

$$(R^i)_k^{\max} = \sqrt{\sum_\alpha \left((R^i_\alpha)_k^{\max}\right)^2 + 2\sum_{\alpha<\beta} \left|(R^i_\alpha)_k^{\max}(R^i_\beta)_k^{\max}\right|}.$$

The frequencies of modes α and β are considered to be within 10% whenever

$$\frac{\omega_\beta - \omega_\alpha}{\omega_\beta} \le 0.1, \quad \alpha < \beta.$$

Complete Quadratic Combination Method

The Complete Quadratic Combination Method (CQC) combines the modal response with the formula

$$(R^i)_k^{\max} = \sqrt{\sum_\alpha \sum_\beta (R^i_\alpha)_k^{\max} \rho_{\alpha\beta} (R^i_\beta)_k^{\max}},$$

where $\rho_{\alpha\beta}$ are cross-correlation coefficients between modes α and β, which depend on the ratio of frequencies and modal damping between the two modes:

$$\rho_{\alpha\beta} = \frac{8\sqrt{\xi_\alpha \xi_\beta}\left(\xi_\alpha + r_{\beta\alpha}\xi_\beta\right) r_{\beta\alpha}^{\frac{3}{2}}}{\left(1 - r_{\beta\alpha}^2\right)^2 + 4\xi_\alpha\xi_\beta r_{\beta\alpha}\left(1 + r_{\beta\alpha}^2\right) + 4\left(\xi_\alpha^2 + \xi_\beta^2\right) r_{\beta\alpha}^2},$$

where

$$r_{\beta\alpha} = \frac{\omega_\beta}{\omega_\alpha}.$$

If double eigenvalues occur with the same damping coefficient, their correlation coefficient will be $\rho_{\alpha\beta} = 1$. If modes are well-spaced, their cross-correlation coefficient will be small ($\rho_{\alpha\beta} << 1$) and the method will give the same results as the SRSS method. This method is usually recommended for asymmetrical building systems, since, in such cases, other methods can underestimate the response in the direction of motion and overestimate the response in the transverse direction (see "Response spectra of a three-dimensional frame building," Section 2.2.3 of the Abaqus Example Problems Guide).

For the case of different base excitations acting along orthogonal directions, we still need to sum over the directions indicated by the subscript k. Regulatory Guide 1.92 specifies that this directional summation be based on the square root of the sum of the squares summation rule:

$$(R^i)^{\max} = \sqrt{\sum_k \left((R^i)_k^{\max}\right)^2}.$$

This rule is appropriate when the base motions in different directions are statistically independent (uncorrelated)—when they are acting along "principal directions." The existence of a set of directions along which the ground motions can be considered uncorrelated is discussed by Penzien and Watabe (1975). Such considerations are especially important when the CQC modal combination method is used. In the Abaqus implementation of the CQC method it is assumed that the horizontal components act along the principal directions and are of equal intensity. For details on the CQC method applied to more general cases, see Smeby and Der Kiureghian (1984).

Grouping Method

The Grouping Method, also known as the NRC Grouping Method, improves the response estimation for structures with closely spaced eigenvalues. The modal responses are grouped such that the lowest and highest frequency modes in a group are within 10% and no mode is in more than one group. The modal responses are summed absolutely within groups before performing a SRSS combination of the groups. Within the group responses are summed as

$$(R^i_{gr})^{max}_k = |(R^i_1)^{max}_k| + |(R^i_2)^{max}_k| ... + |(R^i_n)^{max}_k|,$$

for n frequencies within any gr group and then performing:

$$(R^i)^{max}_k = \sqrt{\sum_{gr=1}^{gr=n} \left((R^i_{gr})^{\max}_k\right)^2}.$$

The above expression includes all the groups, and the group can consist of just one frequency response if the frequency does not have another member that is within the 10% limit. The Ten Percent Method, by comparison, will always produce results higher in value than the grouping method.

Double Sum Combination Method

The Double Sum Combination Method, also known as Rosenblueth's Double Sum Combination, is the first attempt to evaluate modal correlation based on random vibration theory. The method utilizes the time duration t_D of strong earthquake motion. The mode correlation coefficients $c_{\alpha\beta}$ that also depend on frequencies and damping coefficient ξ lead to the following mode combination:

$$(R^i)^{max}_k = \sqrt{\sum_\alpha \sum_\beta (R^i_\alpha)^{max}_k c_{\alpha\beta} (R^i_\beta)^{max}_k},$$

where

$$c_{\alpha\beta} = \frac{1}{1 + \left(\frac{\omega_{\alpha'} - \omega_{\beta'}}{\xi_{\alpha'}\omega_\alpha + \xi_{\beta'}\omega_\beta}\right)^2},$$

where

$$\omega_{\alpha'} = \omega_\alpha \sqrt{1 - \xi_\alpha^2},$$

$$\xi_{\alpha'} = \xi_\alpha + \frac{2}{t_D \omega_\alpha}.$$

Reference

- "Response spectrum analysis," Section 6.3.10 of the Abaqus Analysis User's Guide

2.5.7 STEADY-STATE LINEAR DYNAMIC ANALYSIS

Product: Abaqus/Standard

Steady-state linear dynamic analysis predicts the linear response of a structure subjected to continuous harmonic excitation. In many cases steady-state linear dynamic analysis in Abaqus/Standard uses the set of eigenmodes extracted in a previous eigenfrequency step to calculate the steady-state solution as a function of the frequency of the applied excitation. Abaqus/Standard also has a "direct" steady-state linear dynamic analysis procedure, in which the equations of steady harmonic motion of the system are solved directly without using the eigenmodes, and a "subspace" steady-state linear dynamic analysis procedure, in which the equations are projected onto a subspace of selected eigenmodes of the undamped system. These options are intended for systems in which the behavior is dependent on frequency, for when the model includes damping, or for systems in which the governing equations are not symmetric.

This section describes the linear steady-state response procedure based on the eigenmodes.

The projection of the equations of motion of the system onto the αth mode gives

$$\ddot{q}_\alpha + c_\alpha \dot{q}_\alpha + \omega_\alpha^2 q_\alpha = \frac{1}{m_\alpha}(f_{1\alpha} + i f_{2\alpha}) \exp(i\Omega t), \tag{2.5.7–1}$$

where q_α is the amplitude of mode α (the αth "generalized coordinate"), c_α is the damping associated with this mode (see below), ω_α is the undamped frequency of the mode, m_α is the generalized mass associated with the mode, and $(f_{1\alpha} + i f_{2\alpha}) \exp(i\Omega t)$ is the forcing associated with this mode. The forcing is defined by the frequency, Ω, and the real and imaginary parts of the nodal equivalent forces, F_1^N and F_2^N, projected onto the eigenmode ϕ_α^N:

$$f_{1\alpha} + i f_{2\alpha} = \phi_\alpha^N (F_1^N + i F_2^N).$$

In this equation summation is implied by the repeat of the superscript N indicating a degree of freedom in the model; but throughout this section we are working with only a single modal equation, so no summation is implied by the repeat of the mode subscript α. The load vector is written in terms of its real and imaginary parts, F_1^N and F_2^N, since this is the manner in which the loading is defined in Abaqus/Standard. It is equivalently possible to write the loading in terms of its magnitude, F_0^N, and phase, Ψ, as $F^N = F_0^N \exp i(\Omega t + \Psi)$, where $F_1^N = F_0 \cos \Psi$ and $F_2^N = F_0 \sin \Psi$.

Several representations of modal damping are provided. Modal damping defines $c_\alpha = 2\xi_\alpha \omega_\alpha$, where ξ_α is the fraction of critical damping in the mode. Structural damping gives a damping force proportional to the modal amplitude:

$$c_\alpha \dot{q}_\alpha = i s_\alpha \omega_\alpha^2 q_\alpha,$$

where s_α is the structural damping coefficient for the mode. Rayleigh damping is defined by $c_\alpha = \beta_\alpha + \gamma_\alpha \omega_\alpha^2$; β_α and γ_α are the Rayleigh coefficients damping low and high frequency modes, respectively. Rayleigh damping can be reproduced exactly by modal damping as

$$\xi_\alpha = \frac{\beta_\alpha}{2\omega_\alpha} + \frac{\gamma_\alpha \omega_\alpha}{2}.$$

Introducing all of these damping definitions into Equation 2.5.7–1 gives

$$\ddot{q}_\alpha + 2\xi_\alpha \omega_\alpha \dot{q}_\alpha + \left(\beta_\alpha + \gamma_\alpha \omega_\alpha^2\right) \dot{q}_\alpha + i s_\alpha \omega_\alpha^2 q_\alpha + \omega_\alpha^2 q_\alpha = \frac{1}{m_\alpha} \left(f_{1\alpha} + i f_{2\alpha}\right) \exp\left(i\Omega t\right). \qquad (2.5.7\text{–}2)$$

The solution to this equation is

$$q_\alpha = H_{0\alpha} f_{0\alpha} \exp i(\Omega t + \Psi_\alpha), \qquad (2.5.7\text{–}3)$$

where $f_{0\alpha} = \sqrt{(f_{1\alpha})^2 + (f_{2\alpha})^2}$ is the amplitude of the projected load vector and $H_{0\alpha}(\Omega)$ is the amplitude of the complex "transfer function" for mode α. $H_{0\alpha} f_{0\alpha}$ defines the response in mode α from the force projection onto that mode and is defined by its real and imaginary parts as

$$\Re(H_\alpha) = \frac{1}{m_\alpha} \left[\frac{f_{1\alpha}\left(\omega_\alpha^2 - \Omega^2\right)}{\left(\omega_\alpha^2 - \Omega^2\right)^2 + \left(\eta_\alpha \Omega\right)^2} + \frac{f_{2\alpha} \eta_\alpha \Omega}{\left(\omega_\alpha^2 - \Omega^2\right)^2 + \left(\eta_\alpha \Omega\right)^2} \right]$$

$$\Im(H_\alpha) = \frac{1}{m_\alpha} \left[-\frac{f_{1\alpha} \eta_\alpha \Omega}{\left(\omega_\alpha^2 - \Omega^2\right)^2 + \left(\eta_\alpha \Omega\right)^2} + \frac{f_{2\alpha}\left(\omega_\alpha^2 - \Omega^2\right)}{\left(\omega_\alpha^2 - \Omega^2\right)^2 + \left(\eta_\alpha \Omega\right)^2} \right],$$

where η_α denotes

$$\eta_\alpha = 2\xi_\alpha \omega_\alpha + \beta_\alpha + \gamma_\alpha \omega_\alpha^2 + \frac{s_\alpha \omega_\alpha^2}{\Omega}.$$

The amplitude of the response is

$$H_{0\alpha} f_{0\alpha} = \sqrt{\Re(H_\alpha)^2 + \Im(H_\alpha)^2} = \frac{1}{m_\alpha} \frac{1}{\sqrt{\left(\omega_\alpha^2 - \Omega^2\right)^2 + \left(\eta_\alpha \Omega\right)^2}} f_{0\alpha},$$

and the phase angle of the response is

$$\Psi_\alpha = \arctan\left(\Im(H_\alpha) / \Re(H_\alpha)\right).$$

If a harmonic base motion is applied, the real and imaginary parts of the modal loads are given as

$$f_{1\alpha} = -\frac{1}{m_\alpha} \phi_\alpha^N M^{NM} \hat{e}_j^M a_{1j} \exp\left(i\Omega t\right),$$

$$f_{2\alpha} = -\frac{1}{m_\alpha} \phi_\alpha^N M^{NM} \hat{e}_j^M a_{2j} \exp\left(i\Omega t\right),$$

where M^{NM} is the structure's mass matrix and $\hat{e}_j^M$ is a vector that has unit magnitude in the direction of the base acceleration at any grounded node and is otherwise zero; a_{1j} and a_{2j} are the real and imaginary parts of the base acceleration. If the base motion is given as a velocity or displacement, the corresponding accelerations are $a_1 = -\Omega v_1$ and $a2 = -\Omega v_2$, where v_1 and v_2 are the real and imaginary parts of the velocity, or $a_1 = -\Omega^2 u_1$ and $a_2 = -\Omega^2 u_2$, where u_1 and u_2 are the real and imaginary parts of the displacement.

The peak amplitude of any physical variable, u^N, is available from the modal amplitudes as

$$u^N = \sum_\alpha \phi_\alpha^N q_\alpha .$$

Steady-state response is given as a frequency sweep through a user-specified range of frequencies. Since the structural response peaks around the natural frequencies, a bias function is used to cluster the response points around the frequencies. The biasing is described in "Mode-based steady-state dynamic analysis," Section 6.3.8 of the Abaqus Analysis User's Guide.

Reference

- "Mode-based steady-state dynamic analysis," Section 6.3.8 of the Abaqus Analysis User's Guide

2.5.8 RANDOM RESPONSE ANALYSIS

Product: Abaqus/Standard

Random response linear dynamic analysis is used to predict the response of a structure subjected to a nondeterministic continuous excitation that is expressed in a statistical sense by a cross-spectral density (CSD) matrix. The random response procedure uses the set of eigenmodes extracted in a previous eigenfrequency step to calculate the corresponding power spectral densities (PSD) of response variables (stresses, strains, displacements, etc.) and, hence—if required—the variance and root mean square values of these same variables. This section provides brief definitions and explanations of the terms used in this type of analysis. Detailed discussion of the theory of random response analysis is provided in the books by Clough and Penzien (1975), Hurty and Rubinstein (1964), and Thompson (1988).

Examples of random response analysis are the study of the response of an airplane to turbulence; the response of a car to road surface imperfections; the response of a structure to noise, such as the "jet noise" emitted by a jet engine; and the response of a building to an earthquake.

Since the loading is nondeterministic, it can be characterized only in a statistical sense. We need some assumptions to make this characterization possible. Although the excitation varies in time, in some sense it must be *stationary*—its statistical properties must not vary with time. Thus, if $x(t)$ is the variable being considered (such as the height of the road surface in the case of a car driving down a rough road), then any statistical function of x, $f(x)$, must have the same value regardless of what time origin we use to compute f:

$$f\big(x(t)\big) = f\big(x(t+\tau)\big) \quad \text{for any } \tau.$$

We also need the excitation to be *ergodic*. This term means that, if we take several samples of the excitation, the time average of each sample is the same.

These restrictions ensure that the excitation is, statistically, constant. In the following discussion we also assume that the random variables are real, which is the case for the variables that we need to consider.

Statistical measures

We define some measures of a variable that characterize it in a statistical sense.

The *mean value* of a random variable $x(t)$ is

$$E(x) = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t)\, dt.$$

Since the dynamic response is computed about a static equilibrium configuration, the mean value of any dynamic input or response variable will always be zero:

$$E(x) = 0.$$

The *variance* of a random variable measures the average square difference between the point value of the variable and its mean:

$$\sigma_r^2(x) = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \big(x(t) - E(x)\big)^2 \, dt.$$

Since $E(x) = 0$ for our applications, the variance is the same as the *mean square value*:

$$E(x^2) = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x^2(t) \, dt.$$

The units of variance are (amplitude)^2, so that—for example—the variance of a force has units of (force)^2. Generally we prefer to use the same units as the variable itself. Therefore, output variables in Abaqus/Standard are given as *root mean square* ("RMS") values, $\sigma_r(x) = \sqrt{\sigma_r^2}$.

Correlation

Correlation measures the similarity between two variables. Thus, the *cross-correlation* between two random functions of time, $x_1(t)$ and $x_2(t)$, is the integration of the product of the two variables, with one of them shifted in time by some fixed value τ to allow for the possibility that they are similar but shifted in time.

(Such a case would arise, for example, in studying a car driving along a rough road. If the separation of the axles is d and the car is moving at a steady speed v, the back axle sees the same road profile as the front axle, but delayed by a time $\tau = d/v$. Assume that the road profile moves each wheel which, in turn applies a force to the car frame through the suspension. If F_2 is the force applied to the rear axle (as a concentrated load in Abaqus) and F_1 is the force applied to the front axle, $F_2(t + d/v) = F_1(t)$.)

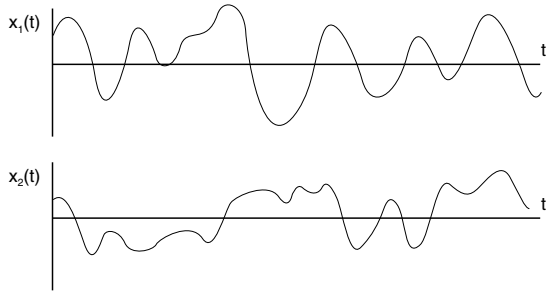


Figure 2.5.8–1 Two random records to be correlated.

The cross-correlation function is, thus, defined as

$$R_{x_1x_2}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x_1(t)x_2(t+\tau) \, dt.$$

Since the mean value of any variable is zero, on average each variable has equal positive and negative content. If the variables are quite similar, their cross-correlation (for some values of τ) will be large; if they are not similar, the product $x_1 x_2$ will sometimes be negative and sometimes positive so that the integral over all time will provide a much smaller value, regardless of the choice of τ. A simple result is

$$\begin{aligned} R_{x_1 x_2}(\tau) &= \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x_1(t) x_2(t+\tau)\, dt \\ &= \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x_1(t-\tau) x_2(t)\, dt \\ &= \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x_2(t) x_1(t-\tau)\, dt \\ &= R_{x_2 x_1}(-\tau). \end{aligned}$$

For convenience the cross-correlation can be normalized to define the nondimensional *normalized cross-correlation*:

$$r_{x_1 x_2}(\tau) = \frac{R_{x_1 x_2}(\tau)}{\sqrt{\sigma_r^2(x_1)\sigma_r^2(x_2)}}.$$

Thus, if $x_1 = x_2$, $r(0) = 1$; if $x_1 = -x_2$, $r(0) = -1$. If x_1 and x_2 are entirely dissimilar, $r \to 0$. (When $r = 0$ the variables are said to be *orthogonal*.) Clearly $-1 \le r \le 1$, with small values of r indicating that x_1 and x_2 have quite different time histories.

Now consider the cross-correlation of a variable with itself: the *autocorrelation*. Intuitively we can see that if the variable is "very random," its autocorrelation will be very small whenever $\tau \neq 0$: there will be no time shift that allows the variable to correlate with itself. However, if the variable is not so random—if it is just a vibration at a fixed frequency—the autocorrelation will be close to ± 1 whenever τ is chosen to be some integer multiple of half the period of the vibration. Thus, the autocorrelation provides a measure of how random a variable really is.

The autocorrelation of a variable $x(t)$ is, therefore,

$$R(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) x(t+\tau)\, dt.$$

Clearly, as $\tau \to 0$, $R(\tau) \to \sigma_r^2$: the autocorrelation equals the variance (the mean square value). We can, therefore, also use the *normalized autocorrelation*:

$$r(\tau) = \frac{R(\tau)}{\sigma_r^2}.$$

Obviously $R(\tau)$ is symmetric about $\tau = 0$:

$$R(-\tau) = R(\tau);$$

and the value of $R(\tau)$ never exceeds its value at $\tau = 0$:

$$R(\tau) \leq R(0) = \sigma_r^2.$$

The autocorrelation function of records with very similar amplitude over the wide range of frequencies drops off rapidly as τ increases. This kind of function is known as a "wide band" random function.

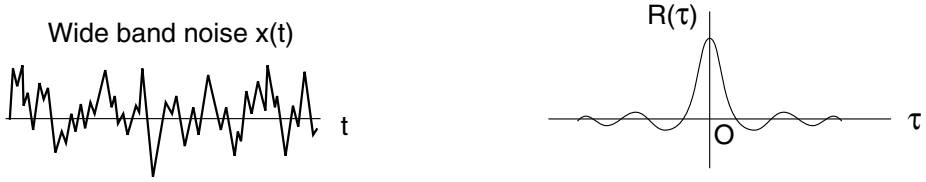


Figure 2.5.8–2 Wide band noise record and its autocorrelation.

The most extreme wide band random function would have an autocorrelation that is just a delta function:

$$R(\tau) = \delta(\tau) = \begin{cases} \sigma_r^2 & \text{if } \tau = 0 \\ 0 & \text{if } \tau \neq 0. \end{cases}$$

Such a function is called *white noise.* White noise has the same amplitude at all frequencies, and its autocorrelation is zero except at $\tau = 0$.

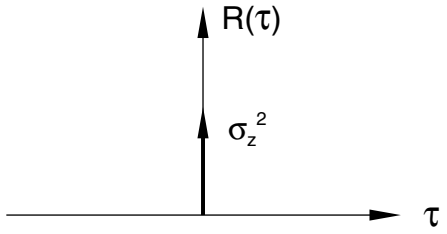


Figure 2.5.8–3 Autocorrelation of white noise.

Let us now consider the opposite case, known as a "narrow band" function. The extreme case of such a function is a simple sinusoidal vibration at a single frequency: $x = A \sin \omega_0 t$. Then $R(\tau)$ must also be periodic, since $R(\tau)$ must attain the same value, $R(0) = \sigma_r^2$, each time the shift, τ, corresponds to the period of vibration. Performing the integration through time,

$$R(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} A^2 \sin\omega_0 t \sin\omega_0(t+\tau)\, dt$$

$$= A^2 \lim_{T\to\infty} \frac{1}{T} \left(\cos\omega_0\tau \int_{-\frac{T}{2}}^{\frac{T}{2}} \sin^2\omega_0 t\, dt + \sin\omega_0\tau \int_{-\frac{T}{2}}^{\frac{T}{2}} \sin\omega_0 t \cos\omega_0 t\, dt \right)$$

$$= A^2 \lim_{T\to\infty} \frac{1}{T} \left(\cos\omega_0\tau \int_{-\frac{T}{2}}^{\frac{T}{2}} \frac{1}{2}(1-\cos 2\omega_0 t)\, dt + \sin\omega_0\tau \int_{-\frac{T}{2}}^{\frac{T}{2}} \frac{1}{2}\sin 2\omega_0 t\, dt \right)$$

$$= \frac{1}{2} A^2 \cos\omega_0\tau = \sigma_r^2 \cos\omega_0\tau.$$

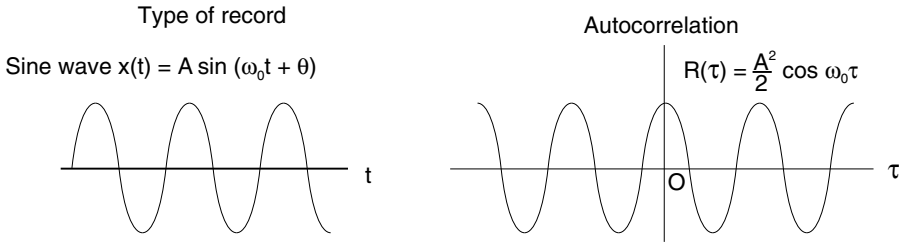


Figure 2.5.8–4 Sine wave and its autocorrelation.

The autocorrelation, $R(\tau)$, thus tells us about the nature of the random variable. If $R(\tau)$ drops off rapidly as the time shift τ moves away from $\tau = 0$, the variable has a broad frequency content; if it drops off more slowly and exhibits a cosine profile, the variable has a narrow frequency content centered around the frequency corresponding to the periodicity of $R(\tau)$.

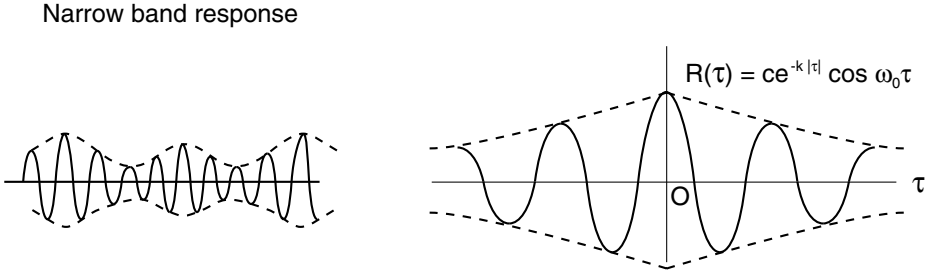


Figure 2.5.8–5 Narrow band record and its autocorrelation.

We can extend this concept to detect the frequency content of a random variable by cross-correlating the variable with a sine wave: sweeping the wave over a range of frequencies and examining the cross-correlation tells us whether the random variable is dominated by oscillation at particular frequencies.

We begin to see that the nature of stationary, ergodic random processes is best understood by examining them in the frequency domain.

As an illustration, consider a variable, $x(t)$, which contains many discrete frequencies. We can write $x(t)$ in terms of a Fourier series expanded in N steps of a fundamental frequency ω_0:

$$x(t) = \sum_{n=1}^{N} \left[a_n \cos(in\omega_0 t) + b_n \sin(in\omega_0 t) \right].$$

The series begins with the $n = 1$ term because the mean of the variable must be zero. We can write this series more compactly as a complex Fourier series (keeping in mind that we will be interested only in the real part):

$$x(t) = \sum_{n=-N}^{N} A_n \exp(in\omega_0 t),$$

where $A_n = \Re(A_n) + i\Im(A_n)$ is the complex amplitude of the nth term, A_n^* is the complex conjugate of A_n:

$$A_n^* = \Re(A_n) - i\Im(A_n),$$

and

$$A_n = \begin{cases} (a_n - ib_n)/2 & \text{for } n > 0, \\ (a_n + ib_n)/2 & \text{for } n < 0, \\ 0 & \text{for } n = 0. \end{cases}$$

The variance of $x(t)$ is

$$\sigma_r^2(x) = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \sum_{n=-N}^{N} A_n^* \exp(-in\omega_0 t) A_n \exp(in\omega_0 t)\, dt,$$

using the orthogonality of Fourier terms. Continuing,

$$\begin{aligned} \sigma_r^2(x) &= \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \sum_{n=-N}^{N} A_n A_n^*\, dt \\ &= \sum_{n=-N}^{N} A_n A_n^* = \sum_{n=-N}^{N} \left| A_n \right|^2 = \sum_{n=-N}^{N} \sigma_r^2(x_n), \end{aligned}$$

where

$$x_n = \Re\big(A_n \exp(in\omega_0 t)\big)$$

is the nth component of the Fourier series.

Thus, thanks to the orthogonality of Fourier terms, the variance (the mean square value) of the series is the sum of the variances (the mean square values) of its components. In particular, we see that $A_n A_n^*$ is the variance, or mean square value, of the variable at the frequency $n\omega_0$.

The contribution to the variance of x, $\sigma_r^2(x)$, at the frequency $f_n = n\omega_0/(2\pi)$, per unit frequency, is thus

$$S_x(f_n) = \frac{2\pi}{\omega_0} A_n A_n^*,$$

since we are stepping up the frequency range in steps of $\Delta f = \omega_0/(2\pi)$. The variance can, therefore, be written as

$$\sigma_r^2(x) = \sum_{n=-N}^{N} S_x(f_n)\Delta f.$$

As we examine x as a function of frequency, $S_x(f_n)$ tells us the amount of "power" (in the sense of mean square value) contained in x, per unit frequency, at the frequency f_n. As we consider smaller and smaller intervals, $\Delta f \to 0$, S_x is the *power spectral density (PSD)* of the variable x:

$$\sigma_r^2(x) = \int_{-\infty}^{\infty} S_x(f)\, df,$$

where f is the frequency in cycles per time (usually Hz).

Notice that S_x has units of (variable)^2/frequency, where (variable) is the unit of the variable (displacement, force, stress, etc.). In this case "frequency" is almost always given in Hz, although—since Abaqus does not have any built-in units—the frequency could be expressed in any other units of cycles per time. However, S_x should not be given per circular frequency (radians per time): Abaqus assumes that $S_x(f)$, not $S_x(\omega)$.

Fourier transforms

Since the variables of interest in random response analysis are characterized as functions of frequency, the Fourier transform plays a major role in converting from the time domain to the frequency domain and *vice versa*. The Fourier transform of $x(t)$, which we write as $\overline{x}(f)$, is defined by

$$x(t) = \int_{-\infty}^{\infty} \overline{x}(f) \exp(i2\pi ft)\, df$$

or, in terms of the circular frequency $\omega = 2\pi f$,

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \overline{x}(\omega) \exp(i\omega t)\, d\omega.$$

Simple manipulation provides

$$x(t) = \frac{1}{2\pi}\int_0^\infty \left[\left(\overline{x}(\omega) + \overline{x}(-\omega)\right)\cos\omega t + i\left(\overline{x}(\omega) - \overline{x}(-\omega)\right)\sin\omega t\right] d\omega,$$

so $\left(\overline{x}(\omega) + \overline{x}(-\omega)\right)/(2\pi)$ is the amplitude of the cosine term in $x(t)$ at the circular frequency ω, while $\left(\overline{x}(\omega) - \overline{x}(-\omega)\right)/(2\pi)$ is the amplitude of the sine term. We, thus, see the physical meaning of the Fourier transform—it provides the complex magnitude (the amplitude and phase) of the content of $x(t)$ at a particular frequency.

If $x(t)$ is real only (which is the case for the variables we need to consider), this expression shows that we must have

$$\Im\left(\overline{x}(\omega)\right) = -\Im\left(\overline{x}(-\omega)\right)$$

and

$$\Re\left(\overline{x}(\omega)\right) = \Re\left(\overline{x}(-\omega)\right),$$

which means that

$$\overline{x}(-\omega) = \overline{x}^*(\omega).$$

For completeness we also note the inverse transformation,

$$\overline{x}(f) = \int_{-\infty}^{\infty} x(t)\exp(-i2\pi ft)\, dt,$$

which shows that the transformations between the time and frequency domains are rather symmetrical.

We now need Parseval's theorem:

$$\begin{aligned}\int_{-\infty}^{\infty} x_1(t)x_2^*(t)\, dt &= \int_{-\infty}^{\infty} x_2^*(t)\int_{-\infty}^{\infty} \overline{x}_1(f)\exp(i2\pi ft)\, df\, dt \\ &= \int_{-\infty}^{\infty} \overline{x}_1(f)\int_{-\infty}^{\infty} x_2^*(t)\exp(i2\pi ft)\, dt\, df \\ &= \int_{-\infty}^{\infty} \overline{x}_1(f)\overline{x}_2^*(f)\, df.\end{aligned}$$

Applying this theorem to the variance (the mean square value):

$$\begin{aligned}\sigma_r^2 &= \lim_{T\to\infty}\frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} x^2(t)\, dt \\ &= \int_{-\infty}^{\infty}\lim_{T\to\infty}\frac{1}{T}\overline{x}(f)\overline{x}^*(f)\, df.\end{aligned}$$

Since $x(t)$ is real, we know that $\overline{x}(-f) = \overline{x}^*(f)$, and so

$$\overline{x}(-f)\overline{x}^*(-f) = \overline{x}^*(f)\overline{x}(f) = \overline{x}(f)\overline{x}^*(f).$$

We have already shown that we can write the variance in terms of the power spectral density as

$$\sigma_r^2(x) = \int_{-\infty}^{\infty} S_x(f)\, df.$$

By comparison,

$$S_x(f) = \lim_{T\to\infty} \frac{1}{T}\overline{x}(f)\overline{x}^*(f).$$

We also see that

$$S_x(-f) = S_x(f).$$

To avoid integration over negative frequencies, we write the variance as

$$\begin{aligned}\sigma_r^2(x) &= \int_0^{\infty} \left[S_x(f) + S_x(-f)\right]\, df \\ &= \int_0^{\infty} \tilde{S}_x(f)\, df,\end{aligned}$$

where $\tilde{S}_x(f)$ is the *single-sided PSD* defined as

$$\tilde{S}_x(f) = S_x(f) + S_x(-f), \quad \text{for } f \geq 0.$$

Since $S_x(-f) = S_x(f)$, we see that

$$\tilde{S}_x(f) = 2S_x(f).$$

Now consider the autocorrelation function:

$$\begin{aligned}R(\tau) &= \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t)x(t+\tau)\, dt \\ &= \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) \left[\int_{-\infty}^{\infty} \overline{x}(f) \exp\big(i2\pi f(t+\tau)\big) df\right] dt \\ &= \int_{-\infty}^{\infty} \lim_{T\to\infty} \frac{1}{T} \left[\int_{-\infty}^{\infty} x(t) \exp(i2\pi f t)\, dt\right] \overline{x}(f) \exp(i2\pi f\tau)\, df \\ &= \int_{-\infty}^{\infty} \left(\lim_{T\to\infty} \frac{1}{T}\overline{x}^*(f)\overline{x}(f)\right) \exp(i2\pi f\tau)\, df \\ &= \int_{-\infty}^{\infty} S_x(f) \exp(i2\pi f\tau)\, df,\end{aligned}$$

(using the result above). Thus, the power spectral density is the Fourier transform of the autocorrelation function. The inverse transform is

$$S_x(f) = \int_{-\infty}^{\infty} R(\tau) \exp(-i2\pi f\tau)\, d\tau.$$

Since $R(\tau)$ is symmetric about $\tau = 0$ ($R(-\tau) = R(\tau)$), we can also write this equation as

$$\begin{aligned} S_x(f) &= \int_0^{\infty} R(\tau)\big(\exp(-i2\pi f\tau) + \exp(i2\pi f\tau)\big)\, d\tau \\ &= 2\int_0^{\infty} R(\tau)\cos(2\pi f\tau)\, d\tau, \end{aligned}$$

so that

$$\tilde{S}_x(f) = 4\int_0^{\infty} R(\tau)\cos(2\pi f\tau)\, d\tau.$$

Cross-spectral density

Following a similar argument to that used above to develop the idea of the power spectral density, we can define the *cross-spectral density (CSD) function*, $S_{x_1x_2}(f)$, which gives the cross-correlation between two variables, $R_{x_1x_2}(\tau)$, as

$$R_{x_1x_2}(\tau) = \int_{-\infty}^{\infty} S_{x_1x_2}(f) \exp(i2\pi f\tau)\, df,$$

with the inverse transformation

$$S_{x_1x_2}(f) = \int_{-\infty}^{\infty} R_{x_1x_2}(\tau) \exp(-i2\pi f\tau)\, d\tau.$$

Transforming the original definition of $R_{x_1x_2}(\tau)$ to the frequency domain provides

$$\begin{aligned} R_{x_1x_2}(\tau) &= \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x_1(t)x_2(t+\tau)\, dt \\ &= \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x_1(t) \int_{-\infty}^{\infty} \overline{x}_2(f) \exp\big(i2\pi f(t+\tau)\big) df\, dt \\ &= \int_{-\infty}^{\infty} \lim_{T\to\infty} \frac{1}{T} \left[\int_{-\infty}^{\infty} x_1(t) \exp(i2\pi ft)\, dt\right] \overline{x}_2(f) \exp(i2\pi f\tau)\, df \\ &= \int_{-\infty}^{\infty} \left(\lim_{T\to\infty} \frac{1}{T} \overline{x}_1^*(f)\overline{x}_2(f)\right) \exp(i2\pi f\tau)\, df. \end{aligned}$$

By comparison,

$$S_{x_1x_2}(f) = \lim_{T\to\infty} \frac{1}{T}\overline{x}_1^*(f)\overline{x}_2(f).$$

We could also write

$$\begin{aligned} S_{x_1x_2}(f) &= \left[\lim_{T\to\infty} \frac{1}{T}\overline{x}_2^*(f)\overline{x}_1(f)\right]^* \\ &= S^*_{x_2x_1}(f) = S_{x_2x_1}(-f). \end{aligned}$$

Random response analysis

The general concept of random response analysis is now clear. A system is excited by some random loads or prescribed base motions, which are characterized in the frequency domain by a matrix of cross-spectral density functions, $S_{NM}(f)$. Here we think of $\boldsymbol{N}$ and $\boldsymbol{M}$ as two of the degrees of freedom of the finite element model that are exposed to the random loads or prescribed base motions.

In typical applications the range of frequencies will be limited to those to which we know the structure will respond—we do not need to consider frequencies that are higher than the modes in which we expect the structure to respond.

The values of $S_{NM}(f)$ might be provided by Fourier transformation of the cross-correlation of time records or by the Fourier transformation of the autocorrelation of a single time record, together with known geometric data, as in the case of the car driving along a roughly grooved road, where the autocorrelation of the road surface profile, together with the speed of the car and the axle separation, allow $S_{12}(f)$ to be defined for the front (1) and rear (2) axles, as shown above. (If, in this case, the road profile seen by the wheels on the left side of the car is not similar to that seen by the wheels on the right side of the car, the cross-correlation of the left and right road surface profiles will also be required to define the excitation.)

The system will respond to this excitation. We are usually interested in looking at the power spectral densities of the usual response variables—stress, displacement, etc. The PSD history of any particular variable will tell us the frequencies at which the system is most excited by the random loading.

We might also compute the cross-spectral densities between variables. These are usually not of interest, and Abaqus/Standard does not provide them. (They might be needed if the analysis involves obtaining results that, in turn, will define the loading for some other system. For example, the response of a building to seismic loading might be used to obtain the motions of the attachment points for a piping system in the building so that the piping system can then be analyzed. The only option would be to model the entire system together.)

An overall picture is provided by looking at the variance (the mean square value) of any variable; the RMS value is provided for this purpose. The RMS value is used instead of variance because it has the same units as the variable itself. Abaqus/Standard computes it by integrating the single-sided power spectral density of the variable over the frequency range, since

$$\sigma_r = \sqrt{R(0)} = \sqrt{\int_0^\infty \tilde{S}_x(f)\, df}.$$

This integration is performed numerically by using the trapezoidal rule over the range of frequencies specified for the random response step:

$$\sigma_r \approx \sqrt{\frac{1}{2}\sum_{i=2}^{N}(\tilde{S}_i + \tilde{S}_{i-1})(f_i - f_{i-1})},$$

where $\tilde{S}_i$ is the $\tilde{S}_x$ at the frequency f_i and N is the number of points at which the response was calculated. N will depend on the number of eigenmodes used in the superposition and on the user-specified number of points between the eigenfrequencies.

The user must ensure that enough frequency points are specified so that this approximate integration will be sufficiently accurate.

The transformation of the problem into the frequency domain inherently assumes that the system under study is responding linearly: the random response procedure is considered as a linear perturbation analysis step.

What remains, then, is for us to consider how Abaqus/Standard finds the linear response to the random excitation.

The frequency response function

Random response is studied in the frequency domain. Therefore, we need the transformation from load to response as a function of frequency. Since the random response is treated as the integration of a series of sinusoidal vibrations, this transformation is based on the same steady-state response function used for a steady-state dynamic analysis and described in "Steady-state linear dynamic analysis," Section 2.5.7.

The discrete (finite element) linear dynamic system has the equilibrium equation

$$\delta u^N \left(M^{NM}\ddot{u}^M + C^{NM}\dot{u}^M + K^{NM}u^M\right) = \delta u^N F^N,$$

where M^{NM} is the mass matrix, C^{NM} is the damping matrix, K^{NM} is the stiffness matrix, F^N are the external loads, u^N is the value of degree of freedom N of the finite element model (usually a displacement or rotation component, or an acoustic pressure), and δu^N is an arbitrary virtual variation.

We project the problem onto the eigenmodes of the system. To do this the modes are first extracted from the undamped system:

$$\left(M^{NM}\omega_{\bar{\alpha}}^2 - K^{NM}\right)\phi_{\bar{\alpha}}^N = 0.$$

(Here, and throughout the remainder of this section, repeated subscripts and superscripts are assumed to be summed over the appropriate range except when they are barred, like $\bar{\alpha}$ above. Roman superscripts and subscripts indicate physical degrees of freedom; Greek superscripts and subscripts indicate modal variables.)

Typically the structural dynamic response is well represented by a small number of the lower modes of the model, so the number of modes is usually $O(10^1)$–$O(10^2)$, while the number of physical degrees of freedom might be $O(10^3)$–$O(10^5)$.

The eigenmodes are orthogonal across the mass and stiffness matrices:

$$\phi_\alpha^N M^{NM} \phi_\beta^M = \begin{cases} m_\alpha & \text{if } \alpha = \beta \\ 0 & \text{if } \alpha \neq \beta; \end{cases}$$
$$\phi_\alpha^N K^{NM} \phi_\beta^M = \begin{cases} k_\alpha & \text{if } \alpha = \beta \\ 0 & \text{if } \alpha \neq \beta. \end{cases}$$

The eigenmodes can be displacement or mass normalized. For displacement normalization the largest entry in ϕ_α^N is 1.0; for mass normalization the largest entry is $m_\alpha = 1$. In SIM architecture we normalize only with mass.

We assume that any damping is in the general form of "Rayleigh damping":

$$C^{NM} = \alpha M^{NM} + \beta K^{NM}$$

so that C^{NM} will also project into a diagonal damping matrix c_α.

The problem, thus, projects into a set of uncoupled modal response equations,

$$m_{\bar{\alpha}} \ddot{q}_{\bar{\alpha}} + c_{\bar{\alpha}} \dot{q}_{\bar{\alpha}} + k_{\bar{\alpha}} q_{\bar{\alpha}} = f_\alpha,$$

where

$$f_\alpha = \phi_\alpha^N F^N$$

is the generalized load for mode α and q_α is the "generalized coordinate" (the modal amplitude) for mode α.

Steady-state excitation is of the form

$$f_\alpha = A \exp(i2\pi ft)$$

and creates response of similar form that we write as

$$q_\alpha = HA \exp(i2\pi ft),$$

where $H(f)$ is the *complex frequency response function* defined in "Steady-state linear dynamic analysis," Section 2.5.7.

Response development

Random loading is defined by the cross-spectral density matrix $S_{NM}^F(f)$, which links all loaded degrees of freedom (N and M). Projecting this matrix onto the modes provides the cross-spectral density function for the generalized (modal) loads:

$$S_{\alpha\beta}^f(f) = \phi_\alpha^N S_{NM}^F(f) \phi_\beta^M.$$

The complex frequency response function then defines the response of the generalized coordinates as

$$S^q_{\alpha\beta}(f) = H_{\bar{\alpha}} S^f_{\bar{\alpha}\bar{\beta}}(f) H^*_{\bar{\beta}},$$

where H^*_α is the complex conjugate of H_α.

Finally, the response of the physical variables is recovered from the modal responses as

$$S^u_{NM}(f) = \phi^N_\alpha S^q_{\alpha\beta}(f) \phi^M_\beta$$

so that the power spectral density of degree of freedom u^N is

$$S^u_{\bar{N}\bar{N}}(f) = \phi^{\bar{N}}_\alpha S^q_{\alpha\beta}(f) \phi^{\bar{N}}_\beta.$$

The PSDs for the velocity and acceleration of the same variable are

$$S^{\dot{u}}_{\bar{N}\bar{N}}(f) = (2\pi f)^2 S^u_{\bar{N}\bar{N}}(f)$$

and

$$S^{\ddot{u}}_{\bar{N}\bar{N}}(f) = (2\pi f)^4 S^u_{\bar{N}\bar{N}}(f).$$

Recall that we may typically have $O(10^1)$–$O(10^2)$ eigenmodes but many more ($O(10^3)$–$O(10^5)$) physical degrees of freedom. Therefore, if many of the physical degrees of freedom are loaded (as in the case of a shell structure exposed to random acoustic noise), it may be computationally expensive to perform operations such as

$$S^f_{\alpha\beta}(f) = \phi^N_\alpha S^F_{NM}(f) \phi^M_\beta,$$

which involve products over all loaded physical degrees of freedom and must be done at each frequency in the range considered.

In contrast, operations such as

$$S^q_{\alpha\beta}(f) = H_{\bar{\alpha}}(f) S^f_{\bar{\alpha}\bar{\beta}}(f) H^*_{\bar{\beta}}(f)$$

are relatively inexpensive since they are done independently for each combination of modes.

Forming

$$S^u_{\bar{N}\bar{N}}(f) = \phi^{\bar{N}}_\alpha S^q_{\alpha\beta}(f) \phi^{\bar{N}}_\beta$$

may be expensive if we choose to compute the results for a large selection of physical variables (displacements, velocities, accelerations, stresses, etc.). Abaqus/Standard will calculate the response only for the element and nodal variables requested. However, if a restart analysis is requested with the random response procedure, all variables are computed at the requested restart frequency, which can add substantially to the computational cost. The user is advised to write the restart file for the last increment only. To reduce the computational cost of random response analysis, Abaqus/Standard assumes that

the cross-spectral density matrix for the loading can be separated into a frequency-dependent scalar function (containing the units of the CSD) and a set of coupling terms that are independent of frequency as follows:

$$
\begin{aligned}
S^F_{NM}(f) &= \sum_J P^J(f) \sum_I \Psi^{IJ}_{NM} \qquad \text{for } N < M,\\
S^F_{NM}(f) &= \left[S^F_{MN}(f)\right]^* \qquad \text{for } N > M,\\
S^F_{NN}(f) &= \Re\left(\sum_J P^J(f) \sum_I \Psi^{IJ}_{NN}\right).
\end{aligned}
$$

Here J is the number of cross-correlation definitions included in the random response step. Each cross-correlation definition refers to an input complex frequency function, $P^J(f)$. The spatial cross-correlations are then defined by the complex set of values Ψ^{IJ}_{NM}.

With this approach the cross-spectral density function for the generalized loads, $S^f_{\alpha\beta}(f)$, can be constructed as

$$
\begin{aligned}
S^f_{\alpha\beta}(f) &= \sum_J P^J(f) \left(\phi^N_\alpha \phi^M_\beta \sum_I \Psi^{IJ}_{NM}\right) \qquad \text{for } N < M,\\
S^f_{\alpha\beta}(f) &= \left[S^f_{\beta\alpha}(f)\right]^* \qquad \text{for } N > M,\\
S^f_{\alpha\alpha}(f) &= \Re\left[\sum_J P^J(f) \left(\phi^N_\alpha \phi^M_\alpha \sum_I \Psi^{IJ}_{NM}\right)\right].
\end{aligned}
$$

Since the $\phi^N_\alpha \phi^M_\beta \sum_I \Psi^{IJ}_{NM}$ are not functions of frequency, they can be computed once only, leaving the frequency-dependent operations to be done only in the space of the eigenmodes.

Although this procedure is not natural for typical correlated loadings (like road excitation or jet noise), loadings can always be defined this way by using enough cross-correlation definitions. The approach then reduces the computational cost for models with many loaded physical degrees of freedom. The approach works well for uncorrelated and fully correlated loadings, which are quite common cases.

Decibel conversion

Abaqus/Standard allows the user to provide an input PSD (say $P(f)$) in decibel units rather than units of power/frequency. There are various ways to convert from decibel units to units of power/frequency, depending on how the frequencies in one octave band are related to the frequencies in the next. A general formula relates the center (midband) frequencies f_c between octaves as

$$
\frac{f_c^{(i)}}{f_c^{(i-1)}} = 2^x,
$$

where the superscript (i) denotes ith octave band and x is a chosen value. For example, $x = 1$ for *full octave band* conversion, and $x = 1/3$ for *one-third octave band* conversion. Abaqus/Standard uses full

octave band conversion to convert from decibel units to units of power/frequency. For full octave band conversion, as shown by the above equation, the center frequency doubles from octave band to octave band.

Since decibel units are based upon log scales, the center frequency for an octave band bounded by lower frequency f_{L} and upper frequency f_{U} is given by

$$\log f_{\mathrm{c}} = \frac{1}{2}\log f_{\mathrm{U}} + \frac{1}{2}\log f_{\mathrm{L}}.$$

Since $f_{\mathrm{U}}/f_{\mathrm{L}} = 2^{x}$, we can easily show that

$$f_{\mathrm{U}} = f_{\mathrm{c}} 2^{0.5x}$$

and

$$f_{\mathrm{L}} = f_{\mathrm{c}} 2^{-0.5x}.$$

Thus, the change in frequency within any given octave band is

$$\Delta f = f_{\mathrm{U}} - f_{\mathrm{L}} = f_{\mathrm{c}}[2^{0.5x} - 2^{-0.5x}].$$

To convert from one type of conversion formula to the next, we need the following general decibel to power/frequency conversion equation:

$$db(f) = 10\log\frac{P(f)}{P_{\mathrm{ref}}^{(x)}/\Delta f},$$

where $P_{\mathrm{ref}}^{(x)}$ is a reference power value. The subscript (x) means that the power reference is given for the type of conversion represented by x (e.g., full octave band conversion for $x = 1$ or one-third octave band conversion for $x = 1/3$). When $x = 1$, we will simply use the notation P_{ref}. Thus, since Abaqus/Standard uses a full octave band conversion, $\Delta f = f_{\mathrm{c}}/\sqrt{2}$ and

$$P(f) = \frac{\sqrt{2}}{f_{\mathrm{c}}} P_{\mathrm{ref}} 10^{db(f)/10}.$$

The PSD data can be given with respect to some other type of octave band frequency scale. In that case we can convert the PSD data at those frequencies coinciding with the full octave band scale by computing an equivalent full octave band reference power based on the following ratio:

$$\frac{P_{\mathrm{ref}}}{P_{\mathrm{ref}}^{(x)}} = \frac{1}{\sqrt{2}[2^{0.5x} - 2^{-0.5x}]}.$$

For example, if we are given $P_{\mathrm{ref}}^{(1/3)}$ (i.e., one-third octave band frequency scale), the equivalent full octave band reference power value would be

$$P_{\text{ref}} = 3.0536 P_{\text{ref}}^{(1/3)}.$$

This conversion would be valid only at the one-third octave band center frequencies that coincide with the full octave band center frequencies. Thus, only every third data point should be considered.

von Mises stress computation

The computation of PSD and RMS of von Mises stresses in Abaqus is based on work by Segalman, et al. (1998). Under this approach the PSD of von Mises stress at a node *a* is given by

$$S^a_{mises}(f) = \sum_{\beta=1}^{m} \sum_{\alpha=1}^{m} S_{\alpha\beta}(f) T^a_{\alpha\beta},$$

where m is the number of modes, $S_{\alpha\beta}(f)$ are the elements of the PSD matrix of generalized displacements,

$$T^a_{\alpha\beta} = \left[\psi^a_\alpha\right]^T [A] \left[\psi^a_\beta\right],$$

$[\psi^a_\alpha]$ are the modal stress components of the αth mode at node *a*, and the constant matrix $\boldsymbol{A}$ is given by

$$\begin{pmatrix} 1 & -1/2 & -1/2 & 0 & 0 & 0 \\ -1/2 & 1 & -1/2 & 0 & 0 & 0 \\ -1/2 & -1/2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix}.$$

Similarly, RMS of von Mises stress at a node *a* is computed as

$$RMISES^a(f) = \sqrt{\sum_{\beta=1}^{m} \sum_{\alpha=1}^{m} V_{\alpha\beta}(f) T^a_{\alpha\beta}},$$

where $V_{\alpha\beta}(f)$ are the elements of the variance matrix of generalized displacements.

Reference

- "Random response analysis," Section 6.3.11 of the Abaqus Analysis User's Guide

2.5.9 BASE MOTIONS IN MODAL-BASED PROCEDURES

Product: Abaqus/Standard

Structures subjected to ground motion by earthquakes or other excitations such as explosions or dynamic action of machinery are examples in which support motions may have to be considered in the analysis of dynamic response. For modal-based dynamic analyses with modal dynamic, steady-state dynamic, or random response steps, the support motions are simulated by prescribed excitations called base motions that are applied to the suppressed degrees of freedom. The suppressed degrees of freedom are grouped into one or more bases. Multiple bases are required if base motions cannot be described by a single set of rigid body motions. A common case is that of a bridge whose supports are subjected to the same earthquake record but with a time shift.

The degrees of freedom that are suppressed without being assigned to a named base make up the *primary* base, which typically is the only base if the motion can be described by a single set of rigid body motions. The suppressed degrees of freedom that are associated with named boundary conditions make up the *secondary* base or bases. Abaqus/Standard uses different approaches to handle primary and secondary base motions. The modal participation method is used for primary base motions, and the "big mass" method is used for secondary base motions. Multiple bases can be used only in modal dynamic and steady-state dynamic analyses.

Primary base motions

Let us consider structural motions *relative* to the base motion, u_b. The total response, $\{u_t\}$, of the dynamic system will now consist of the relative response, $\{u\}$, and the applied base motion excitation, $\{u_b\}$:

$$\{u_t\} = \{u\} + \{u_b\},$$

with similar expressions for velocities and accelerations. Substituting $\{u_t\}$ in the linearized equation of motion gives

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = -[M]\{\ddot{u}_b\}.$$

The base acceleration is converted into applied inertia loads $-[M]\{\ddot{u}_b\}$. Here it has been assumed that there is no damping on rigid body modes (i.e., Rayleigh damping with $\alpha \neq 0$ is not allowed). If the prescribed excitation is given in the form of a displacement or a velocity, Abaqus/Standard differentiates it to obtain the acceleration. The base motion vector can be expressed in terms of the rigid body mode vectors, $\{T\}_j$, and time dependent base motion values, $z_j, j = 1, 2, ..., 6$:

$$\{u_b\} = \sum_{j=1}^{6} \{T\}_j z_j.$$

Projecting the equation of motion into the eigenspace we have

$$\ddot{q}_m + \frac{c_m}{m_m}\dot{q}_m + \frac{k_m}{m_m}q_m = -\sum_{j=1}^{6}(\Gamma_m)_j \ddot{z}_j,$$

where q_m and $\{\phi\}_m$ denote the relative generalized coordinate and mode shape for the mode m; k_m, c_m and m_m are modal stiffness, modal damping, and modal mass, respectively; and

$$(\Gamma_m)_j = \frac{1}{m_m}\{\phi\}_m^T[M]\{T\}_j$$

is the modal participation factor for mode m and degree of freedom j.

Kinematic boundary conditions defined without being assigned a base name in an eigenfrequency step cannot be changed in any of the subsequent modal-based procedures. The kinematic constraints are built into the eigenvectors and into the participation factors for each mode, which implies that all degrees of freedom in the primary base must be subjected to the same rigid body motion.

The participation factors are used to calculate the equivalent forcing function, and the equation of motion is solved for the relative quantities (such as relative displacements, relative velocities, and relative accelerations—output variables U, V, and A, respectively). To obtain total kinematic quantities (such as total displacements, total velocities, and total accelerations—output variables TU, TV, and TA, respectively), the primary base motions are added to the relative responses.

Secondary base motions

The base motion treatment described above cannot be applied to secondary bases. Instead, Abaqus/Standard uses a "big mass" approach to simulate the motion of secondary bases. In this approach a big mass (much bigger than the total mass of the structure) is added to each degree of freedom in a secondary base during the eigenfrequency step. This generates additional low frequency modes associated with the masses M_{big}. As more big masses are applied, more low frequency modes will be extracted in the frequency analysis step. To keep the number of frequencies of interest the same, the number of eigenvalues extracted is automatically increased. Hence, the size of the subspace will grow proportionally to the number of degrees of freedom associated with secondary bases.

The desired base motion is obtained by applying a point force to each degree of freedom in the modal superposition step:

$$P_s{}^N = M_{big}\ddot{u}_s^N,$$

where M_{big} is the big mass and $\ddot{u}_s^N$ is the applied acceleration prescribed for degree of freedom N associated with secondary supports.

Using the notation used in the equation of motion for primary base motions, the equation of motion for combined primary and secondary base motions is readily written as

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = -[M]\{\ddot{u}_b\} + \{P_s\},$$

with

$$\{P_s\} = \sum_i [M^i_{big}]\{\ddot{u}^i_s\},$$

where $[M^i_{big}]$ is the diagonal matrix containing the big masses for secondary base i and $\{\ddot{u}^i_s\}$ is the base motion applied to this base. The mass matrix $[M]$ now contains the mass of the structure as well as the big masses associated with the secondary bases. Projecting the equation of motion into the eigenspace (expanded by the low frequency modes) we obtain

$$\ddot{q}_m + \frac{c_m}{m_m}\dot{q}_m + \frac{k_m}{m_m}q_m = -\sum_{j=1}^{6}(\Gamma_m)_j\ddot{z}_j + \frac{1}{m_m}\{\phi\}^T_m\{P_s\}.$$

Again, the quantities solved for are relative to the primary support, including those obtained at the secondary supports.

The big masses should be chosen as large as possible to obtain accurate base motions but should not be so large as to cause excessive round-off errors or overflows. To provide six digits of numerical accuracy, Abaqus/Standard chooses each big mass equal to 10^6 times the total mass of the structure and each big rotary inertia equal to 10^6 times the total moment of inertia of the structure.

For acoustic pressure degree of freedom the big mass is calculated as equal to 10^6 times the total acoustic mass of the model. Secondary base motions for acoustic pressure are available only with the SIM architecture.

The big masses, which are introduced in the eigenfrequency step, are not included in the model for other steps in a multiple step analysis. Hence, the total mass of the structure and the printed messages about masses and inertia of the entire model are not affected. However, the presence of the masses will be noticeable in the output tables printed in the eigenvalue extraction step, as well as in the information for the generalized masses and effective masses. See "Double cantilever subjected to multiple base motions," Section 1.4.12 of the Abaqus Benchmarks Guide, for an example of the use of the base motion feature.

References

- "Transient modal dynamic analysis," Section 6.3.7 of the Abaqus Analysis User's Guide
- "Mode-based steady-state dynamic analysis," Section 6.3.8 of the Abaqus Analysis User's Guide
- "Random response analysis," Section 6.3.11 of the Abaqus Analysis User's Guide
- "Amplitude curves," Section 34.1.2 of the Abaqus Analysis User's Guide

2.6 Complex harmonic oscillations

- "Direct steady-state dynamic analysis," Section 2.6.1
- "Subspace-based steady-state dynamic analysis," Section 2.6.2

2.6.1 DIRECT STEADY-STATE DYNAMIC ANALYSIS

Product: Abaqus/Standard

For structures subjected to continuous harmonic excitation, Abaqus/Standard offers a "direct" steady-state dynamic analysis procedure in addition to the "modal" procedure described in "Steady-state linear dynamic analysis," Section 2.5.7, and the "subspace" procedure described in "Subspace-based steady-state dynamic analysis," Section 2.6.2. This procedure is a perturbation procedure, where the perturbed solution is obtained by linearization about the current base state. For the calculation of the base state the structure may exhibit material and geometrical nonlinear behavior as well as contact nonlinearities. Structural and viscous damping can be included in the procedure using the Rayleigh and structural damping coefficients specified under the material definition. Discrete damping such as mass, dashpot, spring, and connector elements can be included. In addition, global damping coefficients α_g, β_g, and s_g can be specified at the procedure level to define additional viscous and structural damping contributions. The procedure can also be used for coupled acoustic-structural medium analysis ("Coupled acoustic-structural medium analysis," Section 2.9.1), piezoelectric medium analysis ("Piezoelectric analysis," Section 2.10.1), and viscoelastic material modeling ("Frequency domain viscoelasticity," Section 4.8.3). All properties can be frequency dependent.

The formulation is based on the dynamic virtual work equation,

$$\int_V \rho\, \delta\mathbf{u} \cdot \ddot{\mathbf{u}}\, dV + \int_V \rho\alpha_c \delta\mathbf{u} \cdot \dot{\mathbf{u}}\, dV + \int_V \delta\boldsymbol{\varepsilon} : \boldsymbol{\sigma}\, dV - \int_{S_t} \delta\mathbf{u} \cdot \mathbf{t}\, dS = 0, \tag{2.6.1–1}$$

where $\dot{\mathbf{u}}$ and $\ddot{\mathbf{u}}$ are the velocity and the acceleration, ρ is the density of the material, α_c is the mass proportional damping factor (part of the Rayleigh damping assumption), $\boldsymbol{\sigma}$ is the stress, $\mathbf{t}$ is the surface traction, and $\delta\boldsymbol{\varepsilon}$ is the strain variation that is compatible with the displacement variation $\delta\mathbf{u}$. The discretized form of this equation is

$$\delta u^N \left\{ M^{NM}\ddot{u}^M + C^{NM}_{(m)}\dot{u}^M + I^N - P^N \right\} = 0, \tag{2.6.1–2}$$

where the following definitions apply:

$$\begin{aligned}
M^{NM} &= \int_V \rho \mathbf{N}^N \cdot \mathbf{N}^M\, dV && \text{is the mass matrix,} \\
C^{NM}_{(m)} &= \int_V \rho\alpha_c \mathbf{N}^N \cdot \mathbf{N}^M\, dV + \alpha_g M^{NM} && \text{is the mass proportional damping matrix,} \\
I^N &= \int_V \boldsymbol{\beta}^N : \boldsymbol{\sigma}\, dV && \text{is the internal load vector, and} \\
P^N &= \int_{S_t} \mathbf{N}^N \cdot \mathbf{t}\, dS && \text{is the external load vector.}
\end{aligned} \tag{2.6.1–3}$$

For the steady-state harmonic response we assume that the structure undergoes small harmonic vibrations about a deformed, stressed state, defined by the subscript $\boldsymbol{0}$. Since steady-state dynamics is a perturbation

procedure, the load and response in the step define the change from the base state. The change in internal force vector follows by linearization:

$$\Delta I^N = \int_V \left[\Delta \boldsymbol{\beta}^N : \boldsymbol{\sigma} + \boldsymbol{\beta}^N : \Delta \boldsymbol{\sigma} \right] \, dV.$$

The change in stress can be written in the form

$$\Delta \boldsymbol{\sigma} = \mathbf{D}^{el} : \left(\Delta \boldsymbol{\varepsilon} + is \Delta \boldsymbol{\varepsilon} + \beta_c \Delta \dot{\boldsymbol{\varepsilon}} \right),$$

where $\mathbf{D}^{el}$ is the elasticity matrix for the material, β_c is the stiffness proportional damping factor (the other part of the Rayleigh damping assumption), and s is the structural damping factor that forms the imaginary part of the stiffness matrix (known as the structural damping matrix). The strain and strain rate changes follow from the displacement and velocity changes:

$$\Delta \boldsymbol{\varepsilon} = \boldsymbol{\beta}^M \, \Delta u^M, \qquad \Delta \dot{\boldsymbol{\varepsilon}} = \boldsymbol{\beta}^M \, \Delta \dot{u}^M.$$

This allows us to write Equation 2.6.1–2 as

$$\delta u^N \left\{ M^{NM} \, \ddot{u}^M + (C^{NM}_{(m)} + C^{NM}_{(k)}) \dot{u}^M + (iC^{NM}_{(s)} + K^{NM}) \, u^M - P^N \right\} = 0, \tag{2.6.1–4}$$

where we have defined the stiffness matrix

$$K^{NM} = \int_V \left[\frac{\partial \boldsymbol{\beta}^N}{\partial u^M} : \boldsymbol{\sigma}_0 + \boldsymbol{\beta}^N : \mathbf{D}^{el} : \boldsymbol{\beta}^M \right] \, dV,$$

the stiffness proportional viscous damping matrix

$$C^{NM}_{(k)} = \int_V \beta_c \, \boldsymbol{\beta}^N : \mathbf{D}^{el} : \boldsymbol{\beta}^M \, dV + \beta_g K^{NM},$$

and the stiffness proportional structural damping matrix

$$C^{NM}_{(s)} = \int_V s \, \boldsymbol{\beta}^N : \mathbf{D}^{el} : \boldsymbol{\beta}^M \, dV + s_g K^{NM}.$$

For harmonic excitation and response we can write

$$\Delta u^M = \left(\Re \left(u^M \right) + i \, \Im \left(u^M \right) \right) \exp i\Omega t,$$

and

$$\Delta P^N = \left(\Re \left(P^N \right) + i \, \Im \left(P^N \right) \right) \exp i\Omega t,$$

where $\Re \left(u^M \right)$ and $\Im \left(u^M \right)$ are the real and imaginary parts of the amplitudes of the displacement, $\Re \left(P^N \right)$ and $\Im \left(P^N \right)$ are the real and imaginary parts of the amplitude of the force applied to the structure, and Ω is

the circular frequency. Substituting the expressions for harmonic excitation and response in Equation 2.6.1–4 and writing the result in matrix form yields

$$\begin{bmatrix} \Re\left[A^{NM}\right] & \Im\left[A^{NM}\right] \\ \Im\left[A^{NM}\right] & -\Re\left[A^{NM}\right] \end{bmatrix} \begin{Bmatrix} \Re\left(u^{M}\right) \\ \Im\left(u^{M}\right) \end{Bmatrix} = \begin{Bmatrix} \Re\left(P^{N}\right) \\ -\Im\left(P^{N}\right) \end{Bmatrix}, \tag{2.6.1–5}$$

where

$$\Re\left[A^{NM}\right] = K^{NM} - \Omega^2 M^{NM},$$
$$\Im\left[A^{NM}\right] = -\Omega(C^{NM}_{(m)} + C^{NM}_{(k)}) - C^{NM}_{(s)}.$$

Both the real and imaginary parts of A^{NM} are symmetric.

The procedure is activated by defining a direct-solution steady-state dynamic analysis step. Both real and imaginary loads can be defined.

As output Abaqus/Standard provides amplitudes and phases for all element and nodal variables at the requested frequencies. For this procedure all amplitude references must be given in the frequency domain.

Reference

- "Direct-solution steady-state dynamic analysis," Section 6.3.4 of the Abaqus Analysis User's Guide

2.6.2 SUBSPACE-BASED STEADY-STATE DYNAMIC ANALYSIS

Product: Abaqus/Standard

For structures subjected to continuous harmonic excitation, Abaqus/Standard offers a "subspace" steady-state dynamic analysis procedure in addition to the "modal" procedure described in "Steady-state linear dynamic analysis," Section 2.5.7, and the "direct" procedure described in "Direct steady-state dynamic analysis," Section 2.6.1. The procedure is activated by defining a subspace-based steady-state dynamic analysis step. This procedure is a perturbation procedure, where the perturbed solution is obtained by linearization about the current base state. For the calculation of the base state the structure may exhibit material and geometrical nonlinear behavior as well as contact nonlinearities. Structural and viscous damping can be included in the procedure using the Rayleigh and structural damping coefficients specified under the material definition. The procedure can also be used for viscoelastic material modeling ("Frequency domain viscoelasticity," Section 4.8.3). Discrete damping such as mass, dashpots, spring, and connector elements can be included. The main advantage of this method is that it allows frequency-dependent behavior to be considered at a relatively small cost increase over the purely linear analysis via the "modal" procedure described in "Steady-state linear dynamic analysis," Section 2.5.7.

The discretized form of the linearized dynamic virtual work equation for a solid or structural system can be written as

$$\delta u^N \left\{ M^{NM} \ddot{u}^M + C^{NM} \dot{u}^M + (K^{NM} + iK_s^{NM}) u^M - P^N \right\} = 0, \tag{2.6.2–1}$$

where the following definitions apply:

$$M^{NM} = \int_V \rho \mathbf{N}^N \cdot \mathbf{N}^M \, dV \qquad \text{is the mass matrix,}$$

$$K^{NM} = \int_V \left[\frac{\partial \boldsymbol{\beta}^N}{\partial u^M} : \boldsymbol{\sigma}_0 + \boldsymbol{\beta}^N : \mathbf{D}^{el} : \boldsymbol{\beta}^M \right] dV \qquad \text{is the stiffness matrix,}$$

$$P^N = \int_{S_t} \mathbf{N}^N \cdot \mathbf{t} \, dS \qquad \text{is the external load vector,}$$

C^{NM} is the viscous damping matrix, K_s^{NM} is the imaginary stiffness proportional damping matrix representing a structural damping matrix, $\dot{\mathbf{u}}$ and $\ddot{\mathbf{u}}$ are the velocity and the acceleration, ρ is the density of the material, $\boldsymbol{\sigma}_0$ is the stress in the base state, $\mathbf{t}$ is the surface traction, and $\mathbf{D}^{el}$ is the elasticity matrix for the material. We assume that both the stiffness K^{NM} and the damping C^{NM} are frequency dependent.

For acoustic media, similar equations apply (see "Coupled acoustic-structural medium analysis," Section 2.9.1):

$$\delta \hat{p}^N \left\{ M_f^{NM} \ddot{p}^M + C_f^{NM} \dot{p}^M + (K_f^{NM} + iC_{\mathrm{f(s)}}^{NM}) p^M - P_f^N \right\} = 0. \tag{2.6.2–2}$$

The "structural damping" operator in this acoustic case is defined as the extension of structural damping to the acoustic matrices:

$$C_{\mathrm{f(s)}}^{NM} \equiv s_g K_f^{NM};$$

that is, the acoustic stiffness matrix times a constant. The acoustic "structural damping" operator inherits the frequency dependence of the acoustic stiffness matrix caused by volumetric drag.

The eigenfrequency step prior to the subspace-based steady-state dynamic analysis has extracted n_{eig} eigenmodes of the undamped system using

$$(-\omega^2 M^{MN} + K^{MN})\phi^N = 0,$$

where ω is the eigenfrequency in radians/time. The procedure assumes that the complex displacement changes for the damped system can be written in the form

$$u^M = \phi_\beta^M a_\beta, \quad \beta = 1, ..., n_{eig}, \tag{2.6.2–3}$$

where a_β are the complex modal amplitudes. Using Equation 2.6.2–3 in Equation 2.6.2–1 and premultiplying with ϕ_α, the equation of motion projected onto the subspace is provided:

$$M_{\alpha\beta}\ddot{a}_\beta + C_{\alpha\beta}(\Omega)\dot{a}_\beta + (K_{\alpha\beta}(\Omega) + iK_{\alpha\beta}^s(\Omega))a_\beta = P_\alpha(\Omega) \qquad \alpha, \beta = 1, ..., n_{eig}, \tag{2.6.2–4}$$

where

$$\begin{aligned} M_{\alpha\beta} &= \phi_\alpha^N M^{NM} \phi_\beta^M, \\ K_{\alpha\beta}(\Omega) &= \phi_\alpha^N K^{NM}(\Omega) \phi_\beta^M, \\ K_{\alpha\beta}^s(\Omega) &= \phi_\alpha^N K_s^{NM}(\Omega) \phi_\beta^M, \\ C_{\alpha\beta}(\Omega) &= \phi_\alpha^N C^{NM}(\Omega) \phi_\beta^M, \\ P_\alpha(\Omega) &= \phi_\alpha^N P^N(\Omega), \end{aligned}$$

and Ω is the excitation frequency. Since the eigenmodes are not orthogonal to the damping and stiffness matrices in the equilibrium equations (because of the frequency-dependent properties), the projected damping and stiffness matrices are not diagonal.

For harmonic excitation and response we can write

$$a_\beta = (\Re(a_\beta) + i\,\Im(a_\beta)) \exp i\Omega t \quad \text{and} \quad P_\alpha = (\Re(P_\alpha) + i\,\Im(P_\alpha)) \exp i\Omega t,$$

where $\Re(a_\beta)$ and $\Im(a_\beta)$ are the real and imaginary parts of the modal amplitudes and $\Re(P_\alpha)$ and $\Im(P_\alpha)$ are the real and imaginary parts of the amplitude of the force applied to the structure after projection onto the subspace. Substituting the expressions for harmonic excitation and response in Equation 2.6.2–4 and writing the result in matrix form yields

$$\begin{bmatrix} \Re\left[A_{\alpha\beta}\right] & \Im\left[A_{\alpha\beta}\right] \\ \Im\left[A_{\alpha\beta}\right] & -\Re\left[A_{\alpha\beta}\right] \end{bmatrix} \begin{Bmatrix} \Re\left(a_\beta\right) \\ \Im\left(a_\beta\right) \end{Bmatrix} = \begin{Bmatrix} \Re\left(P_\alpha\right) \\ -\Im\left(P_\alpha\right) \end{Bmatrix},$$

where

$$\Re\left[A_{\alpha\beta}\right] = K_{\alpha\beta} - \Omega^2 M_{\alpha\beta},$$
$$\Im\left[A_{\alpha\beta}\right] = -\Omega C_{\alpha\beta} - K^s_{\alpha\beta}.$$

The equation is solved for the real and imaginary part of the complex modal amplitudes a_β, and Equation 2.6.2–3 can be used to compute the real and imaginary part of the nodal displacements.

For a coupled acoustic-structural analysis the development is similar. However, in this case the "damping" matrix also contains volumetric drag, impedance boundary, radiating boundary, and fluid-solid coupling effects. Although the original formulation of the coupled acoustic-structural eigenvalue extraction problem is unsymmetric, the equations are rearranged such that the fluid-solid coupling effects are retained without making the problem unsymmetric. The resulting modes and frequencies are associated with the coupled, undamped fluid-solid system. The subspace-based steady-state dynamic procedure will project the full, coupled, and damped acoustic-structural system of equations onto the space spanned by the set of coupled fluid-solid modes.

For an acoustic-only analysis the development is very similar to that of the solid-only case. The acoustic material damping equivalent, volumetric drag, is treated in a similar manner to solid material damping effects.

Output

As output Abaqus/Standard provides amplitudes and phases for all element and nodal variables at the requested frequencies. All amplitude references must be given in the frequency domain.

Reference

- "Subspace-based steady-state dynamic analysis," Section 6.3.9 of the Abaqus Analysis User's Guide

2.7 Steady-state transport analysis

- “Steady-state transport analysis,” Section 2.7.1

2.7.1 STEADY-STATE TRANSPORT ANALYSIS

Product: Abaqus/Standard

Abaqus/Standard provides a specialized analysis capability to model the steady-state behavior of a cylindrical deformable body rolling along a flat rigid surface. The capability uses a reference frame that removes the explicit time dependence from the problem so that a purely spatially dependent analysis can be performed. For an axisymmetric body traveling at a constant ground velocity and constant angular rolling velocity, a steady state is possible in a frame that moves at the speed of the ground velocity but does not spin with the body in the rolling motion. This choice of reference frame allows the finite element mesh to remain stationary so that only the part of the body in the contact zone requires fine meshing.

Kinematics of steady-state rolling

The kinematics of the rolling problem are described in terms of a coordinate frame that moves along with the ground motion of the body. In this moving frame the rigid body rotation is described in a spatial or Eulerian manner and the deformation in a material or Lagrangian manner. It is this kinematic description that converts the steady moving contact field problem into a purely spatially dependent simulation.

We consider the case shown in Figure 2.7.1–1, where the ground velocity of the body is described in terms of a constant cornering motion.

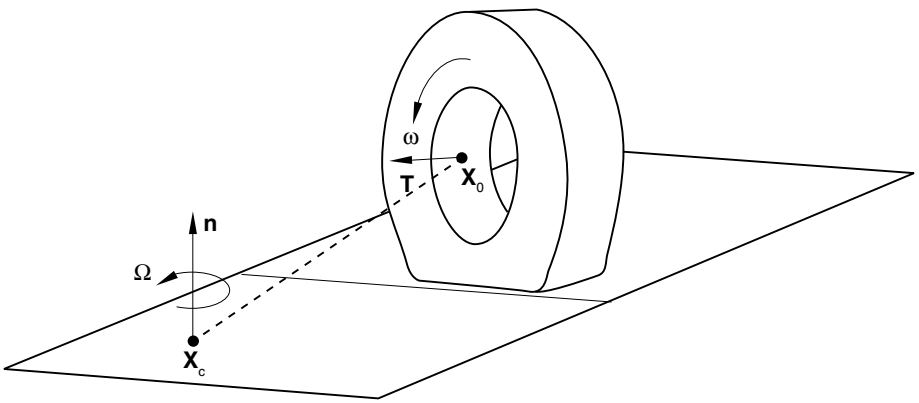


Figure 2.7.1–1 Constant cornering motion.

The body is rotating with a constant angular rolling velocity ω around a rigid axle $\mathbf{T}$ at $\mathbf{X}_0$, which in turn rotates with constant angular velocity Ω around the fixed cornering axis $\mathbf{n}$ through point $\mathbf{X}_c$. Hence, the motion of a particle $\mathbf{X}$ at time t consists of a rigid rolling rotation to position $\mathbf{Y}$, described by

$$\mathbf{Y} = \mathbf{R}_s \cdot (\mathbf{X} - \mathbf{X}_0) + \mathbf{X}_0,$$

followed by a deformation to point $\mathbf{x}$, and a subsequent cornering rotation (or precession) around $\mathbf{n}$ to position $\mathbf{y}$ so that

$$\mathbf{y} = \mathbf{R}_c \cdot (\mathbf{x} - \mathbf{X}_c) + \mathbf{X}_c,$$

where $\mathbf{R}_c$ is the cornering rotation given by $\mathbf{R}_c = \exp(\hat{\mathbf{\Omega}}\, t)$ and $\hat{\mathbf{\Omega}}$ is the skew-symmetric matrix associated with the rotation vector $\mathbf{\Omega} = \Omega\mathbf{n}$. Similarly, $\mathbf{R}_s$ is the spinning rotation matrix defined as $\mathbf{R}_s = \exp(\hat{\boldsymbol{\omega}}\, t)$ and $\hat{\boldsymbol{\omega}}$ is the skew-symmetric matrix associated with the rotation vector $\boldsymbol{\omega} = \omega\mathbf{T}$. The velocity of the particle then becomes

$$\mathbf{v} = \dot{\mathbf{y}} = \dot{\mathbf{R}}_c \cdot (\mathbf{x} - \mathbf{X}_c) \,+\, \mathbf{R}_c \cdot \dot{\mathbf{x}}.$$

To describe the deformation of the body, we define a map $\boldsymbol{\chi}(\mathbf{Y}, t)$, which gives the position of point $\mathbf{X}$ at time t as a function of its location $\mathbf{Y}$ at time t so that $\mathbf{x} = \boldsymbol{\chi}(\mathbf{Y}, t)$. It follows that

$$\dot{\mathbf{x}} = \frac{\partial \boldsymbol{\chi}}{\partial \mathbf{Y}} \cdot \frac{\partial \mathbf{Y}}{\partial t} \,+\, \frac{\partial \boldsymbol{\chi}}{\partial t},$$

where

$$\frac{\partial \mathbf{Y}}{\partial t} = \dot{\mathbf{R}}_s \cdot (\mathbf{X} - \mathbf{X}_0) = \omega\mathbf{T} \times (\mathbf{Y} - \mathbf{X_0}).$$

Noting that $\dot{\mathbf{R}}_s = \hat{\boldsymbol{\omega}} \cdot \mathbf{R}_s = \omega\hat{\mathbf{T}} \cdot \mathbf{R}_s$, and introducing the circumferential direction $\mathbf{S} = \mathbf{T} \times (\mathbf{Y} - \mathbf{X}_0)/R$, where $R = |\mathbf{Y} - \mathbf{X}_0|$ is the radius of a point on the reference body, the velocity of the reference body can be written as $\partial \mathbf{Y}/\partial t = \omega R\, \mathbf{S}$, so that

$$\dot{\mathbf{x}} = \omega R \frac{\partial \boldsymbol{\chi}}{\partial \mathbf{Y}} \cdot \mathbf{S} \,+\, \frac{\partial \boldsymbol{\chi}}{\partial t} = \omega R \frac{\partial \boldsymbol{\chi}}{\partial S} \,+\, \frac{\partial \boldsymbol{\chi}}{\partial t},$$

where S is the distance-measuring coordinate along the streamline. Using this result, together with $\dot{\mathbf{R}}_c = \hat{\mathbf{\Omega}} \cdot \mathbf{R}_c = \Omega\hat{\mathbf{n}} \cdot \mathbf{R}_c$, the velocity of the particle can be written as

$$\mathbf{v} = \Omega\mathbf{n} \times (\mathbf{x} - \mathbf{X}_c) \,+\, \omega R\, \mathbf{R}_c \cdot \frac{\partial \boldsymbol{\chi}}{\partial S} \,+\, \mathbf{R}_c \cdot \frac{\partial \boldsymbol{\chi}}{\partial t}.$$

The acceleration is obtained by a second differentiation and some manipulation:

$$\begin{aligned}\mathbf{a} = \Omega^2(\mathbf{nn} - \mathbf{I}) \cdot (\mathbf{x} - \mathbf{X}_c) \,+\, 2\omega\Omega R\, \mathbf{n} \times \mathbf{R}_c \cdot \frac{\partial \boldsymbol{\chi}}{\partial S} \,+\, 2\Omega\, \mathbf{n} \times \mathbf{R}_c \cdot \frac{\partial \boldsymbol{\chi}}{\partial t} \\ + \omega^2 R^2 \mathbf{R}_c \cdot \frac{\partial^2 \boldsymbol{\chi}}{\partial S^2} \,+\, 2\omega R\, \mathbf{R}_c \cdot \frac{\partial^2 \boldsymbol{\chi}}{\partial S \partial t} \,+\, \mathbf{R}_c \cdot \frac{\partial^2 \boldsymbol{\chi}}{\partial t^2}.\end{aligned}$$

To obtain expressions for the velocity and acceleration in the reference frame tied to the body, we use the transformations

$$\mathbf{v}_r = \mathbf{R}_c^T \cdot \mathbf{v} \quad \text{and} \quad \mathbf{a}_r = \mathbf{R}_c^T \cdot \mathbf{a}$$

so that we obtain

$$\mathbf{v}_r = \Omega \mathbf{n} \times (\mathbf{x} - \mathbf{X}_c) + \omega R \frac{\partial \boldsymbol{\chi}}{\partial S} + \frac{\partial \boldsymbol{\chi}}{\partial t}$$

and

$$\mathbf{a}_r = \Omega^2 (\mathbf{nn} - \mathbf{I}) \cdot (\mathbf{x} - \mathbf{X}_c) + 2\omega\Omega R\, \mathbf{n} \times \frac{\partial \boldsymbol{\chi}}{\partial S} + 2\Omega\, \mathbf{n} \times \frac{\partial \boldsymbol{\chi}}{\partial t} + \omega^2 R^2 \frac{\partial^2 \boldsymbol{\chi}}{\partial S^2} + 2\omega R \frac{\partial^2 \boldsymbol{\chi}}{\partial S \partial t} + \frac{\partial^2 \boldsymbol{\chi}}{\partial t^2}.$$

For steady-state conditions these expressions reduce to

$$\mathbf{v}_r = \Omega\, \mathbf{n} \times (\mathbf{x} - \mathbf{X}_c) + \omega R \frac{\partial \boldsymbol{\chi}}{\partial S}$$

and

$$\mathbf{a}_r = \Omega^2 (\mathbf{nn} - \mathbf{I}) \cdot (\mathbf{x} - \mathbf{X}_c) + 2\omega\Omega R\, \mathbf{n} \times \frac{\partial \boldsymbol{\chi}}{\partial S} + \omega^2 R^2 \frac{\partial^2 \boldsymbol{\chi}}{\partial S^2}.$$

The first term in the last expression can be identified as the acceleration that gives rise to centrifugal forces resulting from rotation about $\mathbf{n}$. Noting that $\omega R \partial \boldsymbol{\chi}/\partial S$ is a measure of velocity, the second term can be identified as the acceleration that gives rise to Coriolis forces. The last term combines the acceleration that gives rise to Coriolis and centrifugal forces resulting from rotation about $\mathbf{T}$. When the deformation is uniform along the circumferential direction, this Coriolis effect vanishes so that the acceleration gives rise to centrifugal forces only.

The velocity of the center of the body $\mathbf{X}_0$ (which must lie on the axis $\mathbf{T}$) is

$$\mathbf{v}_0 = \Omega\, \mathbf{n} \times (\mathbf{X}_0 - \mathbf{X}_c)$$

since the motions due to rolling and deformation vanish on the axis.

To obtain the expression for straight line motion, as shown in Figure 2.7.1–2, we move $\mathbf{X}_c$ far away from the center of the body $\mathbf{x}_0$ but keep $\mathbf{v}_0$ the same. In that case $\Omega \to 0$ and, hence, in the limit

$$\mathbf{v} = \mathbf{v}_0 + \mathbf{v}_r = \mathbf{v}_0 + \omega R \frac{\partial \boldsymbol{\chi}}{\partial S},$$

$$\mathbf{a} = \mathbf{a}_r = \omega^2 R^2 \frac{\partial^2 \boldsymbol{\chi}}{\partial S^2},$$

which corresponds to straight line rolling.

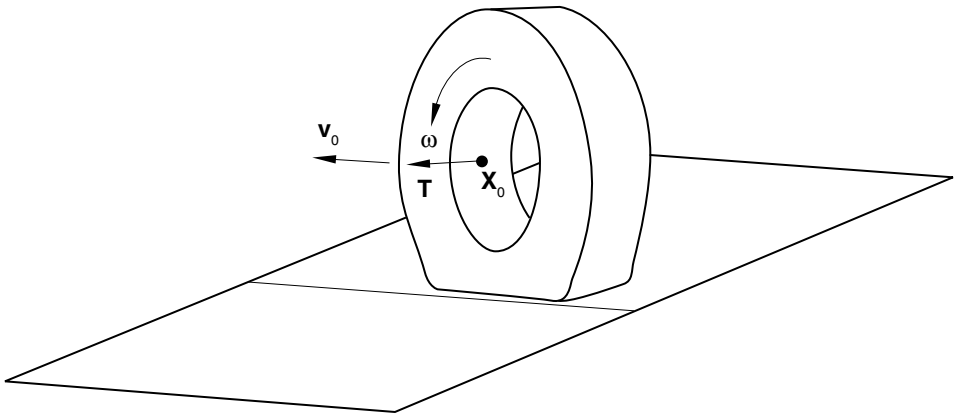


Figure 2.7.1–2 Straight line rolling.

Inertia

The virtual work contribution from the d'Alembert forces is

$$\delta\Pi = -\int_V \rho\,\mathbf{a}\cdot\delta\mathbf{v}\,dV.$$

Using the divergence theorem, the virtual work contribution becomes

$$\delta\Pi = -\Omega^2\int_V \rho(\mathbf{x}-\mathbf{X}_c)\cdot(\mathbf{nn}-\mathbf{I})\cdot\delta\mathbf{v}\,dV \;-\; 2\omega\Omega\int_V \rho R\mathbf{n}\times\frac{\partial\boldsymbol{\chi}}{\partial S}\cdot\delta\mathbf{v}\,dV + \omega^2\int_V \rho R^2\frac{\partial\boldsymbol{\chi}}{\partial S}\cdot\frac{\partial\delta\mathbf{v}}{\partial S}\,dV$$

and the rate of virtual work becomes

$$d\delta\Pi = -\Omega^2\int_V \rho d\mathbf{x}\cdot(\mathbf{nn}-\mathbf{I})\cdot\delta\mathbf{v}\,dV \;-\; 2\omega\Omega\int_V \rho R\mathbf{n}\times\frac{\partial d\boldsymbol{\chi}}{\partial S}\cdot\delta\mathbf{v}\,dV + \omega^2\int_V \rho R^2\frac{\partial d\boldsymbol{\chi}}{\partial S}\cdot\frac{\partial\delta\mathbf{v}}{\partial S}\,dV.$$

For straight line rolling only the last term in each expression needs to be taken into account.

Performing a harmonic analysis about a nonlinear base state

To perform a steady-state dynamic or frequency analysis on a rolling tire, it is necessary to linearize the virtual work expressions about the base state. Assuming a harmonic solution of the form $\mathbf{u}(\mathbf{Y})e^{j\psi t}$, it can be shown that, for the case of straight line rolling, the linearized rate of virtual work is

$$d\delta\Pi = \int_V \rho \left[\omega^2 R^2 \frac{\partial d\mathbf{u}}{\partial S} \cdot \frac{\partial \delta\mathbf{v}}{\partial S} + j\psi\omega R \left(d\mathbf{u} \cdot \frac{\partial \delta\mathbf{v}}{\partial S} - \frac{\partial d\mathbf{u}}{\partial S} \cdot \delta\mathbf{v} \right) + \psi^2 d\mathbf{u} \cdot \delta\mathbf{v} \right] dV.$$

The first term is the load stiffness contribution due to the spinning motion about the axle. The second term is an imaginary antisymmetric gyroscopic operator. The third term is the standard mass operator.

Contact conditions

To obtain the contact conditions, we start with the expressions for velocity derived in the previous section. For points on the surface of the deformable body

$$\mathbf{v}_D = \Omega\,\mathbf{n} \times (\mathbf{x} - \mathbf{X}_c) + \omega R \frac{\partial \boldsymbol{\chi}}{\partial S} + \frac{\partial \boldsymbol{\chi}}{\partial t},$$

where $\mathbf{n}$ is the cornering axis (which must be normal to the rigid surface) and Ω is the cornering angular velocity around $\mathbf{n}$. Assuming that the velocity of a point on the foundation (or rigid surface) is $\mathbf{v}_R$, the relative motion becomes

$$\mathbf{v} = \mathbf{v}_D - \mathbf{v}_R = \Omega\,\mathbf{n} \times \mathbf{r} + \omega R \frac{\partial \boldsymbol{\chi}}{\partial S} + \frac{\partial \boldsymbol{\chi}}{\partial t} - \mathbf{v}_R,$$

where $\mathbf{r} = \mathbf{x} - \mathbf{X}_c$. This equation can be split into normal and tangential components. The rate of penetration is

$$\dot{h} = -\mathbf{n} \cdot \mathbf{v} = \mathbf{n} \cdot \mathbf{v}_R - \omega R\,\mathbf{n} \cdot \frac{\partial \boldsymbol{\chi}}{\partial S} - \mathbf{n} \cdot \frac{\partial \boldsymbol{\chi}}{\partial t}.$$

For any point in contact $\mathbf{n} \cdot \partial\boldsymbol{\chi}/\partial S = 0$; hence,

$$\dot{h} = \mathbf{n} \cdot \mathbf{v}_R - \mathbf{n} \cdot \frac{\partial \boldsymbol{\chi}}{\partial t},$$

which in incremental form reduces to the standard contact condition

$$\Delta h = \mathbf{n} \cdot (\Delta\mathbf{x}_R - \Delta\mathbf{x}_D).$$

For steady-state conditions $\mathbf{n} \cdot \Delta\mathbf{x}_R = 0$ and $\Delta\mathbf{x}_D = 0$.

Similarly, the rate of slip is

$$\dot{\gamma}_\alpha = \mathbf{t}_\alpha \cdot \mathbf{v} = \Omega\,\mathbf{t}_\alpha \cdot (\mathbf{n} \times \mathbf{r}) + \omega R\,\mathbf{t}_\alpha \cdot \frac{\partial \boldsymbol{\chi}}{\partial S} + \mathbf{t}_\alpha \cdot \frac{\partial \boldsymbol{\chi}}{\partial t} - \mathbf{t}_\alpha \cdot \mathbf{v}_R,$$

where $\mathbf{t}_\alpha$ $(\alpha = 1, 2)$ are two orthogonal unit vectors tangent to the contact surface so that $\mathbf{n} = \mathbf{t}_1 \times \mathbf{t}_2$. For steady-state conditions $\partial\boldsymbol{\chi}/\partial t = 0$, so

$$\dot{\gamma}_\alpha = \mathbf{t}_\alpha \cdot \mathbf{v} = \Omega\,\mathbf{t}_\alpha \cdot (\mathbf{n} \times \mathbf{r}) + \omega R\,\mathbf{t}_\alpha \cdot \frac{\partial \boldsymbol{\chi}}{\partial S} - \mathbf{t}_\alpha \cdot \mathbf{v}_R.$$

Variations in $\dot{\gamma}_\alpha$ yield

$$\delta\dot{\gamma}_\alpha = \Omega\,\mathbf{t}_\alpha \cdot (\mathbf{n} \times \delta\mathbf{r}) + \omega R\,\mathbf{t}_\alpha \cdot \frac{\partial \delta\boldsymbol{\chi}}{\partial S} - \mathbf{t}_\alpha \cdot \delta\mathbf{v}_R.$$

For straight line rolling we can replace $\Omega\,\mathbf{n} \times \mathbf{r}$ by $\mathbf{v}_0$ so that we obtain

$$\dot{\gamma}_\alpha = \mathbf{t}_\alpha \cdot \mathbf{v}_0 + \omega R\,\mathbf{t}_\alpha \cdot \frac{\partial \boldsymbol{\chi}}{\partial S} - \mathbf{t}_\alpha \cdot \mathbf{v}_R$$

and

$$\delta\dot{\gamma}_\alpha = \omega R\,\mathbf{t}_\alpha \cdot \frac{\partial \delta\boldsymbol{\chi}}{\partial S} - \mathbf{t}_\alpha \cdot \delta\mathbf{v}_R.$$

To complete the formulation, a relationship between frictional stress and slip velocity must be developed. A Coulomb friction law is provided for steady-state rolling. The law assumes that slip occurs if the frictional stress,

$$\tau_{eq} = \sqrt{\tau_1^2 + \tau_2^2},$$

is equal to the critical stress, $\tau_{crit} = \mu p$, where τ_1 and τ_2 are shear stresses along $\mathbf{t}_\alpha$, μ is the friction coefficient, and $\boldsymbol{p}$ is the contact pressure. On the other hand, when $\tau_{eq} < \tau_{crit}$, no relative motion occurs. The condition of no relative motion is approximated in Abaqus by stiff viscous behavior

$$\tau_\alpha = \kappa_s\,\dot{\gamma}_\alpha^v,$$

where $\dot{\gamma}_\alpha$ is the tangential slip velocity and κ_s is the "stick viscosity," which follows from the relation

$$\kappa_s = \frac{\tau_{crit}}{\dot{\gamma}_{crit}}.$$

The allowable viscous slip velocity is defined as a fraction of the circumferential velocity

$$\dot{\gamma}_{crit} = 2F_f \omega R,$$

where F_f is a user-defined slip tolerance.

These expressions contribute to the standard virtual work contribution for slip,

$$\delta\Pi = \int_A \tau_\alpha\,\delta\gamma_\alpha\,dA,$$

and rate of virtual work for slip,

$$d\delta\Pi = \int_A (\mathrm{d}\tau_\alpha \delta\gamma_\alpha + \tau_\alpha\,\mathrm{d}\delta\gamma_\alpha) dA.$$

Reference

- "Steady-state transport analysis," Section 6.4.1 of the Abaqus Analysis User's Guide

2.8 Analysis of porous media

- "Effective stress principle for porous media," Section 2.8.1
- "Discretized equilibrium statement for a porous medium," Section 2.8.2
- "Constitutive behavior in a porous medium," Section 2.8.3
- "Continuity statement for the wetting liquid phase in a porous medium," Section 2.8.4
- "Solution strategy for coupled diffusion/deformation," Section 2.8.5

2.8.1 EFFECTIVE STRESS PRINCIPLE FOR POROUS MEDIA

Product: Abaqus/Standard

A porous medium is modeled in Abaqus/Standard by the conventional approach that considers the medium as a multiphase material and adopts an effective stress principle to describe its behavior. The porous medium modeling provided considers the presence of two fluids in the medium. One is the "wetting liquid," which is assumed to be relatively (but not entirely) incompressible. Often the other is a gas, which is relatively compressible. An example of such a system is soil containing ground water. When the medium is partially saturated, both fluids exist at a point: when it is fully saturated, the voids are completely filled with the wetting liquid. The elementary volume, dV, is made up of a volume of grains of solid material, dV_g; a volume of voids, dV_v; and a volume of wetting liquid, $dV_w \leq dV_v$, that is free to move through the medium if driven. In some systems (for example, systems containing particles that absorb the wetting liquid and swell in the process) there may also be a significant volume of trapped wetting liquid, dV_t.

The total stress acting at a point, $\boldsymbol{\sigma}$, is assumed to be made up of an average pressure stress in the wetting liquid, u_w, called the "wetting liquid pressure," an average pressure stress in the other fluid, u_a, and an "effective stress," $\overline{\boldsymbol{\sigma}}^*$, defined by

$$\overline{\boldsymbol{\sigma}}^* = \boldsymbol{\sigma} + \big(\chi u_w + (1-\chi)u_a\big)\mathbf{I}. \tag{2.8.1–1}$$

Stress components are stored so that tensile stress is positive, but u_w and u_a are pressure stress values. This explains the sign in this equation. χ is a factor that depends on saturation and on the surface tension of the liquid/solid system (Wu, 1976). χ is 1.0 when the medium is fully saturated and between 0.0 and 1.0 in unsaturated systems when its value depends on the degree of saturation of the medium. There is very sparse experimental evidence of its dependence on saturation; and because of this lack of data, we simply assume that χ is equal to the saturation of the medium (we define saturation later in this section).

We simplify the model by assuming that the pressure applied to the nonwetting fluid is constant throughout the domain being modeled, does not vary with time, and is small enough that its value can be neglected. This requires that the nonwetting fluid can diffuse through the medium sufficiently freely so that its pressure, u_a, never exceeds the pressure applied to this fluid at the boundaries of the medium, which remains constant throughout the process being modeled. The most common example where this simplification applies is a porous medium that is quite permeable to gas flow, in which the nonwetting fluid is air, exposed to atmospheric pressure. The dimensions of the region modeled must not be so large that the gravitational gradient of atmospheric pressure causes a significant change in the air pressure, and there can be no external event that provides a transient change in the air pressure. This assumption allows u_a to be removed from the equation, provided that the corresponding loading term (for example, atmospheric pressure on the boundary of the medium) is also omitted from the equilibrium equations and that u_a is small enough that its effect on the deformation of the medium is not important (or that deformation is measured from the state $\boldsymbol{\sigma} = -u_a\mathbf{I}$). This simplification reduces the effective stress principle to

$$\overline{\boldsymbol{\sigma}}^* = \boldsymbol{\sigma} + \chi u_w \mathbf{I}. \tag{2.8.1–2}$$

In the case where trapped fluid is present in the system, we assume that the effective stress is made up of two components weighted according to the relative volume of trapped fluid and porous material:

$$\overline{\boldsymbol{\sigma}}^* = (1 - n_t)\overline{\boldsymbol{\sigma}} - n_t \overline{p}_t \mathbf{I}, \tag{2.8.1–3}$$

where $\overline{\boldsymbol{\sigma}}$ is the effective stress in the porous material skeleton, $\overline{p}_t$ is the average pressure stress in the trapped liquid, and n_t is the ratio of trapped fluid volume to total volume, as defined later in this section.

We assume that the constitutive response of the porous medium consists of simple bulk elasticity relationships for the liquid and for the soil grains, together with a constitutive theory for the soil skeleton whereby $\overline{\boldsymbol{\sigma}}$ is defined as a function of the strain history and temperature of the soil:

$$\overline{\boldsymbol{\sigma}} = \overline{\boldsymbol{\sigma}}(\text{strain history, temperature, state variables}).$$

Constitutive models that are appropriate for voided materials, such as soils, are described in Chapter 4, "Mechanical Constitutive Theories."

The remaining parts of "Analysis of porous media," Section 2.8, discuss the equilibrium equation for porous media ("Discretized equilibrium statement for a porous medium," Section 2.8.2), fundamental constitutive assumptions that incorporate the effective stress principle as outlined above ("Constitutive behavior in a porous medium," Section 2.8.3), and the continuity equation that governs the flow of the wetting liquid ("Continuity statement for the wetting liquid phase in a porous medium," Section 2.8.4). Newton's method is generally used to solve the governing equations for the implicit time integration procedure. Analysis of small, linearized, perturbations about a deformed state is also sometimes required (for vibration studies, for example). For these reasons the development includes a definition of the form of the Jacobian matrix for the two-phase model.

As preliminaries, porosity, void ratio, and saturation are defined. The porosity of the medium, n, is the ratio of the volume of voids to the total volume:

$$n \overset{\text{def}}{=} \frac{dV_v}{dV} = 1 - \frac{dV_g}{dV} - \frac{dV_t}{dV}.$$

Using the superscript 0 to indicate values in some convenient reference configuration allows the porosity in the current configuration to be expressed as

$$\begin{aligned} n &= 1 - \frac{dV_g}{dV_g^0}\frac{dV^0}{dV}\frac{dV_g^0}{dV^0} - \frac{dV_t}{dV} \\ &= 1 - J_g J^{-1}(1 - n^0 - n_t^0) - n_t, \end{aligned}$$

so that

$$\frac{1 - n - n_t}{1 - n^0 - n_t^0} = \frac{J_g}{J}, \tag{2.8.1–4}$$

where

$$J \stackrel{\text{def}}{=} \left| \frac{dV}{dV^0} \right|$$

is the ratio of the medium's volume in the current configuration to its volume in the reference configuration,

$$J_g \stackrel{\text{def}}{=} \left| \frac{dV_g}{dV_g^0} \right|$$

is the ratio of the current to reference volume for the grains, and

$$n_t \stackrel{\text{def}}{=} \frac{dV_t}{dV}$$

is the volume of trapped wetting liquid per unit of current volume.

Abaqus generally uses void ratio, $e \stackrel{\text{def}}{=} dV_v/(dV_g + dV_t)$, instead of porosity. Conversion relationships are readily derived as

$$e = \frac{n}{1-n}, \quad n = \frac{e}{1+e}, \quad 1-n = \frac{1}{1+e}. \tag{2.8.1–5}$$

Saturation, s, is the ratio of free (untrapped) wetting liquid volume to void volume:

$$s \stackrel{\text{def}}{=} \frac{dV_w}{dV_v}.$$

The volume ratio of free wetting liquid at a point is

$$n_w \stackrel{\text{def}}{=} \frac{dV_w}{dV} = sn.$$

The total volume of wetting liquid (free liquid plus trapped liquid) per unit of current volume is

$$n_f = sn + n_t.$$

References

- "Coupled pore fluid diffusion and stress analysis," Section 6.8.1 of the Abaqus Analysis User's Guide
- "Geostatic stress state," Section 6.8.2 of the Abaqus Analysis User's Guide

2.8.2 DISCRETIZED EQUILIBRIUM STATEMENT FOR A POROUS MEDIUM

Product: Abaqus/Standard

Equilibrium is expressed by writing the principle of virtual work for the volume under consideration in its current configuration at time t:

$$\int_V \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon}\, dV = \int_S \mathbf{t} \cdot \delta\mathbf{v}\, dS + \int_V \hat{\mathbf{f}} \cdot \delta\mathbf{v}\, dV,$$

where $\delta\mathbf{v}$ is a virtual velocity field, $\delta\boldsymbol{\varepsilon} \stackrel{\text{def}}{=} \text{sym}(\partial\delta\mathbf{v}/\partial\mathbf{x})$ is the virtual rate of deformation, $\boldsymbol{\sigma}$ is the true (Cauchy) stress, $\mathbf{t}$ are surface tractions per unit area, and $\hat{\mathbf{f}}$ are body forces per unit volume.

For our system $\hat{\mathbf{f}}$ will often include the weight of the wetting liquid,

$$\mathbf{f}_w = (sn + n_t)\rho_w \mathbf{g},$$

where ρ_w is the density of the wetting liquid and $\mathbf{g}$ is the gravitational acceleration, which we assume to be constant and in a constant direction (so that, for example, the formulation cannot be applied directly to a centrifuge experiment unless the model in the machine is small enough that $\mathbf{g}$ can be treated as constant). For simplicity we consider this loading explicitly so that any other gravitational term in $\hat{\mathbf{f}}$ is associated only with the weight of the dry porous medium. Thus, we write the virtual work equation as

$$\int_V \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon}\, dV = \int_S \mathbf{t} \cdot \delta\mathbf{v}\, dS + \int_V \mathbf{f} \cdot \delta\mathbf{v}\, dV + \int_V (sn + n_t)\rho_w \mathbf{g} \cdot \delta\mathbf{v}\, dV, \tag{2.8.2–1}$$

where $\mathbf{f}$ are all body forces except the weight of the wetting liquid.

In a finite element model equilibrium is approximated as a finite set of equations by introducing interpolation functions. The notation used to indicate such discretization are those quantities with uppercase superscripts (for example, v^N), which represent nodal variables, with the summation convention adopted for the superscripts. The interpolation is assumed to be based on material coordinates in the material skeleton (a "Lagrangian" formulation).

For simplicity, in this section we consider only the case where the problem has no internal constraints—such as incompressibility—and the discretization is made entirely by approximating equilibrium: this results in the displacement (or stiffness) method. Mixed formulation ("hybrid") elements are available for porous medium analysis with Abaqus/Standard, but consideration of such formulations does not require any important extension of the development at this stage.

The virtual velocity field is interpolated by

$$\delta\mathbf{v} = \mathbf{N}^N \delta v^N,$$

where $\mathbf{N}^N(S_i)$ are interpolation functions defined with respect to material coordinates, S_i.

The virtual rate of deformation is interpolated as

$$\delta\boldsymbol{\varepsilon} = \boldsymbol{\beta}^N \delta v^N,$$

where, in the simplest case,

$$\boldsymbol{\beta}^N = \operatorname{sym}\left(\frac{\partial \delta \mathbf{N}^N}{\partial \mathbf{x}}\right),$$

although more general forms are used in some of the elements in Abaqus.

The virtual work equation is thus discretized as

$$\delta v^N \int_V \boldsymbol{\beta}^N : \boldsymbol{\sigma}\, dV = \delta v^N \left[\int_S \mathbf{N}^N \cdot \mathbf{t}\, dS + \int_V \mathbf{N}^N \cdot \mathbf{f}\, dV + \int_V (sn + n_t)\rho_w \mathbf{N}^N \cdot \mathbf{g}\, dV\right],$$

where the δv^N are assumed to be independent.

The term conjugate to δv^N on the left-hand side of this equation is referred to subsequently as the internal force array, I^N:

$$I^N \stackrel{\text{def}}{=} \int_V \boldsymbol{\beta}^N : \boldsymbol{\sigma}\, dV.$$

Likewise, the external force array, P^N, is taken from the right-hand side:

$$P^N \stackrel{\text{def}}{=} \int_S \mathbf{N}^N \cdot \mathbf{t}\, dS + \int_V \mathbf{N}^N \cdot \mathbf{f}\, dV + \int_V (sn + n_t)\rho_w \mathbf{N}^N \cdot \mathbf{g}\, dV$$

(P^N includes any d'Alembert forces).

Choosing each δv^N to be nonzero in turn expresses equilibrium as a balance of internal and external forces:

$$I^N - P^N = 0. \tag{2.8.2–2}$$

These discretized equilibrium equations, together with the continuity equation discussed in "Continuity statement for the wetting liquid phase in a porous medium," Section 2.8.4, define the state of the porous medium. The equilibrium equations are written at the end of a time increment when implicit integration is used and, for all but the simplest cases, they are nonlinear. Newton's method is often used for their solution. Also, small, linear perturbations of the system are sometimes of interest (an example is the small vibration problem). These considerations imply a need for the Jacobian matrix of the system, which defines the variation of each term in the equations with respect to the basic variables of the discretized problem, which—for this case—are the nodal positions, x^N (or, equivalently, the displacements $x^N - X^N$), and the nodal wetting liquid pressure values, u_w^N. Symbolically we write such a variation of a term, $\boldsymbol{f}$ say, as df, meaning

$$df \stackrel{\text{def}}{=} \frac{\partial f}{\partial x^N} dx^N + \frac{\partial f}{\partial u_w^P} du^P.$$

From the variation of discretized equilibrium, Equation 2.8.2–2, the term dP^N gives rise to the mass matrix (for the d'Alembert forces) and the "load stiffness matrix" in the Jacobian. The load stiffness matrix is discussed in Chapter 3, "Elements," and Chapter 6, "Loading and Constraints," for particular load types. The load stiffness term associated with the weight of the wetting liquid is

$$-\int_V \frac{1}{J} d\big[J(sn+n_t)\rho_w\big]\mathbf{N}^N \cdot \mathbf{g}\, dV,$$

where

$$J \stackrel{\text{def}}{=} \left|\frac{dV}{dV^0}\right|$$

is the ratio of volume in the current configuration to volume in the reference configuration.

The term dI^N is

$$\begin{aligned} dI^N &= d\int_V \boldsymbol{\beta}^N : \boldsymbol{\sigma}\, dV \\ &= \int_V \left[\frac{1}{J}\boldsymbol{\beta}^N : d(J\boldsymbol{\sigma}) + \boldsymbol{\sigma} : d\boldsymbol{\beta}^N\right] dV. \end{aligned}$$

The first term includes $d(J\boldsymbol{\sigma})$, which is the variation of stress caused by variations in nodal positions and pore liquid pressure values. In a continuum sense (that is, before the spatial discretization of the solution variables) this term is defined by the effective stress principle and by the constitutive assumptions used for the material and is discussed in more detail below. Introducing the spatial discretization into the second term provides a contribution to the initial stress matrix.

Since the effective stress, $\overline{\boldsymbol{\sigma}}$, is generally stored as components associated with spatial directions, the rotation of the material during an increment must be included in the formulation. This issue is discussed in detail in "Rate of deformation and strain increment," Section 1.4.3; "Stress rates," Section 1.5.3; "State storage," Section 1.5.4; and "Solid element formulation," Section 3.2.2. For the purpose of the present development we assume that the variation of stress is

$$d(J\boldsymbol{\sigma}) = d^{\nabla}(J(1-n_t)\overline{\boldsymbol{\sigma}}) - d(Jn_t\overline{p}_t\mathbf{I}) + J(d\boldsymbol{\Omega}\cdot\boldsymbol{\sigma} + \boldsymbol{\sigma}\cdot d\boldsymbol{\Omega}^T) - d(J\chi u_w)\,\mathbf{I},$$

where $d^{\nabla}(J\overline{\boldsymbol{\sigma}})$ is the variation in effective stress associated with constitutive response in the material (that is, caused by variations in the strain or other state variables) and $d\boldsymbol{\Omega} \stackrel{\text{def}}{=} \operatorname{asym}(\partial d\mathbf{x}/\partial\mathbf{x})$ is the spin of the material. Using this assumption, the Jacobian contribution from stress in the porous medium is

$$\begin{aligned} dI^N = \int_V \big[\frac{1}{J}\boldsymbol{\beta}^N : \{d^{\nabla}(J(1-n_t)\overline{\boldsymbol{\sigma}}) - d(Jn_t\overline{p}_t\mathbf{I})\} + \boldsymbol{\sigma} : (d\boldsymbol{\beta}^N + 2\boldsymbol{\beta}^N\cdot d\boldsymbol{\Omega}) \\ - \boldsymbol{\beta}^N : \mathbf{I}\,(\chi + u_w\frac{d\chi}{du_w})du_w - \boldsymbol{\beta}^N : \mathbf{I}\,\chi u_w\,\mathbf{I} : d\boldsymbol{\varepsilon}\big]\, dV, \end{aligned} \tag{2.8.2–3}$$

where $d\boldsymbol{\varepsilon} \stackrel{\text{def}}{=} \text{sym}(\partial d\mathbf{x}/\partial \mathbf{x})$ is the strain rate (the "rate of deformation") so that

$$\frac{dJ}{J} = \text{trace}(d\boldsymbol{\varepsilon}) = \mathbf{I} : d\boldsymbol{\varepsilon}.$$

References

- "Coupled pore fluid diffusion and stress analysis," Section 6.8.1 of the Abaqus Analysis User's Guide
- "Geostatic stress state," Section 6.8.2 of the Abaqus Analysis User's Guide

2.8.3 CONSTITUTIVE BEHAVIOR IN A POROUS MEDIUM

Product: Abaqus/Standard

A porous medium in Abaqus/Standard is considered to consist of a mixture of solid matter, voids that contain liquid and gas, and entrapped liquid attached to the solid matter. The mechanical behavior of the porous medium consists of the responses of the liquid and solid matter to local pressure and of the response of the overall material to effective stress. The assumptions made about these responses are discussed in this section.

Liquid response

For the liquid in the system (the free liquid in the voids and the entrapped liquid) we assume that

$$\frac{\rho_w}{\rho_w^0} \approx 1 + \frac{u_w}{K_w} - \varepsilon_w^{th}, \tag{2.8.3–1}$$

where ρ_w is the density of the liquid, ρ_w^0 is its density in the reference configuration, $K_w(\theta)$ is the liquid's bulk modulus, and

$$\varepsilon_w^{th} = 3\alpha_w(\theta - \theta_w^0) - 3\alpha_w|_{\theta^I}(\theta^I - \theta_w^0)$$

is the volumetric expansion of the liquid caused by temperature change. Here $\alpha_w(\theta)$ is the liquid's thermal expansion coefficient, θ is the current temperature, θ^I is the initial temperature at this point in the medium, and θ_w^0 is the reference temperature for thermal expansion. Both u/K_w and ε_w^{th} are assumed to be small.

Grains response

The solid matter in the porous medium is assumed to have the local mechanical response under pressure

$$\frac{\rho_g}{\rho_g^0} \approx 1 + \frac{1}{K_g}\left(su_w + \frac{\overline{p}}{1-n-n_t}\right) - \varepsilon_g^{th}, \tag{2.8.3–2}$$

where $K_g(\theta)$ is the bulk modulus of this solid matter, s is the saturation in the wetting fluid, and

$$\varepsilon_g^{th} = 3\alpha_g(\theta - \theta_g^0) - 3\alpha_g|_{\theta^I}(\theta^I - \theta_g^0)$$

is its volumetric thermal strain. Here $\alpha_g(\theta)$ is the thermal expansion coefficient for the solid matter and θ_g^0 is the reference temperature for this expansion. $\left|1 - \rho_g/\rho_g^0\right|$ is assumed to be small.

It is important to distinguish K_g and α_g as properties of the solid grains material. The porous medium as a whole will exhibit a much softer (and generally nonrecoverable) bulk behavior than is indicated by K_g and will also show a different thermal expansion. These effects are partially structural, caused by the medium being made up of irregular grains in partial contact. They may also be caused by

the system being only partially saturated, with the voids containing a mixture of relatively compressible gas and relatively incompressible liquid.

Liquid entrapment

Entrapment of liquid is associated with specific materials that absorb liquid and swell into a "gel." A simple model of this behavior is based on the idealization of this gel as a volume of individual spherical particles of equal radius r_a. Tanaka and Fillmore (1979) show that, when a single sphere of such material is fully exposed to liquid, its radius change can be modeled as

$$r_a = r_a^f - \sum_N a_N \exp\left(-\frac{t}{\tau_N}\right),$$

where r_a^f is the fully swollen radius approached as $t \to \infty$ and $\boldsymbol{N}$, a_N and τ_N are material parameters. Tanaka and Fillmore also show the first term in the series dominates, so the model can be simplified to

$$r_a = r_a^f - a_1 \exp\left(-\frac{t}{\tau_1}\right).$$

This provides the rate form

$$\dot{r}_a = \frac{r_a^f - r_a}{\tau_1}.$$

When the gel particles are only partially exposed to liquid (in an unsaturated system), it seems reasonable to assume that the swelling rate will be lessened according to the level of saturation. Further, we assume that the gel will swell only when the saturation of the surrounding medium exceeds the effective saturation of the gel, $1 - [(r_a^f)^3 - (r_a)^3]/[(r_a^f)^3 - (r_a^{\rm dry})^3]$, where $r_a^{\rm dry}$ is the radius of a gel particle that is completely dry. We combine these into a simple, linear effect:

$$\dot{r}_a = \frac{r_a^f - r_a}{\tau_1} \left\langle s - 1 + \left(\frac{(r_a^f)^3 - (r_a)^3}{(r_a^f)^3 - (r_a^{\rm dry})^3} \right) \right\rangle,$$

where $\langle f \rangle = f$ if $f > 0$, $\langle f \rangle = 0$ otherwise.

The packing density and swelling may cause the gel particles to touch. In that case the surface available to absorb and entrap liquid is reduced until, if the gel particles occupy the entire volume except for solid material, liquid entrapment must cease altogether. With k_a gel particles per unit reference volume, the maximum radius that the gel particles can achieve before they must touch (in a face center cubic arrangement) is

$$r_a^t \stackrel{\rm def}{=} \left(\frac{n^0 J}{4\sqrt{2} k_a} \right)^{\frac{1}{3}},$$

and the volume is entirely occupied with gel and solid matter when the effective gel radius is

$$r_a^s = \left(\frac{3}{4\pi} \frac{n^0 J}{k_a} \right)^{\frac{1}{3}}.$$

The gel swelling behavior is, therefore, further modified to be

$$\dot{r}_a = \frac{r_a^f - r_a}{\tau_1} \left\langle s - 1 + \left(\frac{(r_a^f)^3 - (r_a)^3}{(r_a^f)^3 - (r_a^{\mathrm{dry}})^3} \right) \right\rangle \left(1 - \left\langle \frac{r_a - r_a^t}{r_a^s - r_a^t} \right\rangle^2 \right). \tag{2.8.3–3}$$

Thus, in an unstressed medium the entrapped liquid volume is assumed to be

$$dV_t = h_t dV^0,$$

where

$$h_t \overset{\mathrm{def}}{=} \frac{4}{3}\pi \left(r_a^3 - (r_a^{\mathrm{dry}})^3 \right) k_a,$$

where $r_a(J, s)$ is defined by the integration of Equation 2.8.3–3. This entrapped liquid can be compressed by pressure so that, when the porous medium is under stress, we assume

$$dV_t = \left(1 - \frac{u_w}{K_w} + \varepsilon_w^{th} \right) h_t dV^0,$$

and thus

$$n_t = \frac{dV_t}{dV} = \frac{h_t}{J} \left(1 - \frac{u_w}{K_w} + \varepsilon_w^{th} \right). \tag{2.8.3–4}$$

Combining this with Equation 2.8.3–1 and neglecting small terms compared to unity then provides

$$J \frac{\rho_w}{\rho_w^0} n_t \approx h_t. \tag{2.8.3–5}$$

We assume that, in the initial state, the effective saturation of the gel is the same as the saturation of the surrounding medium:

$$r_a^0 = \left[(r_a^f)^3 - \left((r_a^f)^3 - (r_a^{\mathrm{dry}})^3 \right) (1 - s^0) \right]^{\frac{1}{3}}.$$

The constitutive behavior of the gel containing entrapped fluid is given by the elastic bulk relationship

$$\overline{p}_t = -K_w(\overline{\varepsilon}_{vol} - \varepsilon_w^{th}),$$

where $\overline{p}_t$ is the average pressure stress in the gel fluid and $\overline{\varepsilon}_{vol}$ is its volumetric effective strain.

Effective strain

From Equation 2.8.3–2 we see that the volumetric strain $-u_w/K_g + \varepsilon_g^{th}$ represents that part of the total volumetric strain caused by pore pressure acting on the solid matter in the porous medium and by thermal expansion of that solid matter. In addition, entrapment of liquid in the medium may cause an additional volume change ratio:

$$1 + \frac{dV_t - dV_t^0}{dV^0} = 1 + Jn_t - n_t^0.$$

Finally, $\boldsymbol{\varepsilon}^{ms}(s)$ is a saturation driven moisture swelling strain that represents the volumetric swelling of the solid skeleton in partially saturated flow conditions. This moisture swelling can be isotropic or anisotropic. The remaining part of the strain in the medium,

$$\overline{\boldsymbol{\varepsilon}} \overset{\text{def}}{=} \boldsymbol{\varepsilon} + \left(\frac{1}{3}\left(\frac{su_w}{K_g} - \varepsilon_g^{th}\right) - \frac{1}{3}\ln(1 + Jn_t - n_t^0) \right) \mathbf{I} - \boldsymbol{\varepsilon}^{ms}(s) \tag{2.8.3–6}$$

is the strain that is assumed to modify the effective stress in the medium. That is, we assume

$$\overline{\boldsymbol{\sigma}} = \overline{\boldsymbol{\sigma}}(\text{history of } \overline{\boldsymbol{\varepsilon}}, \theta, \text{ state variables, etc}).$$

Specific constitutive models of this type are discussed in Chapter 4, "Mechanical Constitutive Theories." From this assumption, and using Equation 2.8.3–5, we can write the Jaumann rate of change of the effective stress in terms of the rate of change of the kinematic and pore liquid pressure variables as

$$\begin{aligned} d^{\nabla}(J(1-n_t)\overline{\boldsymbol{\sigma}}) = J(1-n_t)\overline{\mathbf{D}} : \Bigg(d\boldsymbol{\varepsilon} + \left(\frac{s}{3K_g} + \frac{u_w}{3K_g}\frac{ds}{du_w} + \frac{h_t}{3K_w(1 + Jn_t - n_t^0)} \right) \mathbf{I}\, du_w, \\ - \frac{d\boldsymbol{\varepsilon}^{ms}}{ds}\frac{ds}{du_w} du_w \Bigg), \end{aligned} \tag{2.8.3–7}$$

where $\overline{\mathbf{D}}$ is defined for each particular model in Chapter 4, "Mechanical Constitutive Theories."

Also, for the effective pressure stress of the fluid entrapped in the gel,

$$\begin{aligned} d(Jn_t\overline{p}_t\mathbf{I}) = Jn_tK_w\mathbf{II} : \Bigg(d\boldsymbol{\varepsilon} + \left(\frac{s}{3K_g} + \frac{u_w}{3K_g}\frac{ds}{du_w} + \frac{h_t}{3K_w(1 + Jn_t - n_t^0)} \right) \mathbf{I}\, du_w \\ - \frac{d\boldsymbol{\varepsilon}^{ms}}{ds}\frac{ds}{du_w} du_w \Bigg). \end{aligned} \tag{2.8.3–8}$$

Then, from Equation 2.8.2–3,

$$
\begin{aligned}
dI^N = \int_V \Big[& \boldsymbol{\beta}^N : \{(1-n_t)\overline{\mathbf{D}} + n_t K_w \mathbf{II}\} : d\boldsymbol{\varepsilon} \\
& - \boldsymbol{\beta}^N : \Bigg((\chi + u_w \frac{d\chi}{du_w})\mathbf{I} \\
& \quad - \left(\frac{s}{3K_g} + \frac{u_w}{3K_g}\frac{ds}{du_w} + \frac{h_t}{3K_w(1+Jn_t - n_t^0)} \right) \{(1-n_t)\overline{\mathbf{D}} + n_t K_w \mathbf{II}\} : \mathbf{I} \\
& \quad + \frac{ds}{du_w}\{(1-n_t)\overline{\mathbf{D}} + n_t K_w \mathbf{II}\} : \frac{d\boldsymbol{\varepsilon}^{ms}}{ds} \Bigg)\, du_w \\
& + \boldsymbol{\sigma} : (d\boldsymbol{\beta}^N + 2\boldsymbol{\beta}^N \cdot d\boldsymbol{\Omega}) - \boldsymbol{\beta}^N : \mathbf{I}\chi u_w \mathbf{I} : d\boldsymbol{\varepsilon} \Big] dV.
\end{aligned}
\tag{2.8.3–9}
$$

Reference

- "Pore fluid flow properties," Section 26.6 of the Abaqus Analysis User's Guide

2.8.4 CONTINUITY STATEMENT FOR THE WETTING LIQUID PHASE IN A POROUS MEDIUM

Product: Abaqus/Standard

Abaqus/Standard provides capabilities for particular cases of fluid flow through a porous medium. These cases are associated with having a relatively incompressible wetting liquid present in the medium. The medium may be wholly or partially saturated, with this liquid. When the medium is only partially saturated the remainder of the voids is filled with another fluid. An example is a geotechnical problem, with soil containing water and air: continuity is written for the water phase.

The wetting liquid can attach to and, thus, be trapped by certain solid particles in the medium: this volume of trapped liquid attached to solid particles forms a "gel."

A porous medium is modeled approximately in Abaqus by attaching the finite element mesh to the solid phase. Liquid can flow through this mesh. A continuity equation is, therefore, required for the liquid, equating the rate of increase in liquid mass stored at a point to the rate of mass of liquid flowing into the point within the time increment. This continuity statement is defined in this section. It is written in a variational form as a basis for finite element approximation. The liquid flow is described by introducing Darcy's law or, alternatively, Forchheimer's law. The continuity equation is satisfied approximately in the finite element model by using excess wetting liquid pressure as the nodal variable (degree of freedom 8), interpolated over the elements. The equation is integrated in time by using the backward Euler approximation. The total derivative of this integrated variational statement of continuity with respect to the nodal variables is required for the Newton iterations used to solve the nonlinear, coupled, equilibrium and continuity equations. This expression is also derived in this section.

Consider a volume containing a fixed amount of solid matter. In the current configuration this volume occupies space V with surface S. In the reference configuration it occupied space V^0. Wetting liquid can flow through this volume: at any time the volume of such "free" liquid (liquid that can flow if driven by pressure) is written V_w. Wetting liquid can also become trapped in the volume, by absorption into the gel. The volume of such trapped liquid is written V_t.

The total mass of wetting liquid in the control volume is

$$\int_V \rho_w \left[dV_w + dV_t \right] = \int_V \rho_w (n_w + n_t)\, dV,$$

where ρ_w is the mass density of the liquid.

The time rate of change of this mass of wetting liquid is

$$\frac{d}{dt}\left(\int_V \rho_w (n_w + n_t)\, dV \right) = \int_V \frac{1}{J} \frac{d}{dt} \big(J \rho_w (n_w + n_t) \big)\, dV.$$

The mass of wetting liquid crossing the surface and entering the volume per unit time is

$$- \int_S \rho_w n_w \mathbf{n} \cdot \mathbf{v}_w \, dS,$$

where $\mathbf{v}_w$ is the average velocity of the wetting liquid relative to the solid phase (the seepage velocity) and $\mathbf{n}$ is the outward normal to S.

Equating the addition of liquid mass across the surface S to the rate of change of liquid mass within the volume V gives the wetting liquid mass continuity equation

$$\int_V \frac{1}{J}\frac{d}{dt}\big(J\rho_w(n_w+n_t)\big)\,dV = -\int_S \rho_w n_w \mathbf{n}\cdot\mathbf{v}_w\,dS.$$

Using the divergence theorem and because the volume is arbitrary, this provides the pointwise equation

$$\frac{1}{J}\frac{d}{dt}\big(J\rho_w(n_w+n_t)\big) + \frac{\partial}{\partial\mathbf{x}}\cdot(\rho_w n_w\mathbf{v}_w) = 0.$$

The equivalent weak form is

$$\int_V \delta u_w \frac{1}{J}\frac{d}{dt}\big(J\rho_w(n_w+n_t)\big)\,dV + \int_V \delta u_w \frac{\partial}{\partial\mathbf{x}}\cdot(\rho_w n_w\mathbf{v}_w)\,dV = 0,$$

where δu_w is an arbitrary, continuous, variational field. This statement can also be written on the reference volume:

$$\int_{V^0} \delta u_w \frac{d}{dt}\big(J\rho_w(n_w+n_t)\big)\,dV^0 + \int_{V^0} \delta u_w\, J\frac{\partial}{\partial\mathbf{x}}\cdot(\rho_w n_w\mathbf{v}_w)\,dV^0 = 0.$$

In Abaqus/Standard this continuity statement is integrated approximately in time by the backward Euler formula, giving

$$\int_{V^0} \delta u_w \Big[\big(J\rho_w(n_w+n_t)\big)_{t+\Delta t} - \big(J\rho_w(n_w+n_t)\big)_t\Big]\,dV^0 + \Delta t\int_{V^0}\delta u_w\left[J\frac{\partial}{\partial\mathbf{x}}\cdot(\rho_w n_w\mathbf{v}_w)\right]_{t+\Delta t} dV^0 = 0,$$

which, over the current volume, is

$$\int_V \delta u_w \Big[\big(\rho_w(n_w+n_t)\big)_{t+\Delta t} - \frac{1}{J_{t+\Delta t}}\big(J\rho_w(n_w+n_t)\big)_t\Big]\,dV + \Delta t\int_V \delta u_w\left[\frac{\partial}{\partial\mathbf{x}}\cdot(\rho_w n_w\mathbf{v}_w)\right]_{t+\Delta t} dV = 0.$$

We now drop the subscript $t+\Delta t$ by adopting the convention that any quantity not explicitly associated with a point in time is taken at $t+\Delta t$.

The divergence theorem allows the equation to be rewritten as

$$\int_V \left[\delta u_w \left(\frac{\rho_w}{\rho_w^0}(n_w + n_t) - \frac{1}{J}\left(\frac{\rho_w}{\rho_w^0}(J(n_w + n_t)\right)_t \right) - \Delta t \frac{\rho_w}{\rho_w^0} n_w \frac{\partial\, \delta u_w}{\partial \mathbf{x}} \cdot \mathbf{v}_w \right] dV + \Delta t \int_S \delta u_w \frac{\rho_w}{\rho_w^0} n_w \mathbf{n} \cdot \mathbf{v}_w \, dS = 0,$$

where—for convenience—we have normalized the equation by the density of the liquid in the reference configuration, ρ_w^0.

Since $n_w = sn$, this is the same as

$$\int_V \left[\delta u_w \left(\frac{\rho_w}{\rho_w^0}(sn + n_t) - \frac{1}{J}\left(\frac{\rho_w}{\rho_w^0} J(sn + n_t)\right)_t \right) - \Delta t \frac{\rho_w}{\rho_w^0} sn \frac{\partial\, \delta u_w}{\partial \mathbf{x}} \cdot \mathbf{v}_w \right] dV + \Delta t \int_S \delta u_w \frac{\rho_w}{\rho_w^0} sn \mathbf{n} \cdot \mathbf{v}_w \, dS = 0. \tag{2.8.4–1}$$

Constitutive behavior

The constitutive behavior for pore fluid flow is governed either by Darcy's law or by Forchheimer's law. Darcy's law is generally applicable to low fluid flow velocities, whereas Forchheimer's law is commonly used for situations involving higher flow velocities. Darcy's law can be thought of as a linearized version of Forchheimer's law. Darcy's law states that, under uniform conditions, the volumetric flow rate of the wetting liquid through a unit area of the medium, $sn\mathbf{v}_w$, is proportional to the negative of the gradient of the piezometric head (Bear, 1972):

$$sn\mathbf{v}_w = -\widehat{\mathbf{k}} \cdot \frac{\partial \phi}{\partial \mathbf{x}},$$

where $\widehat{\mathbf{k}}$ is the permeability of the medium and ϕ is the piezometric head, defined as

$$\phi \stackrel{\text{def}}{=} z + \frac{u_w}{g\rho_w},$$

where z is the elevation above some datum and g is the magnitude of the gravitational acceleration, which acts in the direction opposite to z. On the other hand, Forchheimer's law states that the negative of the gradient of the piezometric head is related to a quadratic function of the volumetric flow rate of the wetting liquid through a unit area of the medium (Desai, 1975):

$$sn\mathbf{v}_w(1 + \beta\sqrt{\mathbf{v}_w \cdot \mathbf{v}_w}) = -\widehat{\mathbf{k}} \cdot \frac{\partial \phi}{\partial \mathbf{x}},$$

where $\beta(\mathbf{x}, e)$ is a "velocity coefficient" (Tariq, 1987). This nonlinear permeability can be defined to be dependent on the void ratio of the material. We see that, as the fluid velocity tends to zero, Forchheimer's law approaches Darcy's law. Also, if $\beta = 0$, the two flow laws are identical.

$\widehat{\mathbf{k}}$ can be anisotropic and is a function of the saturation and void ratio of the material. $\widehat{\mathbf{k}}$ has units of velocity (length/time). [Some authors refer to $\widehat{\mathbf{k}}$ as the hydraulic conductivity (Bear, 1972) and define the permeability as

$$\widehat{\mathbf{K}} = \frac{\nu}{g} \frac{1}{(1 + \beta\sqrt{\mathbf{v}_w \cdot \mathbf{v}_w})} \widehat{\mathbf{k}},$$

where ν is the kinematic viscosity of the fluid (the ratio of the fluid's dynamic viscosity to its density).]

We assume that $\boldsymbol{g}$ is constant in magnitude and direction, so

$$\frac{\partial \phi}{\partial \mathbf{x}} = \frac{1}{g\rho_w} \left(\frac{\partial u_w}{\partial \mathbf{x}} - \rho_w \mathbf{g} \right),$$

where $\mathbf{g} \stackrel{\text{def}}{=} -g\partial z/\partial \mathbf{x}$ is the gravitational acceleration (we assume that ρ_w varies slowly with position).

The permeability of a particular fluid in a multiphase flow system depends on the saturation of the phase being considered and on the porosity of the medium. We assume these dependencies are separable, so

$$\widehat{\mathbf{k}} = k_s \mathbf{k},$$

where $k_s(s)$ provides the dependency on saturation, with $k_s(1) = 1.0$ and $\mathbf{k}(\mathbf{x}, e)$ is the permeability of the fully saturated medium.

The function k_s can be defined by the user. Nguyen and Durso (1983) observe that, in steady flow through a partially saturated medium, the permeability varies with s^3. We, therefore, take $k_s = s^3$ by default.

Introducing the flow constitutive law allows the mass continuity equation (Equation 2.8.4–1) to be written

$$\begin{aligned} \int_V \Bigg[\delta u_w \left(\frac{\rho_w}{\rho_w^0}(sn + n_t) - \frac{1}{J} \left(\frac{\rho_w}{\rho_w^0} J(sn + n_t) \right)_t \right) \\ + \Delta t \frac{k_s}{\rho_w^0 g(1 + \beta\sqrt{\mathbf{v}_w \cdot \mathbf{v}_w})} \frac{\partial\, \delta u_w}{\partial \mathbf{x}} \cdot \mathbf{k} \cdot \left(\frac{\partial u_w}{\partial \mathbf{x}} - \rho_w \mathbf{g} \right) \Bigg] dV \\ + \Delta t \int_S \delta u_w \frac{\rho_w}{\rho_w^0} sn\mathbf{n} \cdot \mathbf{v}_w \, dS = 0. \end{aligned} \qquad (2.8.4\text{–}2)$$

Volumetric strain in the liquid and grains

The bulk behavior of the grains was discussed in "Constitutive behavior in a porous medium," Section 2.8.3. From Equation 2.8.3–2,

$$\frac{\rho_g}{\rho_g^0} = \frac{1}{J_g} \approx 1 + \frac{1}{K_g} \left(u_w + \frac{\overline{p}}{1 - n - n_t} \right) - \varepsilon_g^{th}.$$

Combining this with Equation 2.8.1–4 and neglecting all but first-order terms in small quantities, we obtain

$$n \approx 1 - n_t + \frac{\overline{p}}{K_g} + \frac{1}{J}(1 - n^0 - n_t^0)\left(\frac{u_w}{K_g} - \varepsilon_g^{th} - 1\right).$$

Using Equation 2.8.3–1 and again neglecting second-order terms in small quantities, we obtain

$$\frac{\rho_w}{\rho_w^0} n \approx 1 - n_t - \frac{1}{J}(1 - n^0 - n_t^0) + \frac{\overline{p}}{K_g} + u_w\left(\frac{1 - n_t}{K_w} + \frac{(1 - n^0 - n_t^0)}{J}\left(\frac{1}{K_g} - \frac{1}{K_w}\right)\right) - (1 - n_t)\varepsilon_w^{th} + \frac{1}{J}(1 - n^0 - n_t^0)(\varepsilon_w^{th} - \varepsilon_g^{th}).$$

Combining this result with Equation 2.8.3–4, and again approximating to first-order in small quantities,

$$\frac{\rho_w}{\rho_w^0} n \approx 1 - \frac{1}{J}(1 - n^0 - n_t^0 + h_t) + \frac{\overline{p}}{K_g} + u_w\left(\frac{1}{K_w} + \frac{(1 - n^0 - n_t^0)}{J}\left(\frac{1}{K_g} - \frac{1}{K_w}\right)\right) - \varepsilon_w^{th} + \frac{1}{J}(1 - n^0 - n_t^0)(\varepsilon_w^{th} - \varepsilon_g^{th}). \quad (2.8.4–3)$$

Saturation

Because u_w measures pressure in the wetting liquid and we neglect the pressure in the other fluid phase in the medium (see "Effective stress principle for porous media," Section 2.8.1), the medium is fully saturated for $u_w > 0$. Negative values of u_w represent capillary effects in the medium. For $u_w < 0$ it is known (see, for example, Nguyen and Durso, 1983) that, at a given value of capillary pressure, $-u_w$, the saturation lies within certain limits. Typical forms of these limits are shown in Figure 2.8.4–1. We write these limits as $s^a \leq s \leq s^e$, where $s^a(u_w)$ is the limit at which absorption will occur (so that $\dot{s} > 0$), and $s^e(u_w)$ is the limit at which exsorption will occur, and thus $\dot{s} < 0$. We assume that these relationships are uniquely invertible and can, thus, also be written as $u_w^a(s)$ during absorption and $u_w^e(s)$ during exsorption. We also assume that some wetting liquid will always be present in the medium: $s > 0$.

Bear (1972) suggests that the transition between absorption and exsorption and vice versa takes place along "scanning" curves. We approximate these with a straight line, as shown in Figure 2.8.4–1.

Saturation is treated as a state variable that may have to change if the wetting liquid pressure is outside the range for which its value is admissible according to that actual data corresponding to Figure 2.8.4–1. The evolution of saturation as a state variable is defined as follows. Assume that the saturation at time t, $s\big|_t$, is known. It must satisfy the constraints

$$s\big|_t = 1.0 \quad \text{if} \quad u_w > 0.0; \qquad s^a(u_w\big|_t) \leq s\big|_t \leq s^e(u_w\big|_t) \quad \text{otherwise.}$$

We solve the continuity equation for $u_w\big|_{t+\Delta t}$, initially assuming $s\big|_{t+\Delta t} = s\big|_t$. We then obtain $s\big|_{t+\Delta t}$ by the following rules:

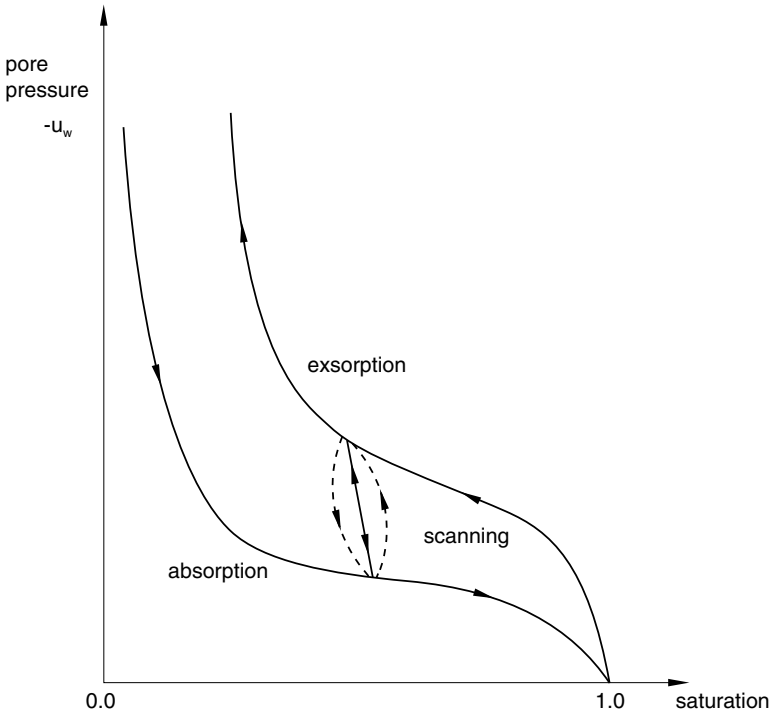


Figure 2.8.4–1 Typical liquid absorption and exsorption behavior.

$$\text{if} \quad u_w\big|_{t+\Delta t} > 0.0 \quad \text{then} \quad s\big|_{t+\Delta t} = 1.0 \quad \text{and} \quad \dot{s} = 0.0;$$

$$\text{else if} \quad u_w\big|_{t+\Delta t} > u_w^a(s_t) \quad \text{then} \quad s\big|_{t+\Delta t} = s^a(u_w\big|_{t+\Delta t}) \quad \text{and} \quad \frac{ds}{du_w} = \left.\frac{ds^a}{du_w}\right|_{u_w\big|_{t+\Delta t}};$$

$$\text{else if} \quad u_w\big|_{t+\Delta t} < u_w^e(s_t) \quad \text{then} \quad s\big|_{t+\Delta t} = s^e(u_w\big|_{t+\Delta t}) \quad \text{and} \quad \frac{ds}{du_w} = \left.\frac{ds^e}{du_w}\right|_{u_w\big|_{t+\Delta t}};$$

$$\text{otherwise} \quad s\big|_{t+\Delta t} = s\big|_t + \left.\frac{ds}{du_w}\right|_s \Delta u_w \quad \text{and} \quad \frac{ds}{du_w} = \left.\frac{ds}{du_w}\right|_s,$$

where $(ds/du_w)\big|_s$ is the slope of the scanning line. These choices are shown in Figure 2.8.4–2.

Jacobian contribution

The Jacobian contribution from the continuity equation is obtained from the variation of Equation 2.8.4–2 with respect to $\mathbf{x}$ and u_w at time $t + \Delta t$.

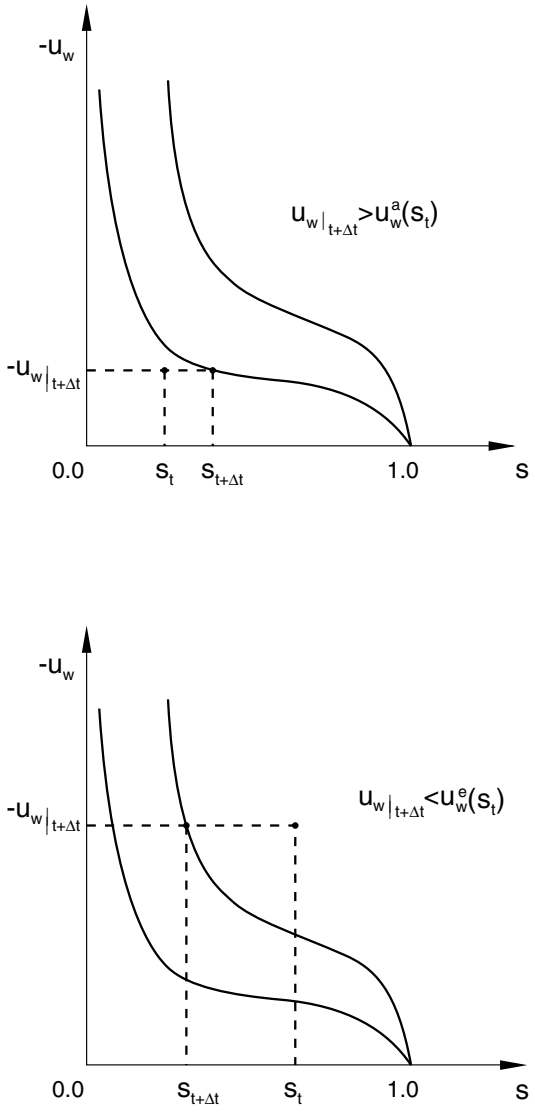


Figure 2.8.4–2 Evolution of ***s*** in unsaturated cases.

Consider first the surface integral. The surface divides into that part across which the liquid mass flow rate, $\rho_w s n \mathbf{n} \cdot \mathbf{v}_w$, is prescribed and that part where the wetting liquid pressure, u_w, is prescribed. Thus, the only contribution of this term to the Jacobian is the variation in the integral caused by change in surface area in that part where the mass flow is prescribed. We neglect this contribution.

The remaining part of the variation of Equation 2.8.4–2 is

$$\int_V \frac{1}{J}\left[\delta u_w\, d\left(J\frac{\rho_w}{\rho_w^0}(sn+n_t)\right)+\frac{\Delta t}{\rho_w^0 g}d\left\{\frac{Jk_s}{(1+\beta\sqrt{\mathbf{v}_w\cdot\mathbf{v}_w})}\frac{\partial\,\delta u_w}{\partial\mathbf{x}}\cdot\mathbf{k}\cdot\left(\frac{\partial u_w}{\partial\mathbf{x}}-\rho_w\mathbf{g}\right)\right\}\right]dV.$$

Using Equation 2.8.4–3 we have

$$J\frac{\rho_w}{\rho_w^0}sn\approx s\Big[J\big(1+\frac{\overline{p}}{K_g}+\frac{u_w}{K_w}-\varepsilon_w^{th}\big)-1+n^0+n_t^0-h_t+u_w(1-n^0-n_t^0)\Big(\frac{1}{K_g}-\frac{1}{K_w}\Big)$$
$$+(1-n^0-n_t^0)(\varepsilon_w^{th}-\varepsilon_g^{th})\Big],$$

and, thus, neglecting small terms compared to unity,

$$d\big(J\frac{\rho_w}{\rho_w^0}sn\big)=s\left[-\frac{J}{3K_g}\mathbf{I}:\overline{\mathbf{D}}+J\mathbf{I}\right]:d\boldsymbol{\varepsilon}$$
$$+\left[\frac{ds}{du_w}(J-1+n^0+n_t^0-h_t)-\frac{sJ}{9K_g}\mathbf{I}:\overline{\mathbf{D}}:\mathbf{I}\Big(\frac{1}{K_g}+\frac{h_t}{K_w(1+h_t-n_t^0)}\Big)\right.$$
$$\left.+\frac{s}{K_w}(J-1+n^0+n_t^0)+\frac{s}{K_g}(1-n^0-n_t^0)\right]du_w.$$

Equation 2.8.3–5 shows that $J(\rho_w/\rho_w^0)n_t\approx h_t$, which is defined by the evolution equation given in "Constitutive behavior in a porous medium," Section 2.8.3, and so makes no contribution to the Jacobian.

Finally, the Jacobian contribution from the permeability term is rather complex in the general case of the nonlinear Forchheimer flow law. Although we include it in the software, here we only write the linearized flow version reflecting Darcy's law ($\beta=0$):

$$d\left[Jk_s\frac{\partial\,\delta u_w}{\partial\mathbf{x}}\cdot\mathbf{k}\cdot\left(\frac{\partial u_w}{\partial\mathbf{x}}-\rho_w\mathbf{g}\right)\right]=$$
$$Jk_s\frac{\partial\,\delta u_w}{\partial\mathbf{x}}\cdot\mathbf{k}\cdot\left(\frac{\partial u_w}{\partial\mathbf{x}}-\rho_w\mathbf{g}\right)\mathbf{I}:d\boldsymbol{\varepsilon}+J\frac{dk_s}{ds}\frac{ds}{du_w}\frac{\partial\,\delta u_w}{\partial\mathbf{x}}\cdot\mathbf{k}\cdot\left(\frac{\partial u_w}{\partial\mathbf{x}}-\rho_w\mathbf{g}\right)du_w$$
$$-Jk_s\frac{\partial\,\delta u_w}{\partial\mathbf{x}}\cdot\frac{\partial\,d\mathbf{x}}{\partial\mathbf{x}}\cdot\mathbf{k}\cdot\left(\frac{\partial u_w}{\partial\mathbf{x}}-\rho_w\mathbf{g}\right)-Jk_s\frac{\partial\,\delta u_w}{\partial\mathbf{x}}\cdot\mathbf{k}\cdot\left(\frac{\partial u_w}{\partial\mathbf{x}}\cdot\frac{\partial\,d\mathbf{x}}{\partial\mathbf{x}}\right)^T$$
$$+Jk_s\frac{\partial\,\delta u_w}{\partial\mathbf{x}}\cdot\mathbf{k}\cdot\frac{\partial\,du_w}{\partial\mathbf{x}}$$
$$+Jk_s\frac{\partial\,\delta u_w}{\partial\mathbf{x}}\cdot\frac{d\mathbf{k}}{de}\cdot\left(\frac{\partial u_w}{\partial\mathbf{x}}-\rho_w\mathbf{g}\right)\frac{1}{(1-n)^2}\left\{\frac{1}{K_g}\mathbf{I}:\overline{\mathbf{D}}:\left[-\frac{1}{3}d\boldsymbol{\varepsilon}\right.\right.$$
$$\left.+(\frac{1}{J}(1-n^0-n_t^0)+n_t)\mathbf{I}:d\boldsymbol{\varepsilon}-\left(\frac{1}{9K_g}+\frac{h_t}{9K_w(1+h_t-n_t^0)}\right)\mathbf{I}\,du_w\right]$$
$$\left.+\frac{(1-n^0-n_t^0)}{JK_g}\,du_w+\frac{n_t}{K_w}\,du_w\right\}.$$

Using these results provides the Jacobian of the continuity equation as

$$
\begin{aligned}
\int_V \Bigg[\delta u_w \Bigg\{ & s\left[-\frac{1}{3K_g}\mathbf{I} : \overline{\mathbf{D}} + \mathbf{I} \right] : d\boldsymbol{\varepsilon} \\
& + \left[\frac{ds}{du_w}\frac{1}{J}(J - 1 + n^0 + n_t^0 - h_t) - \frac{s}{9K_g}\mathbf{I} : \overline{\mathbf{D}} : \mathbf{I}\left(\frac{1}{K_g} + \frac{h_t}{K_w(1 + h_t - n_t^0)} \right) \right. \\
& \left. + \frac{s}{JK_w}(J - 1 + n^0 + n_t^0) + s\frac{1 - n^0 - n_t^0}{JK_g} \right] du_w \Bigg\} \\
& + \Delta t \frac{k_s}{\rho_w^0 g} \Bigg\{ \frac{\partial\, \delta u_w}{\partial \mathbf{x}} \cdot \left[\mathbf{k} \cdot \left(\frac{\partial u_w}{\partial \mathbf{x}} - \rho_w \mathbf{g} \right) \mathbf{I} \right. \\
& \left. + \frac{d\mathbf{k}}{de} \cdot \left(\frac{\partial u_w}{\partial \mathbf{x}} - \rho_w \mathbf{g} \right) \frac{1}{(1 - n)^2} \left(-\frac{1}{3K_g}\mathbf{I} : \overline{\mathbf{D}} + \left(\frac{1}{J}(1 - n^0 - n_t^0) + n_t \right) \mathbf{I} \right) \right] : d\boldsymbol{\varepsilon} \\
& + \frac{\partial\, \delta u_w}{\partial \mathbf{x}} \cdot \mathbf{k} \cdot \frac{\partial\, du_w}{\partial \mathbf{x}} \\
& + \left[\frac{1}{k_s}\frac{dk_s}{ds}\frac{ds}{du_w}\frac{\partial\, \delta u_w}{\partial \mathbf{x}} \cdot \mathbf{k} \cdot \left(\frac{\partial u_w}{\partial \mathbf{x}} - \rho_w \mathbf{g} \right) \right. \\
& + \frac{1}{(1 - n)^2}\frac{\partial\, \delta u_w}{\partial \mathbf{x}} \cdot \frac{d\mathbf{k}}{de} \cdot \left(\frac{\partial u_w}{\partial \mathbf{x}} - \rho_w \mathbf{g} \right) \left(\frac{\mathbf{I} : \overline{\mathbf{D}} : \mathbf{I}}{9K_g} \left(-\frac{1}{K_g} - \frac{h_t}{K_w(1 + h_t - n_t^0)} \right) \right. \\
& \left. \left. + \frac{n_t}{K_w} + \frac{1 - n^0 - n_t^0}{JK_g} \right) \right] du_w \\
& - \left(\frac{\partial\, \delta u_w}{\partial \mathbf{x}} \cdot \frac{\partial\, d\mathbf{x}}{\partial \mathbf{x}} \cdot \mathbf{k} \cdot \left(\frac{\partial u_w}{\partial \mathbf{x}} - \rho_w \mathbf{g} \right) + \frac{\partial\, \delta u_w}{\partial \mathbf{x}} \cdot \mathbf{k} \cdot \left(\frac{\partial u_w}{\partial \mathbf{x}} \cdot \frac{\partial\, d\mathbf{x}}{\partial \mathbf{x}} \right)^T \right) \Bigg\} \Bigg] dV.
\end{aligned}
$$

References

- “Coupled pore fluid diffusion and stress analysis,” Section 6.8.1 of the Abaqus Analysis User’s Guide
- “Pore fluid flow properties,” Section 26.6.1 of the Abaqus Analysis User’s Guide

2.8.5 SOLUTION STRATEGY FOR COUPLED DIFFUSION/DEFORMATION

Product: Abaqus/Standard

The governing equations of pore fluid diffusion/deformation are

$$\text{equilibrium: } K^{MN}\bar{c}_\delta^N - L^{MP}\bar{c}_u^P = P^M - I^M, \text{ and}$$

$$\text{pore fluid flow: } (\widehat{B}^{MQ})^T\bar{v}^M + \widehat{H}^{QP}\bar{u}^P = Q^Q.$$

There are two common approaches to solving these coupled equations. One approach is to solve one set of equations first and then use the results obtained to solve the second set of equations. These results in turn are fed back into the first set of equations to see what changes (if any) result in the solution. This process continues until succeeding iterations produce negligible changes in the solutions obtained. This is the so-called staggered approach to the solution of coupled systems of equations. The second approach is to solve the coupled systems directly. This direct approach is used in Abaqus/Standard because of its rapid convergence even in severely nonlinear cases.

We first introduce a time integration operator in the pore fluid flow equation. The operator chosen is the simple one-step method:

$$\bar{\delta}_{t+\triangle t}^N = \bar{\delta}_t^N + \triangle t[(1-\zeta)\bar{v}_t^N + \zeta\bar{v}_{t+\triangle t}],$$

where $0 \le \zeta \le 1$. In fact, to ensure numerical stability, we choose $\zeta = 1$ (backward difference) so that

$$\bar{v}_{t+\triangle t} = \frac{1}{\triangle t}(\bar{\delta}_{t+\triangle t}^N - \bar{\delta}_t).$$

With this operator the pore fluid flow equation at time $(t+\triangle t)$ can be rewritten as

$$(\widehat{B}^{MQ})^T\bar{\delta}_{t+\triangle t}^M + \triangle t\widehat{H}^{QP}\bar{u}_{t+\triangle t}^P = \triangle t Q_{t+\triangle t}^Q + (\widehat{B}^{MQ})^T\bar{\delta}_t^M.$$

Using the Newton linearization, the flow equation becomes

$$-(B^{MQ})^T\bar{c}_\delta^M - \triangle t H^{QP}\bar{c}_u^P = \triangle t[-Q_{t+\triangle t}^Q + (\widehat{B}^{MQ})^T\bar{v}_{t+\triangle t}^M + \widehat{H}^{QP}\bar{u}_{t+\triangle t}^P].$$

Then the coupled system of equations to be solved is

$$K^{MN}\bar{c}_\delta^N - L^{MP}\bar{c}_u^P = P^M - I^M$$

and

$$-(B^{MQ})^T\bar{c}_\delta^M - \triangle t H^{QP}\bar{c}_u^P = R^Q,$$

where

$$R^Q = \triangle t[-Q^Q_{t+\triangle t} + (\widehat{B}^{MQ})^T \bar{v}^M_{t+\triangle t} + \widehat{H}^{QP} \bar{u}^P_{t+\triangle t}].$$

These equations form the basis of the iterative solution of a time step in a coupled flow deformation solution in Abaqus/Standard. They are, in general, nonsymmetric. The lack of symmetry may be due to a number of effects: changes in geometry, dependence of permeability on void ratio, changes in saturation in partially saturated cases, and inclusion of fluid gravity load terms in total pore pressure analyses. The steady-state version of the coupled problem is also nonsymmetric.

Abaqus/Standard uses the nonsymmetric equation solver by default in all steady-state or partially saturated coupled analyses; in other cases it uses the symmetric solver by default. In the latter cases, if the effects of changes in geometry or nonlinear permeability are significant, or if a total pore pressure (versus excess pore pressure) analysis is performed, the user is advised to activate the unsymmetric solver.

Reference

- "Coupled pore fluid diffusion and stress analysis," Section 6.8.1 of the Abaqus Analysis User's Guide

2.9 Coupled fluid-solid analysis

- “Coupled acoustic-structural medium analysis,” Section 2.9.1

2.9.1 COUPLED ACOUSTIC-STRUCTURAL MEDIUM ANALYSIS

Products: Abaqus/Standard Abaqus/Explicit

Abaqus provides a set of elements for modeling a fluid medium undergoing small pressure variations and interface conditions to couple these acoustic elements to a structural model. These elements are provided to model a variety of phenomena involving dynamic interactions between fluid and solid media.

Steady-state harmonic (linear) response analysis can be performed for a coupled acoustic-structural system, such as the study of the noise level in a vehicle. The steady-state procedure is based on direct solution of the coupled complex harmonic equations, as described in "Direct steady-state dynamic analysis," Section 2.6.1; on a modal-based procedure, as described in "Steady-state linear dynamic analysis," Section 2.5.7; or on a subspace-based procedure, as described in "Subspace-based steady-state dynamic analysis," Section 2.6.2. Mode-based linear transient dynamic analysis is also available, as described in "Modal dynamic analysis," Section 2.5.5.

The acoustic fluid elements can also be used with nonlinear response analysis (implicit or explicit direct integration) procedures: whether such results are useful depends on the applicability of the small pressure change assumption in the fluid. Often in coupled fluid-solid problems the fluid forces in this linear regime are high enough that nonlinear response of the structure needs to be considered. For example, a ship subjected to underwater incident wave loads due to an explosion may experience plastic deformation or large motions of internal machinery may occur.

The acoustic medium in Abaqus may have velocity-dependent dissipation, caused by fluid viscosity or by flow within a resistive porous matrix material. In addition, rather general boundary conditions are provided for the acoustic medium, including impedance, or "reactive," boundaries.

The possible conditions at the surface of the acoustic medium are:

1. Prescribed pressure (degree of freedom 8) at the boundary nodes. (Boundary conditions can be used to specify pressure at any node in the model.)
2. Prescribed inward normal derivative of pressure per unit density of the acoustic medium through the use of a concentrated load on degree of freedom 8 of a boundary node. If the applied load has zero magnitude (that is, if no concentrated load or other boundary condition is present), the inward normal derivative of pressure (and normal fluid particle acceleration) is zero, which means that the default boundary condition of the acoustic medium is a rigid, fixed wall (Neumann condition).
3. Acoustic-structural coupling defined either by using surface-based coupling procedures (see "Surface-based acoustic-structural medium interaction," Section 5.2.7) or by placing ASI coupling elements on the interface between the acoustic medium and a structure.
4. An impedance condition, representing an absorbing boundary between the acoustic medium and a rigid wall or a vibrating structure or representing radiation to an infinite exterior.
5. An incident wave loading, representing the inward normal derivative of pressure per unit density of the acoustic medium resulting from the arrival of a specified wave. The formulation of this loading case is discussed in "Loading due to an incident dilatational wave field," Section 6.3.1. It is applicable to problems involving blast loads and to acoustic scattering problems.

The flow resistance and the properties of the absorbing boundaries may be functions of frequency in steady-state response analysis but are assumed to be constant in the direct integration procedure. This section defines the formulation used in these elements.

Acoustic equations

The equilibrium equation for small motions of a compressible, adiabatic fluid with velocity-dependent momentum losses is taken to be

$$\frac{\partial p}{\partial \mathbf{x}} + \gamma\left(\mathbf{x}, \theta_i\right)\, \dot{\mathbf{u}}^f + \rho_f\left(\mathbf{x}, \theta_i\right)\, \ddot{\mathbf{u}}^f = 0, \tag{2.9.1–1}$$

where p is the excess pressure in the fluid (the pressure in excess of any static pressure); $\mathbf{x}$ is the spatial position of the fluid particle; $\dot{\mathbf{u}}^f$ is the fluid particle velocity; $\ddot{\mathbf{u}}^f$ is the fluid particle acceleration; ρ_f is the density of the fluid; γ is the "volumetric drag" (force per unit volume per velocity); and θ_i are i independent field variables such as temperature, humidity of air, or salinity of water on which ρ_f and γ may depend (see "Acoustic medium," Section 26.3.1 of the Abaqus Analysis User's Guide). The d'Alembert term has been written without convection on the assumption that there is no steady flow of the fluid. This is usually considered sufficiently accurate for steady fluid velocities up to Mach 0.1.

The constitutive behavior of the fluid is assumed to be inviscid, linear, and compressible, so

$$p = -\mathrm{K}_f\left(\mathbf{x}, \theta_i\right)\, \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{u}^f, \tag{2.9.1–2}$$

where K_f is the bulk modulus of the fluid.

For an acoustic medium capable of undergoing cavitation, the absolute pressure (sum of the static pressure and the excess dynamic pressure) cannot drop below the specified cavitation limit. When the absolute pressure drops to this limit value, the fluid is assumed to undergo free expansion without a corresponding drop in the dynamic pressure. The pressure would rebuild in the acoustic medium once the free expansion that took place during the cavitation is reversed sufficiently to reduce the volumetric strain to the level at the cavitation limit. The constitutive behavior for an acoustic medium capable of undergoing cavitation can be stated as

$$p = \max\left\{p_v,\ p_c - p_0\right\},$$

where a pseudopressure p_v, a measure of the volumetric strain, is defined as

$$p_v = -\mathrm{K}_f\left(\mathbf{x}, \theta_i\right)\, \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{u}^f,$$

where p_c is the fluid cavitation limit and p_0 is the initial acoustic static pressure. A total wave formulation is used for a nonlinear acoustic medium undergoing cavitation. This formulation is very similar to the scattered wave formulation presented below except that the pseudopressure, defined as the product of the bulk modulus and the compressive volumetric strain, plays the role of the material state variable instead of

the acoustic excess pressure. The acoustic excess pressure is readily available from this pseudopressure subject to the cavitation condition.

Physical boundary conditions in acoustic analysis

Acoustic fields are strongly dependent on the conditions at the boundary of the acoustic medium. The boundary of a region of acoustic medium that obeys Equation 2.9.1–1 and Equation 2.9.1–2 can be divided into subregions S on which the following conditions are imposed:

S_{fp}, where the value of the acoustic pressure p is prescribed.

S_{ft}, where we prescribe the normal derivative of the acoustic medium. This condition also prescribes the motion of the fluid particles and can be used to model acoustic sources, rigid walls (baffles), incident wave fields, and symmetry planes.

S_{fr}, the "reactive" acoustic boundary, where there is a prescribed linear relationship between the fluid acoustic pressure and its normal derivative. Quite a few physical effects can be modeled in this manner: in particular, the effect of thin layers of material, whose own motions are unimportant, placed between acoustic media and rigid baffles. An example is the carpet glued to the floor of a room or car interior that absorbs and reflects acoustic waves. This thin layer of material provides a "reactive surface," or impedance boundary condition, to the acoustic medium. This type of boundary condition is also referred to as an imposed impedance, admittance, or a "Dirichlet to Neumann map."

S_{fi}, the "radiating" acoustic boundary. Often, acoustic media extend sufficiently far from the region of interest that they can be modeled as infinite in extent. In such cases it is convenient to truncate the computational region and apply a boundary condition to simulate waves passing exclusively outward from the computational region.

S_{fs}, where the motion of an acoustic medium is directly coupled to the motion of a solid. On such an acoustic-structural boundary the acoustic and structural media have the same displacement normal to the boundary, but the tangential motions are uncoupled.

S_{frs}, an acoustic-structural boundary, where the displacements are linearly coupled but not necessarily identically equal due to the presence of a compliant or reactive intervening layer. This layer induces an impedance condition between the relative normal velocity between acoustic fluid and solid structure and the acoustic pressure. It is analogous to a spring and dashpot interposed between the fluid and solid particles. As implemented in Abaqus, an impedance boundary condition surface does not model any mass associated with the reactive lining; if such a mass exists, it should be incorporated into the boundary of the structure.

S_{ff}, a boundary between acoustic fluids of possibly differing material properties. On such an interface, displacement continuity requires that the normal forces per unit mass on the fluid particles be equal. This quantity is the natural boundary traction in Abaqus, so this condition is enforced automatically during element assembly. This is also true in one-dimensional analysis (i.e., piping or ducts), where the relevant acoustic properties include the cross-sectional areas of the elements. Consequently, fluid-fluid boundaries do not require special treatment in Abaqus.

Formulation for direct integration transient dynamics

In Abaqus the finite element formulations are slightly different in direct integration transient and steady-state or modal analyses, primarily with regard to the treatment of the volumetric drag loss parameter and spatial variations of the constitutive parameters. To derive a symmetric system of ordinary differential equations for implicit integration, some approximations are made in the transient case that are not needed in steady state. For linear transient dynamic analysis, the modal procedure can be used and is much more efficient.

To derive the partial differential equation used in direct integration transient analysis, we divide Equation 2.9.1–1 by ρ_f, take its gradient with respect to $\mathbf{x}$, neglect the gradient of γ/ρ_f, and combine the result with the time derivatives of Equation 2.9.1–2 to obtain the equation of motion for the fluid in terms of the fluid pressure:

$$\frac{1}{\mathrm{K}_f}\,\ddot{p} + \frac{\gamma}{\rho_f \mathrm{K}_f}\,\dot{p} - \frac{\partial}{\partial \mathbf{x}} \cdot \left(\frac{1}{\rho_f}\frac{\partial p}{\partial \mathbf{x}}\right) = 0. \tag{2.9.1–3}$$

The assumption that the gradient of γ/ρ_f is small is violated where there are discontinuities in the quantity γ/ρ_f (for example, on the boundary between two elements that have a different γ/ρ_f value).

Variational statement

An equivalent weak form for the equation of motion, Equation 2.9.1–3, is obtained by introducing an arbitrary variational field, δp, and integrating over the fluid:

$$\int_{V_f} \delta p \left(\frac{1}{\mathrm{K}_f}\,\ddot{p} + \frac{\gamma}{\rho_f \mathrm{K}_f}\,\dot{p} - \frac{\partial}{\partial \mathbf{x}} \cdot \left(\frac{1}{\rho_f}\frac{\partial p}{\partial \mathbf{x}}\right)\right) dV = 0.$$

Green's theorem allows this to be rewritten as

$$\int_{V_f} \left[\delta p \left(\frac{1}{\mathrm{K}_f}\,\ddot{p} + \frac{\gamma}{\rho_f \mathrm{K}_f}\,\dot{p}\right) + \frac{1}{\rho_f}\frac{\partial \delta p}{\partial \mathbf{x}} \cdot \frac{\partial p}{\partial \mathbf{x}}\right] dV + \int_S \delta p \left(\frac{1}{\rho_f}\mathbf{n}^- \cdot \frac{\partial p}{\partial \mathbf{x}}\right) dS = 0. \tag{2.9.1–4}$$

Assuming that $\boldsymbol{p}$ is prescribed on S_{fp}, the equilibrium equation, Equation 2.9.1–1, is used on the remainder of the boundary to relate the pressure gradient to the motion of the boundary:

$$\mathbf{n}^- \cdot \left(\frac{1}{\rho_f}\frac{\partial p}{\partial \mathbf{x}} + \frac{\gamma}{\rho_f}\dot{\mathbf{u}}^f + \ddot{\mathbf{u}}^f\right) = 0 \quad \text{on} \quad S - S_{\mathrm{fp}}. \tag{2.9.1–5}$$

Using this equation, the term $\mathbf{n}^- \cdot \frac{\partial p}{\partial \mathbf{x}}$ is eliminated from Equation 2.9.1–4 to produce

$$\int_{V_f} \left[\delta p \left(\frac{1}{\mathrm{K}_f}\,\ddot{p} + \frac{\gamma}{\rho_f \mathrm{K}_f}\,\dot{p}\right) + \frac{1}{\rho_f}\frac{\partial \delta p}{\partial \mathbf{x}} \cdot \frac{\partial p}{\partial \mathbf{x}}\right] dV - \int_{S - S_{\mathrm{fp}}} \delta p\,(T(\mathbf{x}))\; dS = 0, \tag{2.9.1–6}$$

where, for convenience, the boundary "traction" term

$$T(\mathbf{x}) = \mathbf{n}^- \cdot (\ddot{\mathbf{u}}^f + \frac{\gamma}{\rho_f}\dot{\mathbf{u}}^f) = -\mathbf{n}^- \cdot (\frac{1}{\rho_f}\frac{\partial p}{\partial \mathbf{x}}) \quad \text{on} \quad S - S_{\text{fp}} \tag{2.9.1–7}$$

has been introduced.

Except for the imposed pressure on S_{fp}, all the other boundary conditions described above can be formulated in terms of $T(\mathbf{x})$. This term has dimensions of acceleration; in the absence of volumetric drag this boundary traction is equal to the inward acceleration of the particles of the acoustic medium:

$$T(\mathbf{x}) = \mathbf{n}^- \cdot \ddot{\mathbf{u}}^f \quad \text{on} \quad S - S_{\text{fp}}. \tag{2.9.1–8}$$

When volumetric drag is present, the boundary traction is the normal derivative of the pressure field, divided by the true mass density: a force per unit mass of fluid. Consequently, when volumetric drag exists in a transient acoustic model, a unit of $T(\mathbf{x})$ yields a lower local volumetric acceleration, due to drag losses.

In direct integration transient dynamics we enforce the acoustic boundary conditions as follows:

On S_{fp}, $\boldsymbol{p}$ is prescribed and $\delta p = 0$.

On S_{ft}, where we prescribe the normal derivative of the acoustic pressure per unit density:

$$T_{\text{ft}}(\mathbf{x}) \equiv T_0.$$

In the absence of volumetric drag in the medium, this enforces a value of fluid particle acceleration, $\mathbf{n}^- \cdot \ddot{\mathbf{u}}^f = T_0 = a_{in}$. An imposed $T_0 = a_{in}$ can be used to model the oscillations of a rigid plate or body exciting a fluid, for example. A special case of this boundary condition is $a_{in} = 0$, which represents a rigid immobile boundary. As mentioned above, if the medium has nonzero volumetric drag, a unit of T_0 imposed at the boundary will result in a relatively lower imposed particle acceleration. Incident wave fields on a boundary of a fluid are modeled with a T_0 that varies in space and time, corresponding to the effect of the arrival of the wave on the boundary. See "Loading due to an incident dilatational wave field," Section 6.3.1.

On S_{fr}, the reactive boundary between the acoustic medium and a rigid baffle, we apply a condition that relates the velocity of the acoustic medium to the pressure and rate of change of pressure:

$$-\mathbf{n}^- \cdot \dot{\mathbf{u}}^f = (\frac{1}{k_1}\dot{p} + \frac{1}{c_1}p) \quad \text{on} \quad S_{\text{fr}}, \tag{2.9.1–9}$$

where $1/k_1$ and $1/c_1$ are user-prescribed parameters at the boundary. This equation is in the form of an admittance relation; the impedance expression is simply the inverse. The layer of material, in admittance form, acts as a spring and dashpot in series distributed between the acoustic medium and the rigid wall. The spring and dashpot parameters are k_1 and c_1, respectively; they are per unit area of the acoustic

boundary. Using this definition for the fluid velocity, the boundary tractions in the variational statement become

$$T_{\rm fr}(\mathbf{x}) \equiv -\left(\frac{\gamma}{\rho_f}\frac{1}{c_1}\,p + \left(\frac{\gamma}{\rho_f}\frac{1}{k_1} + \frac{1}{c_1}\right)\,\dot{p} + \frac{1}{k_1}\,\ddot{p}\right). \tag{2.9.1–10}$$

On $S_{\rm fi}$, the radiating boundary, we apply the radiation boundary condition by specifying the corresponding impedance:

$$T_{\rm fi}(\mathbf{x}) \equiv -\left(\frac{1}{c_1}\,\dot{p} + \frac{1}{a_1}\,p\right),$$

using the admittance parameters of Equation 2.9.1–47 and Equation 2.9.1–48, defined below.

On $S_{\rm fs}$, the acoustic-structural interface, we apply the acoustic-structural interface condition by equating displacement of the fluid and solid, which enforces the condition

$$\mathbf{n}^- \cdot \mathbf{u}^f = \mathbf{n}^- \cdot \mathbf{u}^m,$$

where $\mathbf{u}^m$ is the displacement of the structure. In the presence of volumetric drag it follows that the acoustic boundary traction coupling fluid to solid is

$$T(\mathbf{x}) = \mathbf{n}^- \cdot (\ddot{\mathbf{u}}^m + \frac{\gamma}{\rho_f}\dot{\mathbf{u}}^m).$$

In Abaqus/Standard the formulation of the transient coupled problem would be made nonsymmetric by the presence of the term $\mathbf{n}^- \cdot (\frac{\gamma}{\rho_f}\dot{\mathbf{u}}^m)$. In the great majority of practical applications the acoustic tractions associated with volumetric drag are small compared to those associated with fluid inertia,

$$\ddot{\mathbf{u}}^m \gg \frac{\gamma}{\rho_f}\dot{\mathbf{u}}^m \quad \forall \mathbf{u}^m(t),$$

so this term is ignored in transient analysis:

$$T_{\rm fs}(\mathbf{x}) \equiv \mathbf{n}^- \cdot \ddot{\mathbf{u}}^m.$$

On $S_{\rm frs}$, the mixed impedance boundary and acoustic-structural boundary, we apply a condition that relates the relative outward velocity between the acoustic medium and the structure to the pressure and rate of change of pressure:

$$\mathbf{n}^- \cdot (\dot{\mathbf{u}}^m - \dot{\mathbf{u}}^f) = \frac{1}{k_1}\dot{p} + \frac{1}{c_1}p \quad \text{on} \quad S_{\rm frs}. \tag{2.9.1–11}$$

This relative normal velocity represents a rate of compression (or extension) of the intervening layer. Applying this equation to the definition of $T(\mathbf{x})$, we obtain for the transient case

$$T_{\mathrm{frs}}(\mathbf{x}) = \mathbf{n}^- \cdot (\ddot{\mathbf{u}}^m) - \frac{\gamma}{\rho_f}\frac{1}{c_1}\,p - \left(\frac{\gamma}{\rho_f}\frac{1}{k_1} + \frac{1}{c_1}\right)\dot{p} - \frac{1}{k_1}\,\ddot{p}. \tag{2.9.1–12}$$

This expression for $T(\mathbf{x})$ is the sum of its definitions for S_{fs} and S_{fr}. In the steady-state case the effect of volumetric drag on the structural displacement term in the acoustic traction is included:

$$T_{\mathrm{frs}}(\mathbf{x}) = \mathbf{n}^- \cdot (\ddot{\mathbf{u}}^m + \frac{\gamma}{\rho_f}\dot{\mathbf{u}}^m) - \frac{\gamma}{\rho_f}\frac{1}{c_1}\,p - \left(\frac{\gamma}{\rho_f}\frac{1}{k_1} + \frac{1}{c_1}\right)\dot{p} - \frac{1}{k_1}\,\ddot{p}. \tag{2.9.1–13}$$

These definitions for the boundary term, $T(\mathbf{x})$, are introduced into Equation 2.9.1–6 to give the final variational statement for the acoustic medium (this is the equivalent of the virtual work statement for the structure):

$$\begin{aligned}
&\int_{V_f}\left[\delta p\left(\frac{1}{\mathrm{K}_f}\,\ddot{p} + \frac{\gamma}{\rho_f \mathrm{K}_f}\,\dot{p}\right) + \frac{1}{\rho_f}\frac{\partial \delta p}{\partial \mathbf{x}}\cdot\frac{\partial p}{\partial \mathbf{x}}\right]dV - \int_{S_{\mathrm{ft}}}\delta p\, T_0\, dS \\
&\quad + \int_{S_{\mathrm{fr}}}\delta p\left(\frac{\gamma}{\rho_f c_1}\,p + \left(\frac{\gamma}{\rho_f k_1} + \frac{1}{c_1}\right)\dot{p} + \frac{1}{k_1}\,\ddot{p}\right)dS \\
&\quad + \int_{S_{\mathrm{fi}}}\delta p\left(\frac{1}{c_1}\,\dot{p} + \frac{1}{a_1}\,p\right)dS - \int_{S_{\mathrm{fs}}}\delta p\,\mathbf{n}^-\cdot\ddot{\mathbf{u}}^m\,dS \\
&+ \int_{S_{\mathrm{frs}}}\delta p\left(\frac{\gamma}{\rho_f c_1}\,p + \left(\frac{\gamma}{\rho_f k_1} + \frac{1}{c_1}\right)\dot{p} + \frac{1}{k_1}\,\ddot{p} - \mathbf{n}^-\cdot\ddot{\mathbf{u}}^m\right)dS = 0.
\end{aligned} \tag{2.9.1–14}$$

The structural behavior is defined by the virtual work equation,

$$\begin{aligned}
&\int_V \delta\boldsymbol{\varepsilon} : \boldsymbol{\sigma}\,dV + \int_V \alpha_c \rho\,\delta\mathbf{u}^m\cdot\dot{\mathbf{u}}^m\,dV + \int_V \rho\,\delta\mathbf{u}^m\cdot\ddot{\mathbf{u}}^m\,dV \\
&\quad + \int_{S_{\mathrm{fs}}} p\,\delta\mathbf{u}^m\cdot\mathbf{n}\,dS - \int_{S_{\mathrm{t}}}\delta\mathbf{u}^m\cdot\mathbf{t}\,dS = 0,
\end{aligned} \tag{2.9.1–15}$$

where $\boldsymbol{\sigma}$ is the stress at a point in the structure, $\boldsymbol{p}$ is the pressure acting on the fluid-structural interface, $\mathbf{n}$ is the outward normal to the structure, ρ is the density of the material, α_c is the mass proportional damping factor (part of the Rayleigh damping assumption for the structure), $\ddot{\mathbf{u}}^m$ is the acceleration of a point in the structure, $\mathbf{t}$ is the surface traction applied to the structure, $\delta\mathbf{u}^m$ is a variational displacement field, and $\delta\boldsymbol{\varepsilon}$ is the strain variation that is compatible with $\delta\mathbf{u}^m$. For simplicity in this equation all other

loading terms except the fluid pressure and surface traction $\mathbf{t}$ have been neglected: they are imposed in the usual way.

The discretized finite element equations

Equation 2.9.1–14 and Equation 2.9.1–15 define the variational problem for the coupled fields $\mathbf{u}^m$ and $\boldsymbol{p}$. The problem is discretized by introducing interpolation functions: in the fluid $p = H^P\, p^P$, $P = 1, 2 \ldots$ up to the number of pressure nodes and in the structure $\mathbf{u}^m = \mathbf{N}^N\, u^N$, $N = 1, 2 \ldots$ up to the number of displacement degrees of freedom. In these and the following equations we assume summation over the superscripts that refer to the degrees of freedom of the discretized model. We also use the superscripts P, Q to refer to pressure degrees of freedom in the fluid and N, M to refer to displacement degrees of freedom in the structure. We use a Galerkin method for the structural system; the variational field has the same form as the displacement: $\delta\mathbf{u}^m = \mathbf{N}^N\, \delta u^N$. For the fluid we use $\delta p = H^P\, \delta p^P$ but with the subsequent Petrov-Galerkin substitution

$$\delta p^P = \frac{\mathrm{d}^2}{\mathrm{d}t^2}(\delta\hat{p}^P). \tag{2.9.1–16}$$

The new function $\delta\hat{p}^P$ makes the single variational equation obtained from summing Equation 2.9.1–14 and Equation 2.9.1–15 dimensionally consistent:

$$\begin{aligned}
&-\delta\hat{p}^P \left\{ (M_{\mathrm{f}}^{PQ} + M_{\mathrm{fr}}^{PQ})\ddot{p}^Q + (C_{\mathrm{f}}^{PQ} + C_{\mathrm{fr}}^{PQ})\dot{p}^Q + (K_{\mathrm{f}}^{PQ} + K_{\mathrm{fr}}^{PQ} + K_{\mathrm{fi}}^{PQ})p^Q - S_{\mathrm{fs}}^{PM}\,\ddot{u}^M - P_f^P \right\} \\
&+\delta u^N \left\{ I^N + M^{NM}\ddot{u}^M + C_{(m)}^{NM}\dot{u}^M + \left[S_{\mathrm{fs}}^{QN}\right]^T\, p^Q - P^N \right\} = 0,
\end{aligned} \tag{2.9.1–17}$$

where, for simplicity, we have introduced the following definitions:

$$\begin{aligned}
M_{\mathrm{f}}^{PQ} &= \int_{V_f} \frac{1}{\mathrm{K}_f} H^P\, H^Q\, dV, \\
M_{\mathrm{fr}}^{PQ} &= \int_{S_{\mathrm{fr}} \cup S_{\mathrm{frs}}} \frac{1}{k_1} H^P\, H^Q\, dS, \\
C_{\mathrm{f}}^{PQ} &= \int_{V_f} \frac{\gamma}{\rho_f} \frac{1}{\mathrm{K}_f} H^P\, H^Q\, dV, \\
C_{\mathrm{fr}}^{PQ} &= \int_{S_{\mathrm{fr}} \cup S_{\mathrm{frs}}} \left(\frac{\gamma}{\rho_f} \frac{1}{k_1} + \frac{1}{c_1} \right) H^P\, H^Q\, dS + \int_{S_{\mathrm{fi}}} \frac{1}{c_1} H^P\, H^Q\, dS, \\
K_{\mathrm{f}}^{PQ} &= \int_{V_f} \frac{1}{\rho_f} \frac{\partial H^P}{\partial \mathbf{x}} \cdot \frac{\partial H^Q}{\partial \mathbf{x}}\, dV, \\
K_{\mathrm{fr}}^{PQ} &= \int_{S_{\mathrm{fr}} \cup S_{\mathrm{frs}}} \frac{\gamma}{\rho_f} \frac{1}{c_1} H^P\, H^Q\, dS,
\end{aligned}$$

$$
\begin{aligned}
K_{\text{fi}}^{PQ} &= \int_{S_{\text{fi}}} \frac{1}{a_1} H^P \, H^Q \, dS, \\
S_{\text{fs}}^{PM} &= \int_{S_{\text{fs}} \cup S_{\text{frs}}} H^P \, \mathbf{n}^- \cdot \mathbf{N}^M \, dS, \\
P_f^P &= \int_{S_{\text{ft}}} H^P T_0 \, dS, \\
M^{NM} &= \int_V \rho \mathbf{N}^N \cdot \mathbf{N}^M \, dV, \\
C_{(m)}^{NM} &= \int_V \alpha_c \rho \mathbf{N}^N \cdot \mathbf{N}^M \, dV, \\
I^N &= \int_V \boldsymbol{\beta}^N : \boldsymbol{\sigma} \, dV, \\
P^N &= \int_{S_{\text{t}}} \mathbf{N}^N \cdot \mathbf{t} \, dS,
\end{aligned}
$$

where $\boldsymbol{\beta}^N$ is the strain interpolator. This equation defines the discretized model. We see that the volumetric drag-related terms are "mass-like"; i.e., proportional to the fluid element mass matrix.

The term P_f^P is the nodal right-hand-side term for the acoustical degree of freedom p^P, or the applied "force" on this degree of freedom. This term is obtained by integration of the normal derivative of pressure per unit density of the acoustic medium over the surface area tributary to a boundary node.

In the case of coupled systems where the forces on the structure due to the fluid—$\left[S_{\text{fs}}^{QN}\right]^T \, p^Q$ are very small compared to the rest of the structural forces—the system can be solved in a "sequentially coupled" manner. The structural equations can be solved with the $\left[S_{\text{fs}}^{QN}\right]^T \, p^Q$ term omitted; i.e., in an analysis without fluid coupling. Subsequently, the fluid equations can be solved, with $\left[S_{\text{fs}}^{PM}\right] \, \ddot{u}^M$ imposed as a boundary condition. This two-step analysis is less expensive and advantageous for systems such as metal structures in air.

Time integration

The equations are integrated through time using the standard implicit (Abaqus/Standard) and explicit (Abaqus/Explicit) dynamic integration options. From the implicit integration operator we obtain relations between the variations of the solution variables (here represented by f) and their time derivatives:

$$
D_a \overset{\text{def}}{=} \frac{\delta \ddot{f}}{\delta f} = \frac{\delta p^P}{\delta \hat{p}^P}; \quad D_v \overset{\text{def}}{=} \frac{\delta \dot{f}}{\delta f}.
$$

The equations of evolution of the degrees of freedom can be written for the implicit case as

$$
\begin{aligned}
-\delta p^P \frac{1}{D_a}\Big\{ & (M_{\rm f}^{PQ} + M_{\rm fr}^{PQ})\ddot{p}^Q + (C_{\rm f}^{PQ} + C_{\rm fr}^{PQ})\dot{p}^Q \\
& + (K_{\rm f}^{PQ} + K_{\rm fr}^{PQ} + K_{\rm fi}^{PQ})p^Q - S_{\rm fs}^{PM}\ddot{u}^M - P_f^P \Big\} \\
& + \delta u^N \Big\{ I^N + M^{NM}\ddot{u}^M + C_{(m)}^{NM}\dot{u}^M + \Big[S_{\rm fs}^{QN}\Big]^T p^Q - P^N \Big\} = 0.
\end{aligned}
$$

The linearization of this equation is

$$
\begin{aligned}
-\delta p^P & \Big\{ (M_{\rm f}^{PQ} + M_{\rm fr}^{PQ}) + \frac{D_v}{D_a}(C_{\rm f}^{PQ} + C_{\rm fr}^{PQ}) + \frac{1}{D_a}(K_{\rm f}^{PQ} + K_{\rm fr}^{PQ} + K_{\rm fi}^{PQ}) \Big\} {\rm d}p^Q \\
& + \delta p^P S_{\rm fs}^{PM}\, {\rm d}u^M + \delta u^N \Big[S_{\rm fs}^{QN}\Big]^T {\rm d}p^Q \\
& + \delta u^N \Big\{ K^{NM} + D_a M^{NM} + D_v(C_{(m)}^{NM} + C_{(k)}^{NM}) \Big\} {\rm d}u^M = 0,
\end{aligned}
$$

where $\mathrm{d}p$ and $\mathrm{d}u$ are the corrections to the solution obtained from the Newton iteration, K^{NM} is the structural stiffness matrix, and $C_{(k)}^{NM}$ is the structural damping matrix. These equations are symmetric if the constituent stiffness, damping, and mass matrices are symmetric.

For explicit integration the fluid mass matrix is diagonalized in a manner similar to the treatment of structural mass. The explicit central difference procedure described in "Explicit dynamic analysis," Section 2.4.5, is applied to the coupled system of equations.

Summary of additional approximations of the direct integration transient formulation

As mentioned above, derivation of symmetric ordinary differential equations in the presence of volumetric drag requires some approximations in addition to those inherent in any finite element method. First, the spatial gradients of the ratio of volumetric drag to mass density in the fluid are neglected. This may be important in lossy, inhomogeneous acoustic media. Second, to maintain symmetry, the effect of volumetric drag on the fluid-solid boundary terms is neglected. Finally, the effect of volumetric drag on the radiation boundary conditions is approximate. If any of these effects is expected to be significant in an analysis, the user should realize that the results obtained are approximate.

Formulation for steady-state response using nodal degrees of freedom

The direct-solution steady-state dynamic analysis procedure is to be preferred over the transient formulation if volumetric drag is significant. This formulation uses the nodal degrees of freedom in the solid and acoustic regions directly to form a large linear system of equations defining the coupled structural-acoustic mechanics at a single frequency. If volumetric drag effects are not significant, the mode-based procedures (see below) are preferred because of their efficiency.

All model degrees of freedom and loads are assumed to be varying harmonically at an angular frequency Ω, so we can write

$$f = \tilde{f} \exp i\Omega t,$$

where $\tilde{f}$ is the constant complex amplitude of the variable f. Thus,

$$\dot{f} = i\Omega f, \quad \ddot{f} = -\Omega^2 f.$$

We begin with the equilibrium equation

$$\frac{\partial p}{\partial \mathbf{x}} + \gamma \dot{\mathbf{u}}^f + \rho_f \ddot{\mathbf{u}}^f = 0$$

and use the harmonic time-derivative relations to obtain

$$\frac{\partial \tilde{p}}{\partial \mathbf{x}} - \Omega^2 (\rho_f + \frac{\gamma}{i\Omega}) \tilde{\mathbf{u}}^f = 0.$$

We define the complex density, $\tilde{\rho}$, as

$$\tilde{\rho} \equiv \rho_f + \frac{\gamma}{i\Omega} \tag{2.9.1–18}$$

and, thus, write

$$\frac{\partial \tilde{p}}{\partial \mathbf{x}} - \Omega^2 (\tilde{\rho}) \tilde{\mathbf{u}}^f = 0. \tag{2.9.1–19}$$

The equilibrium equation is now in a form where the density is complex and the acoustic medium velocity does not enter. We divide this equation by $\tilde{\rho}$ and combine it with the second time derivative of the constitutive law, Equation 2.9.1–2, to obtain

$$-\Omega^2 \frac{1}{\mathrm{K}_f} \tilde{p} - \frac{\partial}{\partial \mathbf{x}} \cdot \left(\frac{1}{\tilde{\rho}} \frac{\partial \tilde{p}}{\partial \mathbf{x}} \right) = 0. \tag{2.9.1–20}$$

We have not used the assumption that the spatial gradient of r/ρ_f is small, as was done in the transient dynamics formulation.

Variational statement

The development of the variational statement parallels that for the case of transient dynamics, as though the volumetric drag were absent and the density complex. The variational statement is

$$\int_{V_f} \delta p \left[-\Omega^2 \frac{1}{\mathrm{K}_f} \tilde{p} - \frac{\partial}{\partial \mathbf{x}} \cdot \left(\frac{1}{\tilde{\rho}} \frac{\partial \tilde{p}}{\partial \mathbf{x}} \right) \right] dV = 0.$$

Integrating by parts, we have

$$-\int_{V_f} \delta p \frac{\Omega^2}{\mathrm{K}_f}\,\tilde{p}\,dV + \int_{V_f} \frac{1}{\tilde{\rho}}\frac{\partial \delta p}{\partial \mathbf{x}}\cdot\frac{\partial \tilde{p}}{\partial \mathbf{x}}\,dV + \int_S \delta p\,\frac{1}{\tilde{\rho}}\frac{\partial \tilde{p}}{\partial \mathbf{x}}\cdot\mathbf{n}^-\,dS = 0.$$

In steady state the boundary traction is defined as

$$\tilde{T}(\mathbf{x}) \equiv -\frac{1}{\tilde{\rho}}\frac{\partial \tilde{p}}{\partial \mathbf{x}}\cdot\mathbf{n}^- = -\Omega^2\mathbf{n}^-\cdot\tilde{\mathbf{u}}^f = \mathbf{n}^-\cdot\ddot{\mathbf{u}}^f.$$

This expression is not the Fourier transform of the boundary traction defined above for the transient case. The steady-state definition is based on the complex density and includes the volumetric drag effect in such a way that it is always equal to the acceleration of the fluid particles. The application of boundary conditions may be slightly different for some cases in steady state due to this definition of the traction.

On S_{fp}, $\tilde{p}$ is prescribed, analogous to transient analysis.

On S_{ft}, we prescribe

$$\tilde{T}_{\mathrm{ft}}(\mathbf{x}) \equiv T_0.$$

The condition $\mathbf{n}^-\cdot\ddot{\mathbf{u}}^f = T_0 = a_{in}$ is enforced, even in the presence of volumetric drag.

On S_{fr}, the reactive boundary between the acoustic medium and a rigid baffle, we apply

$$\tilde{T}_{\mathrm{fr}}(\mathbf{x}) \equiv -\left(\frac{i\Omega}{c_1} - \frac{\Omega^2}{k_1}\right)\tilde{p}. \qquad (2.9.1\text{–}21)$$

On S_{fi}, the radiating boundary, we apply the radiation boundary condition impedance in the same form as for the reactive boundary but with the parameters as defined in Equation 2.9.1–42 and Equation 2.9.1–43.

On S_{fs}, the acoustic-structural interface, we equate the displacement of the fluid and solid as in the transient case. However, the acoustic boundary traction coupling fluid to solid,

$$\tilde{T}(\mathbf{x}) = -\Omega^2\mathbf{n}^-\cdot\tilde{\mathbf{u}}^m,$$

can be applied without affecting the symmetry of the overall formulation. Consequently, the acoustic tractions in the steady-state case make no assumptions about volumetric drag.

On S_{frs}, the mixed impedance boundary and acoustic-structural boundary, the condition

$$\mathbf{n}^-\cdot(\dot{\mathbf{u}}^m - \dot{\mathbf{u}}^f) = \frac{1}{k_1}\dot{p} + \frac{1}{c_1}p \quad \text{on} \quad S_{\mathrm{frs}}$$

results in the definition

$$\tilde{T}_{\rm frs}(\mathbf{x}) = -\Omega^2 \mathbf{n}^- \cdot (\tilde{\mathbf{u}}^m) - \frac{i\Omega}{c_1}\,\tilde{p} + \frac{\Omega^2}{k_1}\,\tilde{p}.$$

In this case the effect of volumetric drag is included without approximation.

The final variational statement becomes

$$\begin{aligned}
&\int_{V_f} \left[-\Omega^2 \delta p \left(\frac{1}{\mathrm{K}_f}\,\tilde{p} \right) + \frac{1}{\tilde{\rho}}\,\frac{\partial \delta p}{\partial \mathbf{x}} \cdot \frac{\partial \tilde{p}}{\partial \mathbf{x}} \right] dV \\
&- \int_{S_{\rm ft}} \delta p\, a_{in}\, dS + \int_{S_{\rm fr} \cup S_{\rm fi}} \delta p \left(\frac{i\Omega}{c_1} - \frac{\Omega^2}{k_1} \right) \tilde{p}\, dS \\
&+ \int_{S_{\rm fs}} \delta p\, \Omega^2\, \mathbf{n}^- \cdot \tilde{\mathbf{u}}^m\, dS \\
&+ \int_{S_{\rm frs}} \delta p \left(\frac{i\Omega}{c_1}\,\tilde{p} - \frac{\Omega^2}{k_1}\,\tilde{p} + \Omega^2 \mathbf{n}^- \cdot \tilde{\mathbf{u}}^m \right) dS = 0.
\end{aligned}$$

This equation is formally identical to Equation 2.9.1–4, except for the pressure "stiffness" term, the radiation boundary conditions, and the imposed boundary traction term. Because the volumetric drag effect is contained in the complex density, the acoustic-structural boundary term in this formulation does not have the limitation that the volumetric drag must be small compared to other effects in the acoustic medium. In addition, in this formulation the applied flux on an acoustic boundary represents the inward acceleration of the acoustic medium, whether or not the volumetric drag is large. Finally, the radiation boundary conditions do not make any approximations with regard to the volumetric drag parameter.

The above equation uses the complex density, $1/\tilde{\rho}$. We manipulate it into a form that has real coefficients and an additional time derivative through the relations

$$\frac{1}{\tilde{\rho}} = \frac{\rho_f}{\rho_f^2 + \gamma^2/\Omega^2} + i \frac{\gamma/\Omega}{\rho_f^2 + \gamma^2/\Omega^2}, \quad i\frac{\partial p}{\partial \mathbf{x}} = \frac{1}{\Omega}\,\frac{\partial \dot{p}}{\partial \mathbf{x}},$$

to obtain

$$\begin{aligned}
&\int_{V_f} \left[-\delta p \left(\frac{\Omega^2}{\mathrm{K}_f}\,\tilde{p} \right) + \frac{\rho_f}{\rho_f^2 + \gamma^2/\Omega^2}\,\frac{\partial \delta p}{\partial \mathbf{x}} \cdot \frac{\partial \tilde{p}}{\partial \mathbf{x}} + (i\Omega) \frac{\gamma/\Omega^2}{\rho_f^2 + \gamma^2/\Omega^2}\,\frac{\partial \delta p}{\partial \mathbf{x}} \cdot \frac{\partial \tilde{p}}{\partial \mathbf{x}} \right] dV \\
&\qquad - \int_{S_{\rm ft}} \delta p\, a_{in}\, dS + \int_{S_{\rm fs} \cup S_{\rm frs}} \delta p\, \Omega^2\, \mathbf{n}^- \cdot \tilde{\mathbf{u}}^m\, dS \\
&\qquad + \int_{S_{\rm fr} \cup S_{\rm fi} \cup S_{\rm frs}} \delta p \left(\frac{i\Omega}{c_1}\,\tilde{p} - \frac{\Omega^2}{k_1}\,\tilde{p} \right) dS = 0.
\end{aligned} \tag{2.9.1–22}$$

The discretized finite element equations

Applying Galerkin's principle, the finite element equations are derived as before. We arrive again at Equation 2.9.1–17 with the same matrices except for the damping and stiffness matrices of the acoustic elements and the surfaces that have imposed impedance conditions, which now appear as

$$
\begin{aligned}
C_{\mathrm{f}}^{PQ} &= \int_{V_f} \frac{\gamma/\Omega^2}{\rho_f^2 + \gamma^2/\Omega^2} \frac{\partial H^P}{\partial \mathbf{x}} \cdot \frac{\partial H^Q}{\partial \mathbf{x}}\, dV, \\
K_{\mathrm{f}}^{PQ} &= \int_{V_f} \frac{\rho_f}{\rho_f^2 + \gamma^2/\Omega^2} \frac{\partial H^P}{\partial \mathbf{x}} \cdot \frac{\partial H^Q}{\partial \mathbf{x}}\, dV, \\
C_{\mathrm{fr}}^{PQ} &= \int_{S_{\mathrm{fr}}} \frac{1}{c_1} H^P\, H^Q\, dS, \\
K_{\mathrm{fr}}^{PQ} &= 0, \\
K_{\mathrm{fi}}^{PQ} &= 0.
\end{aligned}
$$

The matrix modeling loss to volumetric drag is proportional to the fluid stiffness matrix in this formulation.

For steady-state harmonic response we assume that the structure undergoes small harmonic vibrations, identified by the prefix Δ, about a deformed, stressed base state, which is identified by the subscript 0. Hence, the total stress can be written in the form

$$
\boldsymbol{\sigma} = \boldsymbol{\sigma}_0 + \Delta\boldsymbol{\sigma} = \boldsymbol{\sigma}_0 + \mathbf{D}^{el} : \left(\Delta\boldsymbol{\varepsilon} + is\Delta\boldsymbol{\varepsilon} + \beta_c \Delta\dot{\boldsymbol{\varepsilon}}\right),
$$

where $\boldsymbol{\sigma}_0$ is the stress in the base state; $\mathbf{D}^{el}$ is the elasticity matrix for the material; β_c is the stiffness proportional damping factor chosen for the material (to give the stiffness proportional contribution to the Rayleigh damping, thus introducing the viscous part of the material behavior); and, from the discretization assumption,

$$
\Delta\boldsymbol{\varepsilon} = \boldsymbol{\beta}^M\, \Delta u^M.
$$

To solve the steady-state problem, we assume that the governing equations are satisfied in the base state, and we linearize these equations in terms of the harmonic oscillations. For the internal force vector this yields

$$
\Delta I^N = \left(iC_{(s)}^{NM} + K^{NM}\right) \Delta u^M + C_{(k)}^{NM}\, \Delta\dot{u}^M;
$$

and Equation 2.9.1–17 can be rewritten, using the time-harmonic relations, as

$$-\delta\hat{p}^P\left\{\left[-\Omega^2(M_{\rm f}^{PQ}+M_{\rm fr}^{PQ})+i\Omega(C_{\rm f}^{PQ}+C_{\rm fr}^{PQ})+iC_{\rm f(s)}^{PQ}+K_{\rm f}^{PQ}\right]\Delta\tilde{p}^Q+\Omega^2 S_{\rm fs}^{PM}\,\Delta\tilde{u}^M-\Delta\tilde{P}_f^P\right\}$$
$$+\delta u^N\left\{\left[-\Omega^2 M^{NM}+i\Omega(C_{(m)}^{NM}+C_{(k)}^{NM})+iC_{(s)}^{NM}+K^{NM}\right]\Delta\tilde{u}^M+\left[S_{\rm fs}^{QN}\right]^T\Delta\tilde{p}^Q-\Delta\tilde{P}^N\right\}=0, \tag{2.9.1–23}$$

with

$$K^{NM}=\int_V\left[\frac{\partial\boldsymbol{\beta}^N}{\partial u^M}:\boldsymbol{\sigma}_0+\boldsymbol{\beta}^N:\mathbf{D}^{el}:\boldsymbol{\beta}^M\right]\,dV$$

(this stiffness includes the initial stress matrix, so "stress stiffening" and "load stiffness" effects associated with the base state stress and loads are included) and

$$C_{(k)}^{NM}=\int_V\left[\beta_c\,\boldsymbol{\beta}^N:\mathbf{D}^{el}:\boldsymbol{\beta}^M\right]\,dV.$$

We have also added the "global" parts of the "structural damping" terms

$$C_{(s)}^{NM}=\int_V s\,\boldsymbol{\beta}^N:\mathbf{D}^{el}:\boldsymbol{\beta}^M\,dV+s_g K^{NM}$$

and

$$C_{\rm f(s)}^{PQ}\equiv s_g K_{\rm f}^{PQ}$$

to the equation. These damping terms model finite energy loss in the low-frequency limit in steady-state dynamics—see "Direct steady-state dynamic analysis," Section 2.6.1, and "Subspace-based steady-state dynamic analysis," Section 2.6.2. It should be noted that the acoustic "structural damping" operator inherits the frequency dependence of the acoustic stiffness matrix caused by volumetric drag.

We assume that the loads and (because of linearity) the response are harmonic; hence, we can write

$$\Delta\tilde{p}^Q=\left(\Re\left(\tilde{p}^Q\right)+i\,\Im\left(\tilde{p}^Q\right)\right)\exp i\Omega t$$
$$\Delta\tilde{u}^M=\left(\Re\left(\tilde{u}^M\right)+i\,\Im\left(\tilde{u}^M\right)\right)\exp i\Omega t$$

and

$$\Delta\tilde{P}^N=\left(\Re\left(\tilde{P}^N\right)+i\,\Im\left(\tilde{P}^N\right)\right)\exp i\Omega t$$
$$\Delta\tilde{P}_f^P=\left(\Re\left(\tilde{P}_f^P\right)+i\,\Im\left(\tilde{P}_f^P\right)\right)\exp i\Omega t,$$

where $\Re\left(\tilde{p}^Q\right)$, $\Re\left(\tilde{u}^M\right)$, $\Im\left(\tilde{p}^Q\right)$, and $\Im\left(\tilde{u}^M\right)$ are the real and imaginary parts of the amplitudes of the response; $\Re\left(\tilde{P}^N\right)$ and $\Im\left(\tilde{P}^N\right)$ are the real and imaginary parts of the amplitude of the force applied to the structure; $\Re(\tilde{P}_f^P)$ and $\Im(\tilde{P}_f^P)$ are the real and imaginary parts of the amplitude of the acoustic

traction (dimensions of volumetric acceleration) applied to the fluid; and Ω is the circular frequency. We substitute these equations into Equation 2.9.1–23 and use the time-harmonic form of Equation 2.9.1–16, $\delta\hat{p}^P = -\Omega^{-2}\delta p^P$, which yields the coupled complex linear equation system

$$\begin{bmatrix} \Re\left[A_{\mathrm{f}}^{PQ}\right] & \Im\left[A_{\mathrm{f}}^{PQ}\right] & \left[S_{\mathrm{fs}}^{PM}\right] & 0 \\ \Im\left[A_{\mathrm{f}}^{PQ}\right] & -\Re\left[A_{\mathrm{f}}^{PQ}\right] & 0 & -\left[S_{\mathrm{fs}}^{PM}\right] \\ \left[S_{\mathrm{fs}}^{QN}\right]^T & 0 & \Re\left[A_{\mathrm{s}}^{NM}\right] & \Im\left[A_{\mathrm{s}}^{NM}\right] \\ 0 & -\left[S_{\mathrm{fs}}^{QN}\right]^T & \Im\left[A_{\mathrm{s}}^{NM}\right] & -\Re\left[A_{\mathrm{s}}^{NM}\right] \end{bmatrix} \begin{Bmatrix} \Re\left(\tilde{p}^Q\right) \\ \Im\left(\tilde{p}^Q\right) \\ \Re\left(\tilde{u}^M\right) \\ \Im\left(\tilde{u}^M\right) \end{Bmatrix} = \begin{Bmatrix} \Omega^{-2}\Re\left(\tilde{P}_f^P\right) \\ -\Omega^{-2}\Im\left(\tilde{P}_f^P\right) \\ \Re\left(\tilde{P}^N\right) \\ -\Im\left(\tilde{P}^N\right) \end{Bmatrix}, \quad (2.9.1\text{–}24)$$

where

$$\Re\left[A_{\mathrm{f}}^{PQ}\right] = \Omega^{-2}(K_{\mathrm{f}}^{PQ} + K_{\mathrm{fr}}^{PQ}) - (M_{\mathrm{f}}^{PQ} + M_{\mathrm{fr}}^{PQ})$$
$$\Im\left[A_{\mathrm{f}}^{PQ}\right] = -\Omega^{-1}(C_{\mathrm{f}}^{PQ} + C_{\mathrm{fr}}^{PQ}) - \Omega^{-2}C_{\mathrm{f(s)}}^{PQ}$$

and

$$\Re\left[A_{\mathrm{s}}^{NM}\right] = K^{NM} - \Omega^2\, M^{NM}$$
$$\Im\left[A_{\mathrm{s}}^{NM}\right] = -\Omega\,(C_{(m)}^{NM} + C_{(k)}^{NM}) - C_{(\mathrm{s})}^{NM}.$$

If K^{NM} is symmetric, Equation 2.9.1–24 is symmetric. The system may be quite large, because the real and imaginary parts of the structural degrees of freedom and of the pressure in the fluid all appear in the system. This set of equations is solved for each frequency requested in the direct-solution steady-state dynamics procedure. If damping is absent, the user can specify that only the real matrix equation should be factored in the analysis. Nonzero volumetric drag values (γ) for the acoustic medium and nonzero $1/c_1$ values for the impedances represent damping. As mentioned above for the transient case, the coupled system can be split into an uncoupled structural analysis and an acoustic analysis driven by the structural response, provided the fluid forces on the structure are small.

Formulation for eigenvalue extraction and mode-based procedures

From the discretized equation, Equation 2.9.1–17, we can write the frequency domain problem as

$$\begin{bmatrix} K_{\mathrm{s}} & {S_{\mathrm{fs}}}^T \\ 0 & K_{\mathrm{f}} \end{bmatrix} \begin{Bmatrix} u \\ p \end{Bmatrix} + i\omega \begin{bmatrix} C_{(m)} & 0 \\ 0 & C_{\mathrm{f}} \end{bmatrix} \begin{Bmatrix} u \\ p \end{Bmatrix} - \omega^2 \begin{bmatrix} M_{\mathrm{s}} & 0 \\ -S_{\mathrm{fs}} & M_{\mathrm{f}} \end{bmatrix} \begin{Bmatrix} u \\ p \end{Bmatrix} = \begin{Bmatrix} P_s \\ P_s \end{Bmatrix},$$

where ω is a natural (as opposed to forced response) frequency. The indices have been suppressed for brevity. This system is due to Zienkiewicz and Newton (1969) and is used in Abaqus as the starting point for mode-based procedures. Suppressing any damping terms, forcing, and any terms associated with a reactive surface,

$$\begin{bmatrix} K_{\rm s} & {S_{\rm fs}}^T \\ 0 & K_{\rm f} \end{bmatrix} \begin{Bmatrix} u \\ p \end{Bmatrix} - \omega^2 \begin{bmatrix} M_{\rm s} & 0 \\ -S_{\rm fs} & M_{\rm f} \end{bmatrix} \begin{Bmatrix} u \\ p \end{Bmatrix} = 0. \tag{2.9.1–25}$$

Interpreted as a linear eigenvalue problem (where ω^2 is the eigenvalue), this equation cannot be solved directly in Abaqus due to the unsymmetric stiffness and mass matrices. However, it can be shown that these equations do yield real-valued natural frequencies and modes, suggesting that they can be rewritten in symmetric forms.

Application of the modes of Equation 2.9.1–25 to form a reduced system (see below) must be done with some caution, since this unsymmetric system has distinct left and right eigenvector sets. In particular, the "singular modes" associated with zero frequency are of interest because they describe the low-frequency limiting behavior of the system (or the "rigid-body motion" in a kinematic sense) and are, therefore, essential for the construction of an accurate projected frequency domain operator. The right singular modes of the coupled system are

$$\phi_0^{R,S} \equiv \left(K_s u_0^{R,S} = 0,\ p_0^{R,S} = 0 \right),$$

$$\phi_0^{R,A} \equiv \left(K_s u_0^{R,A} = -S_{\rm fs}^T p_0^{R,A},\ K_f p_0^{R,A} = 0 \right).$$

In other words, there is a "structural" singular right mode $\phi_0^{R,S}$ associated with the kernel of K_s and an "acoustic" singular right mode $\phi_0^{R,A}$ associated with the kernel of K_f. The left singular modes are solutions to

$$\phi_0^T \begin{bmatrix} K_{\rm s} & {S_{\rm fs}}^T \\ 0 & K_{\rm f} \end{bmatrix} = 0$$

and are

$$\phi_0^{L,S} \equiv \left(K_s u_0^{L,S} = 0,\ K_f p_0^{L,S} = -S_{\rm fs} u_0^{L,S} \right),$$

$$\phi_0^{L,A} \equiv \left(u_0^{L,A} = 0,\ K_f p_0^{L,A} = 0 \right).$$

The right acoustic and left structural singular modes are coupled, with nontrivial fields on the structural and acoustic domains. These coupled singular modes are a consequence of the stiffness term in Equation 2.9.1–25, and they must be computed if this system is to be projected.

An alternative frequency domain formulation, due to Everstine (1981), involves the substitution $p \mapsto i\omega q$ and results in a formally symmetric system:

$$\begin{bmatrix} K_{\rm s} & 0 \\ 0 & K_{\rm f} \end{bmatrix} \begin{Bmatrix} u \\ q \end{Bmatrix} + i\omega \begin{bmatrix} C_{(m)} & {S_{\rm fs}}^T \\ S_{\rm fs} & C_{\rm f} \end{bmatrix} \begin{Bmatrix} u \\ q \end{Bmatrix} - \omega^2 \begin{bmatrix} M_{\rm s} & 0 \\ 0 & M_{\rm f} \end{bmatrix} \begin{Bmatrix} u \\ q \end{Bmatrix} = \begin{Bmatrix} P_s \\ \frac{P_s}{i\omega} \end{Bmatrix}.$$

The corresponding coupled eigenproblem is quadratic, but the singular mode structure of this system is much simpler—the left and right pairs are identical due to symmetry, and the singular modes are uncoupled due to the diagonal structure of the stiffness matrix. The modes are simply

$$\phi_0^S \equiv \left(K_s u_0^S = 0,\ p_0^S = 0\right),$$

$$\phi_0^A \equiv \left(u_0^A = 0,\ K_f p_0^A = 0\right).$$

Lanczos formulation

Introducing an auxiliary variable, $\psi = p/\omega^2$, augmenting the system of equations with $K_f p = \omega^2 K_f \psi$, and manipulating the equations yields

$$\left(\begin{bmatrix} K_{\mathrm{s}} & {S_{\mathrm{fs}}}^T & 0 \\ S_{\mathrm{fs}} & -M_{\mathrm{f}} & K_{\mathrm{f}} \\ 0 & {K_{\mathrm{f}}}^T & 0 \end{bmatrix} - \omega^2 \begin{bmatrix} M_{\mathrm{s}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & K_{\mathrm{f}} \end{bmatrix} \right) \begin{Bmatrix} u \\ p \\ \psi \end{Bmatrix} = 0. \tag{2.9.1–26}$$

This augmented system of equations is due to Ohayon and is used only for Lanczos eigenvalue extraction. The auxiliary variable ψ is internal to Abaqus/Standard and is not available for output. If K_{f} is singular, orthogonalization against the singular acoustic modes is done in the Lanczos eigensolver.

The left and right eigenvectors for the original system of equations, Equation 2.9.1–25, can be constructed from the Lanczos solution. As mentioned above, the singular modes are essential for construction of an accurate projected operator. It is easy to verify that the Lanczos system has the following structural singular mode:

$$\phi_0^S \equiv \left(K_{\mathrm{s}} u_0^S = 0,\ p_0^S = 0,\ K_{\mathrm{f}} \psi_0^S = -S_{\mathrm{fs}} u_0^S\right).$$

However, if we seek nontrivial acoustic singular modes (i.e., $p_0^A \neq 0$, such that $K_{\mathrm{f}} p_0^A = 0$), we easily find that $K_{\mathrm{s}} u_0^A = -S_{\mathrm{fs}}^T p_0^A$ but also that

$$K_{\mathrm{f}} \psi_0^A = M_{\mathrm{f}} p_0^A - S_{\mathrm{fs}} u_0^A.$$

If a nontrivial p_0^A exists, K_{f} is singular; therefore, for a solution ψ_0^A to exist, the right-hand-side must be orthogonal to the null space of K_{f}. But we quickly observe that

$${p_0^A}^T \left(M_{\mathrm{f}} p_0^A - S_{\mathrm{fs}} u_0^A\right) = {p_0^A}^T M_{\mathrm{f}} p_0^A + {p_0^A}^T S_{\mathrm{fs}}^T K_{\mathrm{s}}^{-1} S_{\mathrm{fs}} p_0^A > 0 \ \forall\ p_0^A.$$

Consequently, to find an acoustic singular mode using the Lanczos formulation, we construct a perturbation "force" f_{null} such that $f_{\mathrm{null}}^T p_0^A \neq 0$. The Lanczos formulation will yield the nontrivial singular acoustic mode

$$\phi_0^A \equiv \left({K_{\mathrm{s}}}^{-1} u_0^A = -S_{\mathrm{fs}}^T p_0^A,\ K_{\mathrm{f}} p_0^A = 0,\ K_{\mathrm{f}} \psi_0^A = M_{\mathrm{f}} p_0^A - S_{\mathrm{fs}} u_0^A + f_{\mathrm{null}}\right).$$

The left and right eigenvectors of the original, unsymmetric system Equation 2.9.1–25, including the singular modes, can be constructed from the Lanczos solutions $(u,\ p,\ \psi)$:

$$\phi^L = \left\{ \begin{array}{c} v \\ \psi \end{array} \right\}; \quad \phi^R = \left\{ \begin{array}{c} u \\ p \end{array} \right\},$$

where

$$v = \left\{ \begin{array}{c} 0, \text{if a singular acoustic mode} \\ u, \text{otherwise} \end{array} \right\}.$$

For any nonsingular acoustic mode α, $p_\alpha = \omega_\alpha^2 \psi_\alpha$, where ω_α is the circular eigenfrequency. The left and right eigenvector subspaces are then used to compute modal quantities (generalized mass, participation factors, and effective mass) and to project the mass, stiffness, and damping matrices in mode-based procedures (such as subspace-based steady-state dynamic analysis or transient modal dynamic analysis) to obtain a reduced system of equations. Most of these computations are conducted in a very similar fashion to the way they are carried out in the pure structural problem and will not be discussed here. In addition, for each mode an acoustic fraction of the generalized mass is computed as the ratio between acoustic contributions to the generalized mass and to the total generalized mass.

The only exception worth a brief discussion is the choice for the calculation of the acoustic participation factors and effective masses, as follows. First, a "rigid body" acoustic mode, T_{a}, analogous to the rigid body modes for the structural problem outlined in "Variables associated with the natural modes of a model," Section 2.5.2, is chosen to be a constant pressure field of unity. A total "acoustic mass" is then defined as $m_{\mathrm{ac}} = {T_{\mathrm{a}}}^T M_{\mathrm{f}} T_{\mathrm{a}}$. Left and right acoustic participation factors are defined as

$$\Gamma_\alpha^L = \frac{1}{m_\alpha} \begin{bmatrix} v^M \\ \psi^Q \end{bmatrix}^T \begin{bmatrix} M^{NM} & 0 \\ -S_{\mathrm{fs}} & M_{\mathrm{f}} \end{bmatrix} \begin{bmatrix} 0 \\ T_{\mathrm{a}} \end{bmatrix} = \frac{1}{m_\alpha} \psi^Q M_{\mathrm{f}} T_{\mathrm{a}}$$

and

$$\Gamma_\alpha^R = \frac{1}{m_\alpha} \begin{bmatrix} 0 \\ T_{\mathrm{a}} \end{bmatrix}^T \begin{bmatrix} M^{NM} & 0 \\ -S_{\mathrm{fs}} & M_{\mathrm{f}} \end{bmatrix} \begin{bmatrix} u^M \\ p^Q \end{bmatrix} = \frac{1}{m_\alpha} \left(-T_{\mathrm{a}} S_{\mathrm{fs}} u^M + T_{\mathrm{a}} M_{\mathrm{f}} p^Q \right).$$

Abaqus/Standard will then report the acoustic participation factor computed as

$$\Gamma_\alpha^{\mathrm{ac}} = \sqrt{\Gamma_\alpha^L \Gamma_\alpha^R / m_{\mathrm{ac}}}$$

and an acoustic effective mass computed as

$$m_{\alpha\ \mathrm{ac}}^{\mathrm{eff}} = \left(\Gamma_\alpha^{\mathrm{ac}}\right)^2 m_\alpha.$$

The scaling by m_{ac} in the equation for $\Gamma_\alpha^{\mathrm{ac}}$ is arbitrary. However, this scaling ensures that when all eigenmodes are extracted, the sum of all acoustic effective masses is 1.0 (minus the contributions from nodes constrained in the acoustic degree of freedom).

Frequency-domain solution using projections onto modal spaces

Distinct modal space projection methods for coupled forced structural-acoustic response exist in Abaqus for the following cases: using coupled modes from Lanczos, using uncoupled modes from Lanczos, and using uncoupled modes from Abaqus/AMS. In the Lanczos mode cases the forced response is computed using the Zienkiewicz-Newton equation, with separate right and left projection operators. In the Abaqus/AMS uncoupled mode case the Everstine equation is used for the forced response and the right and left projection operators are identical. This case is described in more detail below.

Using uncoupled Abaqus/AMS modes

In this case the Everstine equation is used for the coupled forced response problem and modes are computed from decoupled structural and acoustic Abaqus/AMS runs. In nodal degrees of freedom the forced response is governed by

$$\begin{bmatrix} K_{\mathrm{s}} & 0 \\ 0 & K_{\mathrm{f}} \end{bmatrix} \begin{Bmatrix} u \\ q \end{Bmatrix} + i \begin{bmatrix} C_{\mathrm{s}} & \Omega {S_{\mathrm{fs}}}^T \\ \Omega S_{\mathrm{fs}} & C_{\mathrm{f}} \end{bmatrix} \begin{Bmatrix} u \\ q \end{Bmatrix} - \Omega^2 \begin{bmatrix} M_{\mathrm{s}} & 0 \\ 0 & M_{\mathrm{f}} \end{bmatrix} \begin{Bmatrix} u \\ q \end{Bmatrix} = \begin{Bmatrix} P_s \\ \frac{P_f}{i\Omega} \end{Bmatrix},$$

where C_s and C_f here are the complete assembled damping matrices for the structure and fluid, including viscous and structural damping, as well as boundary impedance effects. Using transformations constructed from the acoustic and structural modes,

$$\left\{\phi_i^S\right\} \mapsto \Phi_S,$$

$$\left\{\phi_j^A\right\} \mapsto \Phi_A,$$

and representations of the structural and acoustic fields in the spaces spanned by these modes,

$$u \mapsto \Phi_S \left(\tilde{u} + i\tilde{v}\right),$$

$$q \mapsto \Phi_A \left(\tilde{q} + i\tilde{r}\right),$$

we obtain

$$\begin{bmatrix} k_{\mathrm{s}} - \Omega^2 m_{\mathrm{s}} & \Omega c_{\mathrm{s}} & 0 & -\Omega s^T \\ \Omega c_{\mathrm{s}} & -k_{\mathrm{s}} + \Omega^2 m_{\mathrm{s}} & -\Omega s^T & 0 \\ 0 & -\Omega s_{\mathrm{fs}} & k_{\mathrm{f}} - \Omega^2 m_{\mathrm{f}} & \Omega c_{\mathrm{f}} \\ -\Omega s & 0 & \Omega c_{\mathrm{f}} & -k_{\mathrm{f}} + \Omega^2 m_{\mathrm{f}} \end{bmatrix} \begin{Bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{q} \\ \tilde{r} \end{Bmatrix} = \begin{Bmatrix} \Re\left(\tilde{f}_{\mathrm{s}}\right) \\ -\Im\left(\tilde{f}_{\mathrm{s}}\right) \\ \Re\left(\tilde{f}_{\mathrm{f}}\right) \\ -\Im\left(\tilde{f}_{\mathrm{f}}\right) \end{Bmatrix}.$$

The terms in this matrix correspond to the nodal degree-of-freedom operators, projected onto the modal spaces. The damping and coupling matrices in modal coordinates are full and unsymmetric.

Volumetric drag and fluid viscosity

The medium supporting acoustic waves may be flowing through a porous matrix, such as fiberglass used for sound deadening. In this case the parameter γ is the *flow resistance*, the pressure drop required to force a unit flow through the porous matrix. A propagating plane wave with nominal particle velocity $\dot{\mathbf{u}}^f$ loses energy at a rate

$$\dot{E} = -\gamma \left[\dot{\mathbf{u}}^f\right]^2. \tag{2.9.1–27}$$

Fluids also exhibit momentum losses without a porous matrix resistive medium through coefficients of shear viscosity μ and bulk viscosity η. These are proportionality constants between components of the stress and spatial derivatives of the shear strain rate and volumetric strain rate, respectively. In fluid mechanics the shear viscosity term μ is usually more important than the bulk term η; however, acoustics is the study of volumetrically straining flows, so both constants can be important. The linearized Navier-Stokes equations for adiabatic perturbations about a base state can be expressed in terms of the pressure field alone (Morse and Ingard, 1968):

$$\frac{\partial}{\partial \mathbf{x}} \cdot \frac{\partial p}{\partial \mathbf{x}} = \frac{\rho_f}{K_f} \ddot{p} - \frac{\eta + \frac{4}{3}\mu}{K_f} \frac{\partial}{\partial \mathbf{x}} \cdot \frac{\partial \dot{p}}{\partial \mathbf{x}}. \tag{2.9.1–28}$$

In steady state this linearized equation can be written in the form of Equation 2.9.1–19, with

$$\Re\left(\tilde{\rho}_f\right) = \left[\frac{1}{1 + \left(\frac{\Omega}{K_f}\right)^2 \left(\eta + \frac{4}{3}\mu\right)^2}\right] \rho_f,$$

$$\Im\left(\tilde{\rho}_f\right) = -\left[\frac{\left(\frac{\Omega}{K_f}\right)\left(\eta + \frac{4}{3}\mu\right)}{1 + \left(\frac{\Omega}{K_f}\right)^2 \left(\eta + \frac{4}{3}\mu\right)^2}\right] \rho_f,$$

so that the viscosity effects can be modeled as a volumetric drag parameter with the value

$$\gamma = \left[\frac{\left(\frac{\Omega}{K_f}\right)^2 \left(\eta + \frac{4}{3}\mu\right)}{1 + \left(\frac{\Omega}{K_f}\right)^2 \left(\eta + \frac{4}{3}\mu\right)^2}\right] \rho_f K_f.$$

If the combined viscosity effects are small,

$$\frac{\partial}{\partial \mathbf{x}} \cdot \frac{\partial p}{\partial \mathbf{x}} \approx \frac{\rho_f}{K_f} \ddot{p}, \tag{2.9.1–29}$$

so we can write

$$K_f \frac{\partial}{\partial \mathbf{x}} \cdot \frac{\partial p}{\partial \mathbf{x}} - \rho_f \frac{\partial^2}{\partial t^2} p + (\eta + \frac{4}{3}\mu)(\frac{\rho_f}{K_f}) \frac{\partial^3}{\partial t^3} p = 0. \tag{2.9.1–30}$$

In steady-state form

$$\frac{1}{\rho_f} \frac{\partial}{\partial \mathbf{x}} \cdot \frac{\partial p}{\partial \mathbf{x}} + \Omega^2 \left(\frac{1}{K_f} - i\Omega \frac{\eta + \frac{4}{3}\mu}{K_f^2} \right) p = 0, \tag{2.9.1–31}$$

where Ω is the forcing frequency. This leads to the following analogy between viscous fluid losses and volumetric drag or flow resistance:

$$\gamma = \frac{\Omega^2 \rho_f}{K_f} \left(\eta + \frac{4}{3}\mu \right), \tag{2.9.1–32}$$

with density constant with respect to frequency. The energy loss rate for a propagating plane wave in this linearized, adiabatic, small-viscosity case is

$$\dot{E} = - \left(\eta + \frac{4}{3}\mu \right) \frac{\Omega^2 \rho_f}{K_f} \left[\dot{\mathbf{u}}^f \right]^2. \tag{2.9.1–33}$$

Acoustic output quantities

Several secondary quantities are useful in acoustic analysis, derived from the fundamental acoustic pressure field variable. In steady-state dynamics the acoustic particle velocity at any field point is

$$\tilde{\mathbf{v}} = \frac{i}{\tilde{\rho}\Omega} \frac{\partial \tilde{p}}{\partial \mathbf{x}}.$$

The acoustic intensity vector, a measure of the rate of flow of energy at a point, is

$$\tilde{\mathbf{I}} = -\frac{1}{2} \boldsymbol{\sigma} \cdot \hat{\tilde{\mathbf{v}}}.$$

In an acoustic medium the stress tensor is simply the acoustic pressure times the identity tensor, $\boldsymbol{\sigma} = -p\mathbf{I}$, so this expression simplifies to

$$\tilde{\mathbf{I}} = \frac{-1}{2i\hat{\tilde{\rho}}\Omega} \left[\tilde{p} \frac{\partial \hat{\tilde{p}}}{\partial \mathbf{x}} \right].$$

The "hats" denote complex conjugation. The real part of the intensity is referred to as the "active intensity," and the imaginary part is the "reactive intensity."

Acoustic contribution factors

Acoustic contribution factors help the user interpret the behavior of a coupled structural-acoustic system by showing the relationship between the acoustic pressure and either specific structural surfaces or specific structural modes. In the literature they are sometimes referred to as acoustic "participation factors," but since this term is used in Abaqus to describe characteristics of modes (see "Variables associated with the natural modes of a model," Section 2.5.2), a different nomenclature is chosen here.

First, consider an acoustic medium in contact with a structure undergoing time-harmonic vibration. The structure exerts a traction on the fluid at each point on the wetted surface, causing harmonic pressure in the acoustic medium. In a given solution to a coupled forced response problem, it is sometimes useful to separate the pressure into constituent parts, each due to the vibration of a portion of the wetted surface. For example, in an automotive acoustic problem it may be useful to determine the parts of the acoustic pressure field due to the windows, floor, and other panels separately. The pressure field π_J generated by some given structural vibration acting only on structural surface J, with all other parts of the wetted surface held fixed, is defined as the acoustic contribution factor of that surface:

$$A_{\mathrm{f}}\pi_J \equiv -\Omega^2 S_J u, \tag{2.9.1–34}$$

where $A_{\mathrm{f}} \equiv K_{\mathrm{f}} + i\Omega C_{\mathrm{f}} - \Omega^2 M_{\mathrm{f}}$ and S_J is the coupling matrix associated with surface partition J. S_J can correspond to a group of disjoint surfaces (for example, all the window glass in an automobile) or to a single node. Because the natural boundary condition in Abaqus for acoustic elements is a rigid wall, Equation 2.9.1–34 corresponds physically to an acoustic field coupled to the structure only at surface J, with all other bounding surfaces rigid.

For example, if a single panel's acoustic contribution is separated from the total acoustic pressure,

$$p = p_0 + \pi_1,$$

the coupled system of equations for the structural acoustic problem can be written

$$\begin{bmatrix} A_{\mathrm{s}} & S_{\mathrm{fs}}^T & S_1^T \\ \Omega^2 S_{\mathrm{fs}} & A_{\mathrm{f}} & A_{\mathrm{f}} \\ \Omega^2 S_1 & 0 & A_{\mathrm{f}} \end{bmatrix} \begin{pmatrix} u \\ p_0 \\ \pi_1 \end{pmatrix} = \begin{pmatrix} P \\ P_f \\ 0 \end{pmatrix}, \tag{2.9.1–35}$$

where $A_{\mathrm{s}} \equiv K_{\mathrm{s}} + i\Omega C_{\mathrm{s}} - \Omega^2 M_{\mathrm{s}}$. This equation makes it clear that the panel's acoustic contribution factor depends on the solution to the specific coupled harmonic forced response problem. However, it is more efficient to solve for p and u instead and then solve for π_1 using Equation 2.9.1–34.

When subspace-based steady-state dynamics or mode-based steady-state dynamics is used, A_{f} and S_J are projected; in turn, these projected matrices depend on whether the preceding eigenanalysis step was coupled or uncoupled. For the uncoupled case separate modal transformations Φ_{f} and Φ_{s} correspond to the acoustic and structural modes, and

$$\pi_J \equiv \Phi_{\mathrm{f}}\tilde{\pi}_J,$$

$$u \equiv \Phi_{\mathrm{s}} \tilde{u}.$$

The transformed equation defining π_J becomes

$$\Phi_{\mathrm{f}}^T A_f \Phi_{\mathrm{f}} \tilde{\pi}_J = -\Omega^2 \Phi_{\mathrm{f}}^T S_J \Phi_{\mathrm{s}} \tilde{u}.$$

The contribution of a specific mode to the acoustic pressure of a forced harmonic coupled system may be of interest as well. Physically, a modal acoustic contribution factor is the part of the acoustic field in a forced response problem due to the action of one structural (or coupled) mode on the acoustic fluid. The calculation of a modal acoustic contribution factor depends on whether the modes in question are uncoupled or coupled structural-acoustic modes. However, its definition is analogous to the surface or panel acoustic contribution factor: it is the acoustic response due to forcing on the wetted surface due only to a single mode of interest, with all other modes held fixed. Starting from Equation 2.9.1–34, but using the entire wetted surface coupling operator S,

$$A_{\mathrm{f}} \pi_J \equiv -\Omega^2 S u_J, \tag{2.9.1–36}$$

where u_J is the structural response of the coupled problem, restricted to mode J. If coupled mode transformations are used, this equation becomes

$$\hat{\Phi}_{\mathrm{f}}^T A_f \hat{\Phi}_{\mathrm{f}} \hat{\pi}_J = -\Omega^2 \hat{\Phi}_{\mathrm{f}}^T S \hat{\Phi}_{\mathrm{s}} \hat{u}_J. \tag{2.9.1–37}$$

If there is no acoustic force P_f in the coupled system of interest and no damping or boundary impedances in the acoustic fluid, this equation is simply the *J*th row of the acoustic part of the projected coupled harmonic forced response problem. Consequently, the modal acoustic contribution $\hat{\pi}_J$ due to mode *J* is simply equal to the *J*th modal coefficient of the solution to the coupled problem, $\hat{u}_J$, times the *J*th column of the pressure partition of the modal transformation, $\hat{\Phi}_{\mathrm{f}}$. Thus, no additional solution is required to obtain modal acoustic contribution factors when using coupled mode projections if acoustic forcing is absent. If acoustic forcing or damping is present in the coupled response problem defining $\hat{u}_J$, Equation 2.9.1–37 must be solved after the solution $\hat{u}_J$ is obtained.

When uncoupled modes are used in the projection for the solution of a coupled system, there is no direct relationship between acoustic and structural mode shapes. Therefore, application of the uncoupled modal transformations to the harmonic forced response problem does not produce the same trivial result as in the coupled mode case. The system resulting from the application of the separate uncoupled mode transformations Φ_{f} and Φ_{s} to Equation 2.9.1–36 must be solved for the modal coefficients $\tilde{\pi}_J$ corresponding to forcing via structural mode J:

$$\Phi_{\mathrm{f}}^T A_f \Phi_{\mathrm{f}} \tilde{\pi}_J = -\Omega^2 \Phi_{\mathrm{f}}^T S \Phi_{\mathrm{s}} \tilde{u}_J.$$

Impedance and admittance at fluid boundaries

Equation 2.9.1–11 (or alternatively Equation 2.9.1–9) can be written in a complex admittance form for steady-state analysis:

$$\mathbf{n}^- \cdot (\dot{\mathbf{u}}^m - \dot{\mathbf{u}}^f) = (\frac{1}{c_1} + \frac{i\Omega}{k_1})p = \frac{1}{Z(\Omega)}p = -T(\mathbf{x})(i\Omega)^{-1}, \tag{2.9.1–38}$$

where we define

$$\frac{1}{Z(\Omega)} \equiv \frac{1}{c_1} + \frac{i\Omega}{k_1}. \tag{2.9.1–39}$$

The term $1/Z(\Omega)$ is the complex admittance of the boundary, and $Z(\Omega)$ is the corresponding complex impedance. Thus, a required complex impedance or admittance value can be entered for a given frequency by fitting data to the parameters $1/c_1$ and $1/k_1$ using Equation 2.9.1–39.

For absorption of plane waves in an infinite medium with volumetric drag, the complex impedance can be shown to be

$$Z(\Omega) = \sqrt{K_f \tilde{\rho}} = \sqrt{K_f(\rho_f + \frac{\gamma}{i\Omega})}. \tag{2.9.1–40}$$

For the impedance-based nonreflective boundary condition in Abaqus/Standard, the equations above are used to determine the required constants $1/c_1$ and $1/k_1$. They are a function of frequency if the volumetric drag is nonzero. The small-drag versions of these equations are used in the direct time integration procedures, as in Equation 2.9.1–46.

Radiation boundary conditions

Many acoustic studies involve a vibrating structure in an infinite domain. In these cases we model a layer of the acoustic medium using finite elements, to a thickness of $1/3$ to a full wavelength, out to a "radiating" boundary surface. We then impose a condition on this surface to allow the acoustic waves to pass through and not reflect back into the computational domain. For radiation boundaries of simple shapes—such as planes, spheres, and the like—simple impedance boundary conditions can represent good approximations to the exact radiation conditions. In particular, we include local algebraic radiation conditions of the form

$$\mathbf{n}^- \cdot \frac{\partial p}{\partial \mathbf{x}} = Mp = f(i\tilde{k} + \beta)p, \tag{2.9.1–41}$$

where $\tilde{k} = \Omega\sqrt{\tilde{\rho}/K_f}$ is the wave number and $\tilde{\rho}$ is the complex density (see Equation 2.9.1–18). f is a geometric factor related to the metric factors of the curvilinear coordinate system used on the boundary, and β is a spreading loss term (see Table 2.9.1–1). Comparison of Equation 2.9.1–41 and

Table 2.9.1–1 Boundary condition parameters.

Geometry	f	β
Plane	*1*	*0*
Circle or circular cylinder	*1*	$\frac{1}{2r_1}$
Ellipse or elliptical cylinder	$\sqrt{\frac{1-\epsilon^2}{1-\epsilon^2[(\mathbf{x}_g-\mathbf{x}_0)\cdot\mathbf{e}_m/r_1]^2}}$	$\frac{1}{2r_1}$
Sphere	*1*	$\frac{1}{r_1}$
Prolate spheroid	$\sqrt{\frac{1-\epsilon^2}{1-\epsilon^2[(\mathbf{x}_g-\mathbf{x}_0)\cdot\mathbf{e}_m/r_1]^2}}$	$\frac{1}{r_1}$

Equation 2.9.1–9 reveals that, for steady-state analysis, there exists a direct analogy to the reactive boundary equation, Equation 2.9.1–21, with

$$\frac{1}{k_1} = \Im\left(\frac{f}{\Omega\sqrt{\tilde{\rho}K_f}}\right) - \frac{f\beta}{\Omega^2\rho_f\left(1+(\gamma/\Omega\rho_f)^2\right)}, \tag{2.9.1–42}$$

and

$$\frac{1}{c_1} = \Re\left(\frac{f}{\sqrt{\tilde{\rho}K_f}}\right) + \frac{f\beta r/\rho_f}{\Omega^2\rho_f\left(1+(\gamma/\Omega\rho_f)^2\right)}. \tag{2.9.1–43}$$

For transient procedures the treatment of volumetric drag in the acoustic equations and the radiation conditions necessitates an approximation. In the acoustics equation we use the boundary term

$$-\mathbf{n}^- \cdot \frac{\partial p}{\partial \mathbf{x}}\frac{1}{\rho_f} = \mathbf{n}^- \cdot \left(\ddot{\mathbf{u}}^f + \frac{\gamma}{\rho_f}\dot{\mathbf{u}}^f\right). \tag{2.9.1–44}$$

Combining Equation 2.9.1–41 with Equation 2.9.1–44, expanding about $\gamma = 0$, and retaining only first-order terms leads to

$$\mathbf{n}^- \cdot \frac{\partial p}{\partial \mathbf{x}}\frac{1}{\rho_f} = f\left[\frac{i\Omega}{\sqrt{\rho_f K_f}}\right]p + f\left[\frac{\beta}{\rho_f} + \left(\frac{\gamma}{2\rho_f\sqrt{\rho_f K_f}}\right)\right]p. \tag{2.9.1–45}$$

The Fourier inverse of the steady-state form results in the transient boundary condition

$$\mathbf{n}^- \cdot \frac{\partial p}{\partial \mathbf{x}}\frac{1}{\rho_f} = f\left[\frac{1}{\sqrt{\rho_f K_f}}\right]\dot{p} + f\left[\frac{\beta}{\rho_f} + \frac{\gamma}{2\rho_f\sqrt{\rho_f K_f}}\right]p. \tag{2.9.1–46}$$

This expression involves independent coefficients for pressure and its first derivative in time, unlike the transient reactive boundary expression (Equation 2.9.1–10), which includes independent coefficients for the first and second derivatives of pressure only. Consequently, to implement this expression, we define the admittance parameters

$$\frac{1}{c_1} = \left[\frac{f}{\sqrt{\rho_f K_f}}\right] \tag{2.9.1–47}$$

and

$$\frac{1}{a_1} = f\left[\frac{\beta}{\rho_f} + \frac{\gamma}{2\rho_f\sqrt{\rho_f K_f}}\right], \tag{2.9.1–48}$$

so the boundary traction for the transient radiation boundary condition can be written

$$\mathbf{n}^- \cdot \frac{\partial p}{\partial \mathbf{x}}\frac{1}{\rho_f} = \frac{1}{c_1}\dot{p} + \frac{1}{a_1}p.$$

The values of the parameters f and β vary with the geometry of the boundary of the radiating surface of the acoustic medium. The geometries supported in Abaqus are summarized in Table 2.9.1–1. In the table ϵ refers to the eccentricity of the ellipse or spheroid; r_1 refers to the radius of the circle, sphere, or the semimajor axis of the ellipse or spheroid; $\mathbf{x}_g$ is the vector locating the integration point on the ellipse or spheroid; $\mathbf{x}_0$ is the vector locating the center of the ellipse or spheroid; and $\mathbf{e}_m$ is the vector that orients the major axis.

These algebraic boundary conditions are approximations to the exact impedance of a boundary radiating into an infinite exterior. The plane wave condition is the exact impedance for plane waves normally incident to a planar boundary. The spherical condition exactly annihilates the first Legendre mode of a radiating spherical surface; the circular condition is asymptotically correct for the first mode (Bayliss et al., 1982). The elliptical and prolate spheroidal conditions are based on expansions of elliptical and prolate spheroidal wave functions in the low-eccentricity limit (Grote and Keller, 1995); the prolate spheroidal condition exactly annihilates the first term of its expansion, while the elliptical condition is asymptotic.

An improvement on radiation boundary conditions for plane waves

As already pointed out, the radiation boundary conditions derived in the previous section for plane waves are actually based on the presumption that the sound wave impinges on the boundary from an orthogonal direction. But this is not always the case. Figure 2.9.1–1 shows a general example for plane waves in which the sound wave direction differs from the boundary normal by an angle of θ. To consider this situation accurately, we adopt the plane-wave radiation equation used in Sandler (1998); i.e.,

$$\mathbf{n}^- \cdot \frac{\partial p}{\partial \mathbf{x}} = \frac{1}{C_n}\dot{p} = \frac{\cos\theta}{C}\dot{p}, \tag{2.9.1–49}$$

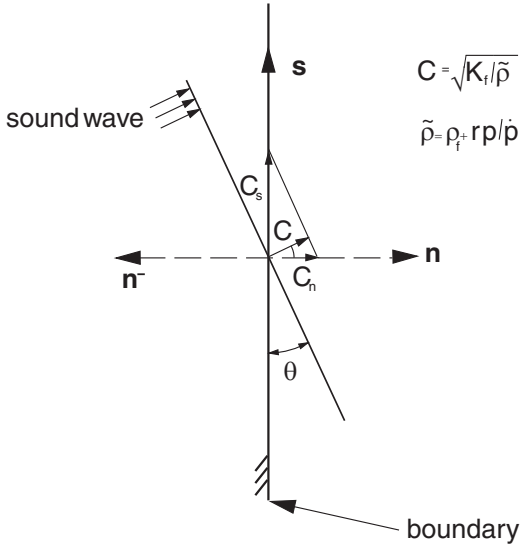


Figure 2.9.1–1 A plane wave not normally incident to the boundary.

where $C = \sqrt{K_f/\tilde{\rho}}$ is the sound speed with $\tilde{\rho} = \rho_f + \gamma p/\dot{p}$ and C_n is the corresponding speed normal to the boundary. This exact description of the geometry is the starting point for many developments of approximate absorbing boundary conditions (see, for example, Engquist and Majda, 1977). Thus, we have

$$\mathbf{n}^- \cdot \frac{\partial p}{\partial \mathbf{x}} \frac{1}{\rho_f} = \frac{\cos\theta}{\sqrt{\rho_f K_f}} \sqrt{1 + \frac{\gamma}{\rho_f} \frac{p}{\dot{p}}} \cdot \dot{p}. \tag{2.9.1–50}$$

Using the first-order expanding approximation to the second term in the square root in the above equation (similar to what we did to reach Equation 2.9.1–45), we can obtain an improved radiation boundary condition

$$\mathbf{n}^- \cdot \frac{\partial p}{\partial \mathbf{x}} \frac{1}{\rho_f} = \frac{\cos\theta}{\sqrt{\rho_f K_f}} \left(\dot{p} + \frac{\gamma}{2\rho_f} p \right). \tag{2.9.1–51}$$

It can be found from comparison that this equation differs from Equation 2.9.1–46 only by a factor of $\cos\theta$ for plane waves. In two dimensions the $\cos\theta$ can be calculated as

$$\cos\theta = |\frac{\partial p}{\partial \mathbf{n}}| / \sqrt{\left(\frac{\partial p}{\partial \mathbf{n}}\right)^2 + \left(\frac{\partial p}{\partial \mathbf{s}}\right)^2}. \tag{2.9.1–52}$$

The normal and tangential derivatives $\partial p/\partial \mathbf{n}$ and $\partial p/\partial \mathbf{s}$ at the integration points can be evaluated using the corresponding element along the radiation boundary surface (see Figure 2.9.1–2); i.e.,

$$\frac{\partial p}{\partial \mathbf{n}} = \mathbf{n} \cdot \frac{\partial H^P}{\partial \mathbf{x}} p^P \qquad \text{and} \qquad \frac{\partial p}{\partial \mathbf{s}} = \mathbf{s} \cdot \frac{\partial H^P}{\partial \mathbf{x}} p^P, \tag{2.9.1–53}$$

where p^P are the nodal pressure values of the element.

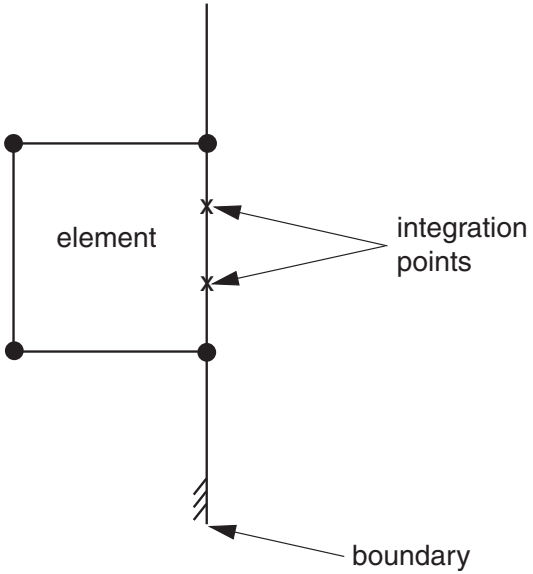


Figure 2.9.1–2 An element along the boundary.

The method described in this section can be used only for direct integration transient dynamics; it cannot be used with steady-state or modal response. In addition, it is available for planar, axisymmetric, and three-dimensional geometries.

Finally, the method makes the equilibrium equations nonlinear, as shown in Equation 2.9.1–52. Although in theory the iteration process in Abaqus/Standard can solve the nonlinear equilibrium equations accurately, the use of a small half-increment residual tolerance is strongly suggested since in many cases the pressure and its related residual along the radiation boundaries are very weak relative to the other places in the modeled domain. The computation of $\cos\theta$ at the integration point is based on the nodal pressures. The nodal pressures are updated using the explicit central difference procedure described in "Explicit dynamic analysis," Section 2.4.5.

References

- "Acoustic, shock, and coupled acoustic-structural analysis," Section 6.10.1 of the Abaqus Analysis User's Guide
- "Acoustic medium," Section 26.3.1 of the Abaqus Analysis User's Guide
- "Acoustic and shock loads," Section 34.4.6 of the Abaqus Analysis User's Guide

2.10 Piezoelectric analysis

- “Piezoelectric analysis,” Section 2.10.1

2.10.1 PIEZOELECTRIC ANALYSIS

Product: Abaqus/Standard

The piezoelectrical effect is the coupling of stress and electrical field in a material: an electrical field causes the material to strain, and vice versa. Abaqus/Standard has the capability to perform fully coupled piezoelectric analysis. The elements that are used in this case contain both displacement degrees of freedom and the electric potential as nodal variables.

Equilibrium and flux conservation

The piezoelectric effect is governed by coupled mechanical equilibrium and electric flux conservation equations.

The mechanical equilibrium equation is

$$\int_V \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon}\, dV = \int_S \mathbf{t} \cdot \delta\mathbf{u}\, dS + \int_V \mathbf{f} \cdot \delta\mathbf{u}\, dV, \tag{2.10.1–1}$$

where $\boldsymbol{\sigma}$ is the "true" (Cauchy) stress at a point currently at $\mathbf{x}$; $\mathbf{t}$ is the traction across a point of the surface of the body; $\mathbf{f}$ is the body force per unit volume in the body (such as the d'Alembert force $\mathbf{f} = -\rho\ddot{\mathbf{u}}$, in which ρ is the density of the body); and $\delta\boldsymbol{\varepsilon} \stackrel{\text{def}}{=} \text{sym}(\partial\delta\mathbf{u}/\partial\mathbf{x})$, where $\delta\mathbf{u}$ is an arbitrary, continuous vector field (the virtual velocity field).

The electrical flux conservation equation is

$$\int_V \mathbf{q} \cdot \delta\boldsymbol{E}\, dV = \int_S q_S \delta\varphi\, dS + \int_V q_V \delta\varphi\, dV, \tag{2.10.1–2}$$

where $\mathbf{q}$ is the electric flux vector; q_S is the electric flux per unit area entering the body at a point on its surface; q_V is the electric flux entering the body per unit volume; and $\delta\boldsymbol{E} \stackrel{\text{def}}{=} -\partial\delta\varphi/\partial\mathbf{x}$, where $\delta\varphi$ is an arbitrary, continuous, scalar field (the virtual potential). These quantities are also known by other terms that are frequently used within electrical engineering references. The electric flux vector $\boldsymbol{q}$ is known as the electrical displacement, and the potential gradient $\boldsymbol{E}$ is known as the electrical field.

Constitutive behavior: material coupling

Currently the assumption of linear materials is utilized. The basic equations for a piezoelectric linear medium are defined in the following. Following Ikeda (1990), three alternative forms of the constitutive equations are presented:

e-form:

$$\sigma_{ij} = D^E_{ijkl}\varepsilon_{kl} - e^{\varphi}_{mij}E_m,$$

$$q_i = e^{\varphi}_{ijk}\varepsilon_{jk} + D^{\varphi(\varepsilon)}_{ij}E_j,$$

d-form:

$$\varepsilon_{ij} = S^{E}_{ijkl}\sigma_{kl} + d^{\varphi}_{mij}E_m,$$

$$q_i = d^{\varphi}_{ijk}\sigma_{jk} + D^{\varphi(\sigma)}_{ij}E_j,$$

g-form:

$$\varepsilon_{ij} = S^{q}_{ijkl}\sigma_{kl} + g^{\varphi}_{mij}q_m,$$

$$E_i = -g^{\varphi}_{ijk}\sigma_{jk} + (D^{\varphi(\sigma)})^{-1}_{ij}q_j,$$

where S_{ijkl} are the material compliances; D_{ijkl} are the material stiffnesses; $e^{\varphi}_{ijk}, d^{\varphi}_{ijk}$, and g^{φ}_{ijk} are piezoelectric constants; and D^{φ}_{ij} are the dielectric constants. In these equations the superscript E, q, ε, or σ above a particular property indicates that the property is defined at zero electrical gradient, at zero electrical displacement, at zero strain, and at zero stress, respectively. Since all these forms describe the same constitutive relationships, the different mechanical and piezoelectrical constants can be expressed in terms of one another. The following relationships exist among the properties (Ikeda, 1990):

$$e^{\varphi}_{mij} = D^{E}_{ijkl}d^{\varphi}_{mkl}$$

$$d^{\varphi}_{mij} = D^{\varphi(\sigma)}_{mk}g^{\varphi}_{kij}$$

$$D^{\varphi(\sigma)}_{ij} - D^{\varphi(\varepsilon)}_{ij} = d^{\varphi}_{ikl}e^{\varphi}_{jkl}$$

$$D^{q}_{ijkl} - D^{E}_{ijkl} = e^{\varphi}_{mij}(D^{\varphi(\varepsilon)})^{-1}_{mn}e^{\varphi}_{nkl}$$

In Abaqus the constitutive equations in the *e*-form are used:

$$\sigma_{ij} = D^{E}_{ijkl}\varepsilon_{kl} - e^{\varphi}_{mkl}E_m,$$

$$q_i = e^{\varphi}_{ijk}\varepsilon_{jk} + D^{\varphi(\varepsilon)}_{ij}E_j. \qquad (2.10.1\text{–}3)$$

These are expressed in terms of the piezoelectric stress coefficient matrix e^{φ}_{mkl}. However, Abaqus also allows the input of piezoelectric constants in terms of the piezoelectric strain coefficient matrix d^{φ}_{mkl}.

The constitutive equations in the ***g***-form can also be expressed as

$$\sigma_{ij} = D^q_{ijkl}\varepsilon_{kl} - D^q_{ijkl}g^{\varphi}_{mkl}q_m$$

and

$$q_i = D^{\varphi(\sigma)}_{im}g^{\varphi}_{mjk}\sigma_{jk} + D^{\varphi(\sigma)}_{ij}E_j.$$

These equations can be convenient in interpreting and verifying the results of piezoelectrical analyses.

Kinematics

For the piezoelectric elements both displacements and electric potentials exist at the nodal locations. The displacements and electrical potentials are approximated within the element as

$$\mathbf{u} = \mathbf{N}^N u^N$$

and

$$\varphi = \mathbf{N}^N \varphi^N,$$

where $\mathbf{N}^N$ is the array of interpolating functions and u^N and φ^N are nodal quantities. The body forces and charges as well as the surface forces and charges are interpolated in a similar manner.

The strains and electrical potential gradients are given as

$$\boldsymbol{\varepsilon} = \mathbf{B}^N_u u^N$$

and

$$\mathbf{E} = -\mathbf{B}^N_{\varphi}\varphi^N,$$

where $\mathbf{B}^N_u$ and $\mathbf{B}^N_{\varphi}$ are the spatial derivatives of $\mathbf{N}^N$. In geometrically nonlinear analyses these spatial derivatives are defined in the current configuration.

System equations

With these approximate fields and the constitutive properties given above, in conjunction with the equilibrium and conservation equations, the following system of equations is derived in terms of nodal quantities:

$$M^{MN}\ddot{u}^N + K^{MN}_{uu}u^N + K^{MN}_{\varphi u}\varphi^N = P^M \qquad (2.10.1\text{–}4)$$

and

$$K_{\varphi u}^{MN} u^N - K_{\varphi\varphi}^{MN} \varphi^N = -Q^M, \tag{2.10.1–5}$$

where

$$M^{MN} = \int_V \rho \mathbf{N}^M \cdot \mathbf{N}^N \, dV$$

is the mass matrix (no inertia terms exist for the electrical flux conservation equation), ρ is the mass density,

$$K_{uu}^{MN} = \int_V \mathbf{B}_u^M : \mathbf{D}_m : \mathbf{B}_u^N \, dV$$

is the displacement stiffness matrix,

$$K_{\varphi\varphi}^{MN} = \int_V \mathbf{B}_\varphi^M \cdot \mathbf{D}_\varphi \cdot \mathbf{B}_\varphi^N \, dV$$

is the dielectric "stiffness" matrix,

$$K_{\varphi u}^{MN} = \int_V \mathbf{B}_\varphi^M \cdot \mathbf{e} : \mathbf{B}_u^N \, dV$$

is the piezoelectric coupling matrix,

$$P^M = \int_V \mathbf{N}^M \cdot \mathbf{P}_v \, dV + \int_S \mathbf{N}^M \cdot \mathbf{P}_s \, dS + P_c^M$$

is the mechanical force vector, and

$$Q^M = \int_V \mathbf{N}^M \cdot \mathbf{Q}_v \, dV + \int_S \mathbf{N}^M \cdot \mathbf{Q}_s \, dS + Q_c^M$$

is the electrical charge vector. In these expressions the constitutive properties are specified in a matrix form where $\mathbf{D}_m$ is the mechanical relationship, $\mathbf{D}_\varphi$ is the electrical relationship, and $\mathbf{e}$ is the piezoelectrical relationship. The "load" vectors include the body, surface, and concentrated quantities, as shown. The unknowns are the nodal displacements and potentials. Once these are determined, the strains and potential gradients can be computed using the expressions given above. The stresses and electrical flux densities are computed by means of constitutive relationships.

Reference

- "Piezoelectric analysis," Section 6.7.2 of the Abaqus Analysis User's Guide

2.11 Heat transfer

- “Uncoupled heat transfer analysis,” Section 2.11.1
- “Shell heat conduction,” Section 2.11.2
- “Convection/diffusion,” Section 2.11.3
- “Cavity radiation,” Section 2.11.4
- “View factor calculation,” Section 2.11.5

2.11.1 UNCOUPLED HEAT TRANSFER ANALYSIS

Product: Abaqus/Standard

The Abaqus/Standard capability for uncoupled heat transfer analysis is intended to model solid body heat conduction with general, temperature-dependent conductivity; internal energy (including latent heat effects); and quite general convection and radiation boundary conditions. This section describes the basic energy balance, constitutive models, boundary conditions, finite element discretization, and time integration procedures used.

Heat transfer in flowing materials (convection) is discussed in "Convection/diffusion," Section 2.11.3. Radiation heat transfer in cavities is discussed in "Cavity radiation," Section 2.11.4. All such heat transfer mechanisms can be present in a model.

Energy balance

The basic energy balance is (Green and Naghdi)

$$\int_V \rho \dot{U}\, dV = \int_S q\, dS + \int_V r\, dV, \tag{2.11.1–1}$$

where V is a volume of solid material, with surface area S; ρ is the density of the material; $\dot{U}$ is the material time rate of the internal energy; q is the heat flux per unit area of the body, flowing into the body; and r is the heat supplied internally into the body per unit volume.

It is assumed that the thermal and mechanical problems are uncoupled in the sense that $U = U(\theta)$ only, where θ is the temperature of the material, and q and r do not depend on the strains or displacements of the body. For simplicity a Lagrangian description is assumed, so "volume" and "surface" mean the volume and surface in the reference configuration.

Constitutive definition

This relationship is usually written in terms of a specific heat, neglecting coupling between mechanical and thermal problems:

$$c(\theta) = \frac{dU}{d\theta},$$

except for latent heat effects at phase changes, which are given separately in terms of solidus and liquidus temperatures (the lower and upper temperature bounds of the phase change range) and the total internal energy associated with the phase change, called the latent heat. When latent heat is given, it is assumed to be in addition to the specific heat effect (see Figure 2.11.1–1). For many cases it is reasonable to assume that the phase change occurs within a known temperature range, which can be specified by the user. However, in some cases it may be necessary to include a kinetic theory for the phase change to model the effect accurately (an example would be the prediction of crystallization in a polymer casting process).

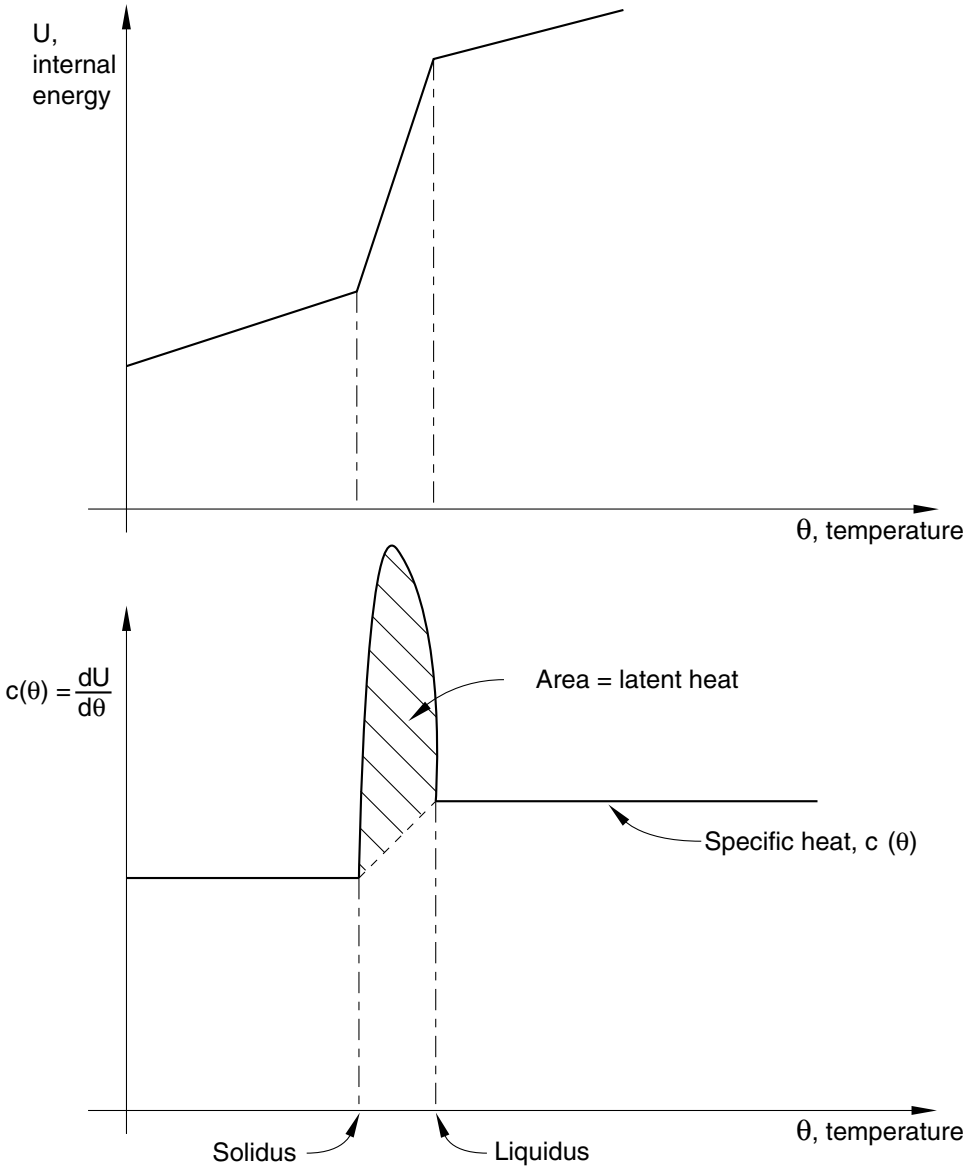


Figure 2.11.1–1 Specific heat, latent heat definition.

For such cases the user can model the process in considerable detail, using the solution-dependent state variable feature in Abaqus, with user subroutine **HETVAL**.

Heat conduction is assumed to be governed by the Fourier law,

$$\mathbf{f} = -\mathbf{k}\frac{\partial\theta}{\partial\mathbf{x}}, \tag{2.11.1–2}$$

where $\mathbf{k}$ is the conductivity matrix, $\mathbf{k} = \mathbf{k}(\theta)$; $\mathbf{f}$ is the heat flux; and $\mathbf{x}$ is position. The conductivity $\mathbf{k}$ can be fully anisotropic, orthotropic, or isotropic.

Boundary conditions

Boundary conditions can be specified as prescribed temperature, $\theta = \theta(\mathbf{x}, t)$; prescribed surface heat flux, $q = q(\mathbf{x}, t)$ per area; prescribed volumetric heat flux, $q = r(\mathbf{x}, t)$ per volume; surface convection: $q = h(\theta - \theta^0)$, where $h = h(\mathbf{x}, t)$ is the film coefficient and $\theta^0 = \theta^0(\mathbf{x}, t)$ is the sink temperature; and radiation: $q = A\Big((\theta - \theta^Z)^4 - (\theta^0 - \theta^Z)^4\Big)$, where A is the radiation constant (emissivity times the Stefan-Boltzmann constant) and θ^Z is the value of absolute zero on the temperature scale being used. Surfaces can also participate in cavity radiation effects. The cavity radiation formulation in Abaqus is described in "Cavity radiation," Section 2.11.4.

Spatial discretization

A variational statement of the energy balance, Equation 2.11.1–1, together with the Fourier law, Equation 2.11.1–2, is obtained directly by the standard Galerkin approach as

$$\int_V \rho\dot{U}\delta\theta\, dV + \int_V \frac{\partial\delta\theta}{\partial\mathbf{x}} \cdot \mathbf{k} \cdot \frac{\partial\theta}{\partial\mathbf{x}} dV = \int_V \delta\theta\, r\, dV + \int_{s_q} \delta\theta\, q\, dS, \tag{2.11.1–3}$$

where $\delta\theta$ is an arbitrary variational field satisfying the essential boundary conditions. The body is approximated geometrically with finite elements, so the temperature is interpolated as

$$\theta = N^N(\mathbf{x})\,\theta^N, \quad N = 1, 2, \ldots,$$

where θ^N are nodal temperatures. The Galerkin approach assumes that $\delta\theta$, the variational field, is interpolated by the same functions:

$$\delta\theta = N^N \delta\theta^N.$$

First- and second-order polynomials in one, two, and three dimensions are used for the N^N. With these interpolations the variational statement, Equation 2.11.1–3, becomes

$$\delta\theta^N \left\{ \int_V N^N \rho\dot{U} dV + \int_V \frac{\partial N^N}{\partial\mathbf{x}} \cdot \mathbf{k} \cdot \frac{\partial\theta}{\partial\mathbf{x}} dV = \int_V N^N r\, dV + \int_{S_q} N^N q\, dS \right\},$$

and since the $\delta\theta^N$ are arbitrarily chosen, this gives the system of equations

$$\int_V N^N \rho \dot{U} dV + \int_V \frac{\partial N^N}{\partial \mathbf{x}} \cdot \mathbf{k} \cdot \frac{\partial \theta}{\partial \mathbf{x}} dV = \int_V N^N r \, dV + \int_{S_q} N^N q \, dS. \tag{2.11.1–4}$$

This set of equations is the "continuous time description" of the geometric approximation.

Time integration

Abaqus/Standard uses the backward difference algorithm:

$$\dot{U}_{t+\Delta t} = (U_{t+\Delta t} - U_t)(1/\Delta t). \tag{2.11.1–5}$$

This operator is chosen for a number of reasons. First of all, we choose from one-step operators of the form

$$f_{t+\Delta t} = f_t + \left((1-\gamma)\dot{f}_t + \gamma \dot{f}_{t+\Delta t}\right)\Delta t$$

because of their simplicity in implementation (for example, no special starting procedures are needed) and well-understood behavior. For $\gamma < 1/2$ such operators are only conditionally stable for linear heat transfer problems. We prefer to work with unconditionally stable methods, because Abaqus is most commonly applied to problems where the solution is sought over very long time periods (compared to the stability limit for the explicit form of the operator, $\gamma = 0$), and so choose $\gamma \geq 1/2$. Of these operators the central difference method, $\gamma = 1/2$, has the highest accuracy. However, that form of the operator tends to produce oscillations in the early time solution that are not present in the backward difference form. Thus, we use $\gamma = 1$: backward difference. Introducing the operator, Equation 2.11.1–5, into the energy balance Equation 2.11.1–4 gives

$$\frac{1}{\Delta t}\int_V N^N \rho (U_{t+\Delta t} - U_t) dV + \int_V \frac{\partial N^N}{\partial \mathbf{x}} \cdot \mathbf{k} \cdot \frac{\partial \theta}{\partial \mathbf{x}} dV - \int_V N^N r \, dV - \int_{S_q} N^N q \, dS = 0. \tag{2.11.1–6}$$

This nonlinear system is solved by a modified Newton method. The method is modified Newton because the tangent matrix (the Jacobian matrix)—that is, the rate of change of the left-hand side of Equation 2.11.1–6 with respect to $\theta^N_{t+\Delta t}$—is not formed exactly. The formation of the terms in this tangent matrix is now described.

The internal energy term gives a Jacobian contribution:

$$\frac{1}{\Delta t}\int_V N^N \rho \frac{dU}{d\theta}\bigg|_{t+\Delta t} N^M dV.$$

$(dU/d\theta)|_{t+\Delta t}$ is the specific heat, $c(\theta)$, outside the latent heat range, and is $c + L/(\theta_L - \theta_S)$ if $\theta_L > \theta_{t+\Delta t} > \theta_S$ at the integration point, where θ_L and θ_S are the liquidus and solidus temperatures and $\boldsymbol{L}$ is the latent heat associated with this phase change.

In severe latent heat cases this term can result in numerical instabilities, as the stiffness term $dU/d\theta$ is small outside the solidus-liquidus temperature range and is very stiff inside that rather narrow range. To avoid such instabilities in those cases this term is modified to a secant term during the early iterations of the solution to a time step. Since the modification occurs only in cases involving latent heat, it affects only those problems.

The conductivity term gives a Jacobian contribution:

$$\int_V \frac{\partial N^N}{\partial \mathbf{x}} \cdot \mathbf{k}\Big|_{t+\Delta t} \cdot \frac{\partial N^M}{\partial \mathbf{x}} dV + \int_V \frac{\partial N^N}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{k}}{\partial \theta}\Big|_{t+\Delta t} \cdot \frac{\partial \theta}{\partial \mathbf{x}}\Big|_{t+\Delta t} N^M dV.$$

The second of these terms is typically small, since the conductivity usually varies only slowly with temperature. Because of this, and because the term is not symmetric, it is usually more efficient to omit it. This term is omitted unless the unsymmetric solver is chosen. Prescribed surface fluxes and body fluxes can also be temperature dependent and will then give rise to Jacobian contributions.

With film and radiation conditions, the surface flux term gives a Jacobian contribution:

$$\int_S N^N \frac{\partial q}{\partial \theta}\Big|_{t+\Delta t} N^M dS.$$

For film conditions, $q = h(\theta)(\theta - \theta^o)$,

$$\frac{\partial q}{\partial \theta} = \frac{\partial h}{\partial \theta}(\theta - \theta^o) + h,$$

while for radiation, $q = A(\theta^4 - \theta^{o4})$,

$$\frac{\partial q}{\partial \theta} = 4A\theta^3.$$

These terms are included in exactly this form in the Jacobian. The modified Newton method is then

$$\begin{aligned}
\left[\frac{1}{\Delta t}\int_V N^N \rho \frac{dU}{d\theta}\Big|_{t+\Delta t} N^M dV + \int_V \frac{\partial N^N}{\partial \mathbf{x}} \cdot \mathbf{k}\Big|_{t+\Delta t} \cdot \frac{\partial N^M}{\partial \mathbf{x}}\, dV \right.& \\
\left. + \int_S N^N \left(\frac{\partial h}{\partial \theta}(\theta - \theta^o) + h + 4A\theta^3\right) N^M dS\right] \overline{c}^M & \\
= \int_V N^N r\, dV + \int_{S_q} N^N q\, dS - \frac{1}{\Delta t}\int_V N^N \rho (U_{t+\Delta t} - U_t)\, dV & \\
- \int_V \frac{\partial N^N}{\partial \mathbf{x}} \cdot \mathbf{k} \cdot \frac{\partial \theta}{\partial \mathbf{x}}\, dV, & \\
\text{with } \theta^N_{t+\Delta t, i+1} = \theta^N_{t+\Delta t, i} + \overline{c}^N, i = \text{ iteration number.} &
\end{aligned} \tag{2.11.1–7}$$

For purely linear systems Equation 2.11.1–7 is linear in $\overline{c}^M$ and, hence, in $\theta^N_{t+\Delta t}$, so a single equation solution provides the $\theta^N_{t+\Delta t}$. Since the method usually is only a minor modification of Newton's method, convergence is rapid.

Abaqus/Standard uses an automatic (self-adaptive) time stepping algorithm to choose Δt. This is based on a user-supplied tolerance on the maximum temperature change allowed in a time increment, and the increment is adjusted according to this parameter, as well as the convergence rate of Equation 2.11.1–7 in nonlinear cases.

The first-order heat transfer elements (such as 2-node link, 4-node quadrilateral, and 8-node brick) use a numerical integration rule with the integration stations located at the corners of the element for the heat capacitance terms. This means that the Jacobian term associated with the internal energy rate is diagonal. This approach is especially effective when strong latent heat effects are present. The second-order elements use conventional Gaussian integration. Thus, second-order elements are to be preferred for problems when the solution will be smooth (without latent heat effects), whereas the first-order elements should be used in nonsmooth cases (with latent heat).

The HEATCAP element is available for modeling lumped heat capacitance at a point. The associated concentrated film and concentrated radiation loading options are specified by the user. These loading options are also allowed in coupled temperature-displacement, coupled thermal-electrical, and coupled thermal-electrical-structural analyses.

Reference

- "Uncoupled heat transfer analysis," Section 6.5.2 of the Abaqus Analysis User's Guide

2.11.2 SHELL HEAT CONDUCTION

Product: Abaqus/Standard

This section describes the formulation used in the shell heat conduction elements in Abaqus/Standard. The basis of the elements is a combination of piecewise quadratic interpolation of temperature through the thickness of the shell and either linear interpolation (in elements DS3 and DS4) or quadratic interpolation (in elements DS6 and DS8) on the reference surface of the shell. The isoparametric interpolation functions for the shell reference surface are identical in form to those used for the solid quadrilateral and triangular elements and can be found in "Solid isoparametric quadrilaterals and hexahedra," Section 3.2.4, and "Triangular, tetrahedral, and wedge elements," Section 3.2.6, respectively. Nodal temperature values are stored at a set of points through the thickness (points P below) at each node of the element (nodes N below). For the purpose of numerical integration of the finite element equations, a 2 × 2 Gauss integration scheme with a 2 × 2 nodal integration scheme for the internal energy and specific heat term is used for the quadrilateral element DS4 and a 3 × 3 Gauss integration scheme is used for the quadrilateral element DS8. Three- and six-point integration schemes are used for the triangular elements DS3 and DS6, respectively, the details of which can be found in "Triangular, tetrahedral, and wedge elements," Section 3.2.6.

Let (ϕ_1, ϕ_2) be material coordinates of a point in the reference surface of the shell, and let s_3 measure position through the thickness of the shell so that $-h/2 - z_0 \le s_3 \le h/2 - z0$, where h is the thickness of the shell, z_0 is the offset of the reference surface from the midsurface as discussed in "Transverse shear stiffness in composite shells and offsets from the midsurface," Section 3.6.8. The position of any point in the shell is given by

$$\mathbf{x} = \mathbf{x}^o(\phi_1, \phi_2) + s_3\, \mathbf{n}(\phi_1, \phi_2),$$

where

$\mathbf{x}^o$ is the position of a point in the reference surface, and

$\mathbf{n}$ is the unit normal to the reference surface of the shell.

The temperature interpolation can be written as

$$\theta = N^N(\phi_1, \phi_2) M^P(s_3) \overline{\theta}^{NP}, \tag{2.11.2–1}$$

where

$M^P(s_3)$ is a piecewise parabolic interpolation,

$N^N(\phi_1, \phi_2)$ is an interpolator in the reference surface, and

$\overline{\theta}^{NP}$ are nodal temperature values (at node N, point P through the thickness).

The basic heat energy balance is Equation 2.11.1–3, with the approximate Jacobian matrix for the Newton method based on

$$\int_V \rho\delta\theta \frac{d\dot{U}}{d\theta_{t+\Delta t}} d\theta \, dV + \int_V \frac{\partial \delta\theta}{\partial \mathbf{x}} \cdot \mathbf{k} \cdot \frac{\partial d\theta}{\partial \mathbf{x}} \, dV - \int_S \delta\theta \frac{\partial q}{\partial \theta_{t+\Delta t}} d\theta \, dS,$$

where $d\theta$ is the correction to the temperature solution at time $t + \Delta t$. The derivation of this form is discussed in "Uncoupled heat transfer analysis," Section 2.11.1. The form of these terms for shell heat conduction elements is obtained by introducing the interpolator, Equation 2.11.2–1, and neglecting the change in area, with respect to s_3, of surfaces parallel to the reference surface.

The internal energy rate term (the first term in Equation 2.11.1–3) contributes, to the residual,

$$\int_A \left[N^N \int_{-h/2}^{h/2} M^P \rho \dot{U}_{t+\Delta t} ds_3 \right] dA,$$

and to the Jacobian,

$$\int_A \left[N^N \int_{-h/2}^{h/2} M^P \rho \frac{d\dot{U}}{d\theta_{t+\Delta t}} M^Q ds_3 \right] N^M dA.$$

For the second term the temperature derivatives are taken with respect to a local orthogonal system (s_1, s_2, s_3), where s_1 and s_2 measure distance along local base vectors $\mathbf{e}_1$ and $\mathbf{e}_2$, in the reference surface of the shell, set up according to the standard convention in Abaqus for such local systems in shells. The term is formed by first introducing an intermediate set of temperature values,

$$\left[\theta^P, \frac{\partial \theta^P}{\partial s_1}, \frac{\partial \theta^P}{\partial s_2} \right],$$

corresponding to each temperature value point through the thickness, at each section where integration through the thickness is performed. Since the number of temperature values on the section is the same as the number of integration points, M^P is unity in the appropriate locations and zero everywhere else. Then we can interpolate to the section by

$$\left\{ \begin{array}{c} \theta^P \\ \dfrac{\partial \theta^P}{\partial s_1} \\ \dfrac{\partial \theta^P}{\partial s_2} \end{array} \right\} = \left\{ \beta^M \right\} \overline{\theta}^{MP},$$

where

$$\left\{ \beta^M \right\} = \left\{ \begin{array}{c} N^M \\ \dfrac{\partial N^M}{\partial s_1} \\ \dfrac{\partial N^M}{\partial s_2} \end{array} \right\}.$$

The piecewise quadratic interpolation through the thickness then gives

$$\left\{ \begin{array}{c} \dfrac{\partial\theta}{\partial s_1} \\ \dfrac{\partial\theta}{\partial s_2} \\ \dfrac{\partial\theta}{\partial s_3} \end{array} \right\} = \left[\gamma^P \right] \left\{ \begin{array}{c} \theta^P \\ \dfrac{\partial\theta^P}{\partial s_1} \\ \dfrac{\partial\theta^P}{\partial s_2} \end{array} \right\},$$

where

$$\left[\gamma^P \right] = \left[\begin{array}{ccc} 0 & M^P & 0 \\ 0 & 0 & M^P \\ dM^P/ds_3 & 0 & 0 \end{array} \right].$$

The conductivity term in the Jacobian then is

$$\int_A \left\lfloor \beta^N \right\rfloor \left[\int_{-h/2}^{h/2} \left[\gamma^P \right]^T \left[k \right] \left[\gamma^Q \right] ds_3 \right] \left\{ \beta^M \right\} dA,$$

where $[\,k\,]$ is the local conductivity matrix and the same term multiplied by $\theta^{MQ}_{t+\Delta t}$ appears in the residual.

Finally, the external flux terms contribute

$$\int_A N^N \int_{-h/2}^{h/2} M^P r \, ds_3 dA + \int_A N^N \left[I^A q^A + I^B q^B \right] dA$$

to the residual and

$$\int_A N^N \left[I^A \frac{dq}{d\theta}\bigg|_{t+\Delta t, A} + I^B \frac{dq}{d\theta}\bigg|_{t+\Delta t, B} \right] N^M dA$$

to the Jacobian, where

$I^A = 1$ at point A through the thickness

$I^A = 0$ at all other points through the thickness,

and points A and B are on the top and bottom surfaces of the shell.

Reference

- "Three-dimensional conventional shell element library," Section 29.6.7 of the Abaqus Analysis User's Guide

2.11.3 CONVECTION/DIFFUSION

Product: Abaqus/Standard

The formulation in this section describes a capability for modeling heat transfer with convection in Abaqus/Standard. The resulting elements can be used in any general heat transfer mesh. These elements have a nonsymmetric Jacobian matrix: the nonsymmetric capability is invoked automatically if elements of this type are included in the model. Both steady-state and transient capabilities are provided. The transient capability introduces a limit on the time increment (the limit is defined below): the time increment is adjusted to satisfy this limit if necessary. The steady-state versions of the elements can be used in a transient analysis, which means that transient effects in the fluid are not included in the model. The formulation is based on the work of Yu and Heinrich (1986, 1987).

Thermal equilibrium equation

The thermal equilibrium equation for a continuum in which a fluid is flowing with velocity $\mathbf{v}$, is

$$\int \delta\theta \left[\rho c \left\{ \frac{\partial \theta}{\partial t} + \mathbf{v} \cdot \frac{\partial \theta}{\partial \mathbf{x}} \right\} - \frac{\partial}{\partial \mathbf{x}} \cdot \left(\mathbf{k} \cdot \frac{\partial \theta}{\partial \mathbf{x}} \right) - q \right] dV + \int_{S_q} \delta\theta \left[\mathbf{n} \cdot \mathbf{k} \cdot \frac{\partial \theta}{\partial \mathbf{x}} - q_s \right] dS = 0,$$

where $\theta(\mathbf{x}, t)$ is the temperature at a point, $\delta\theta(\mathbf{x}, t)$ is an arbitrary variational field, $\rho(\theta)$ is the fluid density, $c(\theta)$ is the specific heat of the fluid, $\mathbf{k}(\theta)$ is the conductivity of the fluid, q is the heat added per unit volume from external sources, q_s is the heat flowing into the volume across the surface on which temperature is not prescribed (S_q), $\mathbf{n}$ is the outward normal to the surface, $\mathbf{x}$ is spatial position, and t is time. Although most fluids will have isotropic conductivity, so that $\mathbf{k} = k\,\mathbf{I}$ (where $k(\theta)$ is a scalar and $\mathbf{I}$ is a unit matrix), we provide for anisotropic conductivity to cover such cases as that of fluid flowing through a set of baffle plates whose conductivity is smeared into that of the fluid.

The boundary conditions are that $\theta(\mathbf{x})$ is prescribed over some part of the surface, S_θ, and that the heat flux per unit area entering the domain across the rest of the surface, $q_s(\mathbf{x})$, is prescribed or is defined by convection and/or radiation conditions. For example, the boundary layer between fluid convection elements and solid elements might be modeled by DINTERx-type elements. The boundary term in the thermal equilibrium equation defines

$$q_s = -\mathbf{n} \cdot \mathbf{k} \cdot \frac{\partial \theta}{\partial \mathbf{x}}.$$

This implies that q_s is the flux associated with conduction across the surface only—any convection of energy across the surface is not included in q_s. This makes no difference if the surface is part of a solid body (where q_s would be defined by heat transfer into the adjacent body), since then the normal velocity into that body, $\mathbf{v} \cdot \mathbf{n}$, is zero. But it does make a difference when there is fluid crossing the surface, as—for example—on the upstream and downstream boundaries of the mesh. In this case the choice of q_s for the natural boundary condition (instead of using the total flux crossing the surface) is desirable because it avoids spurious reflections of energy back into the mesh as the fluid flows through the surface.

These equations are discretized with respect to position by using first-order isoparametric elements. The fluid velocity, $\mathbf{v}$, is assumed to be known. (Abaqus actually requires that the mass flow rate of the fluid per unit area be defined, because this is generally more convenient for the user. The velocity is computed from the mass flow rate and the density of the fluid.)

The time discretization generates the solution at time $t + \Delta t$ from the known solution at time t.

The interpolation for the temperature, θ, is defined over an element and over a time increment as

$$\theta(\mathbf{x}, t) = N^N(\mathbf{x}) A^n(t) \theta^{(N,n)}, \quad \text{for} \quad N = 1, 2, \ldots, \quad n = t, t + \Delta t,$$

where the N^N are standard isoparametric functions and the time interpolation, A^n, is linear:

$$A^t = 1 - \frac{\tau}{\Delta t}, \quad A^{t+\Delta t} = \frac{\tau}{\Delta t},$$

where Δt is the time increment and $0 \le \tau \le \Delta t$.

The Petrov-Galerkin discretization proposed by Yu and Heinrich couples this linear interpolation with the weighting functions

$$\begin{aligned} \delta\theta &= W^N \delta\theta^N \\ &= \left[N^N \bar{A} + \frac{h}{2} \left(\alpha \bar{A} + \beta \frac{\Delta t}{2} \frac{d\bar{A}}{dt} \right) \frac{\mathbf{v}}{|\mathbf{v}|} \cdot \frac{\partial N^N}{\partial \mathbf{x}} \right] \delta\theta^N, \end{aligned}$$

where

$$\bar{A} = 6 \frac{\tau}{\Delta t^2} \left(1 - \frac{\tau}{\Delta t} \right);$$

$\mathbf{v}$ is the average fluid velocity over the element; $\mathbf{v}|$ is its magnitude; and h is a characteristic element length measure, defined below. α and β are control parameters. The α term in the weighting is introduced to eliminate artificial diffusion of the solution, while the β term is introduced to avoid numerical dispersion. Yu and Heinrich show that the optimal choices are

$$\alpha = \coth \frac{\gamma}{2} - \frac{2}{\gamma} \quad \text{and} \quad \beta = \frac{C}{3} - \frac{2}{\gamma} \frac{\alpha}{C},$$

where γ is the local Péclet number in an element and C is the local Courant number, defined as

$$\gamma = |\mathbf{v}| h \frac{\rho c}{k} \quad \text{and} \quad C = |\mathbf{v}| \frac{\Delta t}{h}.$$

The above expression for β yields negative values for very small fluid velocities, which may destabilize the solution; hence, for low velocities dispersion control is switched off.

The characteristic element length measure, h, is defined by Yu and Heinrich as follows.

Let $\mathbf{h}_\alpha$ be the α isoparametric line across the element passing through its centroid. The projection of $\mathbf{h}_\alpha$ in the direction of the fluid velocity vector at the element's centroid is

$$h_\alpha = \mathbf{h}_\alpha \cdot \frac{\mathbf{v}}{|\mathbf{v}|}.$$

Then we define $\boldsymbol{h}$ as

$$h = \sum_\alpha |h_\alpha|.$$

When β is nonzero, these elements require that $C \leq 1$ for numerical stability.

Since the weighting functions are biased ("upwinding"), they are discontinuous from one element to the next. Some care is, therefore, required in manipulating the weak form of the thermal equilibrium equation (see Hughes and Brooks, 1982). In particular, the usual integration by parts of the conduction term

$$\delta\theta \frac{\partial}{\partial \mathbf{x}} \cdot \left(\mathbf{k} \cdot \frac{\partial \theta}{\partial \mathbf{x}} \right)$$

can be performed only for the continuous part of the weighting functions used to discretize $\delta\theta$: otherwise, continuity of heat flux between elements is not assured. For convenience we write the discontinuous part of the weighting as

$$P^N = \frac{h}{2} \left(\alpha \bar{A} + \beta \frac{\Delta t}{2} \frac{d\bar{A}}{dt} \right) \frac{\mathbf{v}}{|\mathbf{v}|} \cdot \frac{\partial N^N}{\partial \mathbf{x}}.$$

The weak form of thermal equilibrium is

$$\int \left[W^N \rho c \left\{ N^M \frac{dA^n}{dt} + \mathbf{v} \cdot \frac{\partial N^M}{\partial \mathbf{x}} A^n \right\} + \bar{A} \frac{\partial N^N}{\partial \mathbf{x}} \cdot \mathbf{k} \cdot \frac{\partial N^M}{\partial \mathbf{x}} A^n - P^N \mathbf{k} : \frac{\partial^2 N^M}{\partial \mathbf{x} \partial \mathbf{x}} A^n \right] dV$$
$$- \int W^N q \, dV - \bar{A} \int_S N^N q_s \, dS = 0.$$

This can be rewritten as

$$\begin{aligned}
&\bar{A} \frac{dA^n}{dt} \int \rho c \left(N^N + \alpha \frac{h}{2} \frac{\mathbf{v}}{|\mathbf{v}|} \cdot \frac{\partial N^N}{\partial \mathbf{x}} \right) N^M \, dV \\
&+ \bar{A} A^n \int \rho c \left(N^N + \alpha \frac{h}{2} \frac{\mathbf{v}}{|\mathbf{v}|} \cdot \frac{\partial N^N}{\partial \mathbf{x}} \right) \mathbf{v} \cdot \frac{\partial N^M}{\partial \mathbf{x}} \, dV \\
&+ \beta \frac{h \Delta t}{4} \frac{d\bar{A}}{dt} \left(\frac{dA^n}{dt} \int \rho c \frac{\mathbf{v}}{|\mathbf{v}|} \cdot \frac{\partial N^N}{\partial \mathbf{x}} N^M \, dV + A^n \int \rho c \frac{\mathbf{v}}{|\mathbf{v}|} \cdot \frac{\partial N^N}{\partial \mathbf{x}} \mathbf{v} \cdot \frac{\partial N^M}{\partial \mathbf{x}} \, dV \right)
\end{aligned}$$

$$
\begin{aligned}
&+ \bar{A}A^n \int \left(\frac{\partial N^N}{\partial \mathbf{x}} \cdot \mathbf{k} \cdot \frac{\partial N^M}{\partial \mathbf{x}} - \alpha \frac{h}{2} \frac{\mathbf{v}}{|\mathbf{v}|} \cdot \frac{\partial N^N}{\partial \mathbf{x}} \, \mathbf{k} : \frac{\partial^2 N^M}{\partial \mathbf{x} \partial \mathbf{x}} \right) dV \\
&- \beta \frac{h \Delta t}{4} \frac{d\bar{A}}{dt} A^n \int \frac{\mathbf{v}}{|\mathbf{v}|} \cdot \frac{\partial N^N}{\partial \mathbf{x}} \, \mathbf{k} : \frac{\partial^2 N^M}{\partial \mathbf{x} \partial \mathbf{x}} \, dV \\
&- \bar{A} \int_V \left[N^N + \alpha \frac{h}{2} \frac{\mathbf{v}}{|\mathbf{v}|} \cdot \frac{\partial N^N}{\partial \mathbf{x}} \right] q \, dV - \frac{d\bar{A}}{dt} \int_V \beta \frac{\Delta t}{2} \frac{\mathbf{v}}{|\mathbf{v}|} \cdot \frac{\partial N^N}{\partial \mathbf{x}} q \, dV - \bar{A} \int_S N^N q_s \, dS = 0.
\end{aligned}
$$

We now integrate this equation from time t to $t + \Delta t$ to provide an average equilibrium statement for the increment. We use the results

$$
\int_{\Delta t} \bar{A} \, dt = 1, \quad \int_{\Delta t} \frac{d\bar{A}}{dt} \, dt = 0,
$$

$$
\int_{\Delta t} \bar{A} \frac{dA^{t+\Delta t}}{dt} \, dt = - \int_{\Delta t} \bar{A} \frac{dA^t}{dt} \, dt = \frac{1}{\Delta t}, \quad \int_{\Delta t} \bar{A} A^{t+\Delta t} \, dt = \int_{\Delta t} \bar{A} A^t \, dt = \frac{1}{2},
$$

and

$$
\int_{\Delta t} \frac{d\bar{A}}{dt} \frac{dA^{t+\Delta t}}{dt} \, dt = \int_{\Delta t} \frac{d\bar{A}}{dt} \frac{dA^t}{dt} \, dt = 0, \quad \int_{\Delta t} \frac{d\bar{A}}{dt} A^{t+\Delta t} \, dt = - \int_{\Delta t} \frac{d\bar{A}}{dt} A^t \, dt = - \frac{1}{\Delta t},
$$

to give

$$
\begin{aligned}
&\frac{1}{\Delta t} \int \rho c \left(N^N + \alpha \frac{h}{2} \frac{\mathbf{v}}{|\mathbf{v}|} \cdot \frac{\partial N^N}{\partial \mathbf{x}} \right) N^M \, dV \left(\theta^{M,t+\Delta t} - \theta^{M,t} \right) \\
&+ \frac{1}{2} \int \rho c \left(N^N + \alpha \frac{h}{2} \frac{\mathbf{v}}{|\mathbf{v}|} \cdot \frac{\partial N^N}{\partial \mathbf{x}} \right) \mathbf{v} \cdot \frac{\partial N^M}{\partial \mathbf{x}} \, dV \left(\theta^{M,t+\Delta t} + \theta^{M,t} \right) \\
&- \beta \frac{h}{4} \int \rho c \frac{\mathbf{v}}{|\mathbf{v}|} \cdot \frac{\partial N^N}{\partial \mathbf{x}} \mathbf{v} \cdot \frac{\partial N^M}{\partial \mathbf{x}} \, dV \left(\theta^{M,t+\Delta t} - \theta^{M,t} \right) \\
&+ \left(\frac{1}{2} \int \frac{\partial N^N}{\partial \mathbf{x}} \cdot \mathbf{k} \cdot \frac{\partial N^M}{\partial \mathbf{x}} \, dV - \alpha \frac{h}{4} \int \frac{\mathbf{v}}{|\mathbf{v}|} \cdot \frac{\partial N^N}{\partial \mathbf{x}} \, \mathbf{k} : \frac{\partial^2 N^M}{\partial \mathbf{x} \partial \mathbf{x}} \, dV \right) \left(\theta^{M,t+\Delta t} + \theta^{M,t} \right) \\
&+ \beta \frac{h}{4} \int \frac{\mathbf{v}}{|\mathbf{v}|} \cdot \frac{\partial N^N}{\partial \mathbf{x}} \, \mathbf{k} : \frac{\partial^2 N^M}{\partial \mathbf{x} \partial \mathbf{x}} \, dV \left(\theta^{M,t+\Delta t} - \theta^{M,t} \right) \\
&- \int \left(N^N + \alpha \frac{h}{2} \frac{\mathbf{v}}{|\mathbf{v}|} \cdot \frac{\partial N^N}{\partial \mathbf{x}} \right) q \, dV - \int_S N^N q_s \, dS = 0.
\end{aligned}
$$

For the steady-state case the third term in this equation is omitted. In both transient and steady-state forms the contribution of such a convective element to the system of equations for the heat transfer model is not symmetric, requiring the use of the nonsymmetric matrix storage and solution scheme.

Reference

- "Uncoupled heat transfer analysis," Section 6.5.2 of the Abaqus Analysis User's Guide

2.11.4 CAVITY RADIATION

Product: Abaqus/Standard

The formulation described in this section provides a capability for modeling heat transfer with cavity thermal radiation (in addition to the radiation boundary conditions described in "Uncoupled heat transfer analysis," Section 2.11.1). Cavities are defined in Abaqus/Standard as collections of surfaces that are composed of facets. In axisymmetric and two-dimensional cases a facet is a side of an element; in three-dimensional cases a facet can be a face of a solid element or a surface of a shell element. For the purposes of the cavity radiation calculations, each facet is assumed to be isothermal and to have a uniform emissivity.

Based on the cavity definition, cavity radiation elements are created internally by Abaqus. These elements can generate large matrices since they couple the temperature degree of freedom of every node on the cavity surface. Their Jacobian matrix is nonsymmetric: the nonsymmetric solution capability is automatically invoked if cavity radiation calculations are requested in the analysis. Both steady-state and transient capabilities are provided.

The theory on which this cavity radiation formulation is based is well-known and can be found in Holman (1990) and Siegel and Howell (1980). This section describes the formulation of the cavity radiation flux contributions and respective Jacobian for the Newton method used for the solution of the nonlinear radiation problem. The geometrical issues associated with the calculation of radiation view factors necessary in the formulation are addressed in "View factor calculation," Section 2.11.5.

Thermal radiation

Our formulation is based on *gray body* radiation theory, which means that the monochromatic emissivity of the body is independent of the wavelength of propagation of the radiation. Only *diffuse* (nondirectional) reflection is considered. Attenuation of the radiation in the cavity medium is not considered. Using these assumptions together with the assumption of isothermal and isoemissive cavity facets, we can write the equations for radiation fluxes per unit area into cavity facets as

$$\sum_j \left(\delta_{ij} + (\epsilon_j - 1) F_{ij} \right) \frac{q_j^c}{\epsilon_j} = \sigma \sum_j F_{ij} \left((\theta_j - \theta^Z)^4 - (\theta_i - \theta^Z)^4 \right), \tag{2.11.4–1}$$

where q_j^c is the flux into facet j; ϵ_i, ϵ_j are the emissivities of facets i, j; σ is the Stefan-Boltzmann constant; F_{ij} is the geometrical view factor matrix; θ_i, θ_j are the temperatures of facets i, j; θ^Z is the value of absolute zero on the temperature scale being used; and δ_{ij} is the Kronecker delta.

In the special case of *blackbody* radiation, where no reflection takes place (emissivity equal to one), Equation 2.11.4–1 reduces to

$$q_i^c = \sigma \sum_j F_{ij} \left((\theta_j - \theta^Z)^4 - (\theta_i - \theta^Z)^4 \right). \tag{2.11.4–2}$$

Spatial interpolation

The variables used to solve the discrete approximation of the heat transfer problem with cavity radiation are the temperatures of the nodes on the cavity surface. Since we assume that for cavity radiation purposes each facet is isothermal, it is necessary to calculate an average facet *temperature radiation power*. We first define temperature radiation power as

$$\eta_i = \left(\theta_i - \theta^Z\right)^4, \qquad \eta^N = \left(\theta^N - \theta^Z\right)^4,$$

where the subscript i refers to facet quantities and the superscript N refers to nodal quantities.

Then, we interpolate the average facet temperature radiation power from the facet nodal temperatures as

$$\eta_i = \sum_N P_i^N \eta^N, \tag{2.11.4–3}$$

where N is the number of nodes forming the facet and P_i^N are nodal contribution factors calculated from area integration as

$$P_i^N = \frac{1}{A_i} \int_{A_i} N_i^N \, dA_i,$$

where A_i is the area and N_i^N are the interpolation functions for facet i.

The radiation flux into facet i can now be written as

$$Q_i = q_i^c A_i,$$

and the nodal contributions from the radiation flux on each facet can be written as

$$Q_i^N = \int_{A_i} q_i^c N_i^N \, dA_i = P_i^N Q_i.$$

The total radiation flux at node N is then

$$Q^N = \sum_i Q_i^N = \sum_i P_i^N Q_i. \tag{2.11.4–4}$$

Cavity radiation flux and Jacobian contributions

Abaqus/Standard provides two different schemes for obtaining the cavity radiation flux defined in Equation 2.11.4–1: a robust, serial method, suitable for small cavities, and a fully parallel method recommended for large cavities.

Serial solution of the cavity radiation equations

Thermal radiation problems involving small cavities allow us to solve Equation 2.11.4–1 for the radiation flux per unit area into a cavity facet as

$$q_i^c = \sigma \epsilon_i \sum_j \epsilon_j \sum_k F_{ik} C_{kj}^{-1} (\eta_j - \eta_i), \tag{2.11.4–5}$$

where

$$C_{ij} = \delta_{ij} - (1 - \epsilon_i) F_{ij}. \qquad \text{(no summation)}$$

Equation 2.11.4–5 requires the computation of the inverse C_{kj}^{-1}, which is why this method is suitable only for small cavities. The radiation flux into facet i can then be written as

$$Q_i = q_i^c A_i = \sum_j R_{ij} \, (\eta_j - \eta_i) \, ,$$

where

$$R_{ij} = \sigma \epsilon_i \epsilon_j \sum_k F_{ik} C_{kj}^{-1},$$

or more compactly as

$$Q_i = \sum_j \overline{R}_{ij} \eta_j, \tag{2.11.4–6}$$

where

$$\overline{R}_{ij} = R_{ij} - \left(\sum_k R_{ik} \right) \delta_{ij}.$$

Substituting Equation 2.11.4–3 and Equation 2.11.4–6 above into Equation 2.11.4–4, we can write the nodal contributions from the radiation flux as

$$Q^N = \sum_M \overline{R}^{NM} \eta^M, \tag{2.11.4–7}$$

where

$$\overline{R}^{NM} = \sum_i \sum_j P_i^N \overline{R}_{ij} P_j^M.$$

The radiation flux q_i^c is evaluated based on temperatures at the end of the increment, coordinates at the end of the increment, and emissivities at the beginning of the increment. Any time variation of the

coordinates during the heat transfer analysis is predefined as translational and/or rotational motion and, therefore, provides no contribution to the Jacobian. Any variation of the emissivities as a function of temperature and predefined field variable changes with time is treated explicitly (values at the beginning of the increment are used) and, therefore, also provides no contribution to the Jacobian. You can specify the maximum allowable emissivity change during an increment of the heat transfer analysis. Thus, the only Jacobian contribution is provided by temperature variations.

The Jacobian contribution arising from the cavity radiation flux is then written trivially as

$$J^{NM} = \frac{\partial Q^N}{\partial \theta^M} = 4\overline{R}^{NM}\left(\theta^M - \theta^Z\right)^3. \qquad \text{(no summation)} \tag{2.11.4–8}$$

In all practical cases this Jacobian is unsymmetric. This exact unsymmetric Jacobian is always used when the serial method for cavity radiation analysis is performed.

Solution of parallel-decomposed cavities

Abaqus/Standard provides a parallel scheme for the calculation of view factors and the solution of the cavity radiation equations of large cavities. Once parallel decomposition is enabled for a particular cavity, Abaqus/Standard will use an iterative solution technique for obtaining the radiative heat fluxes from Equation 2.11.4–1. This iterative technique is based on a Krylov method with a preconditioner.

Since we do not obtain the inverse C_{kj}^{-1} as in the serial method above, we do not have access to the exact Jacobian in Equation 2.11.4–8. Instead, we use an approximation to the Jacobian based on small changes to the irradiation (any part not due to emission from the surface). Since the resulting approximation is sparse, iterations during solution of the heat transfer finite element equations are carried out much faster than with the exact expression. However, since the Jacobian is approximate, convergence will not be quadratic in the vicinity of the solution. In fact, Abaqus/Standard may require many more iterations when cavity parallel decomposition is enabled than with the serial method, especially in the case of steady-state analyses and models containing surfaces with low emissivities. In these cases we recommend switching the analysis to transient steps and allowing for more iterations per increment in the solution of the heat transfer finite element equations.

Reference

- “Cavity radiation,” Section 41.1.1 of the Abaqus Analysis User’s Guide

2.11.5 VIEW FACTOR CALCULATION

Product: Abaqus/Standard

Cavity radiation occurs when surfaces of the model can see each other and, thus, exchange heat with each other by radiation (Figure 2.11.5–1). Such exchange depends on view factors that measure the relative interaction between the surfaces composing the cavity. View factor calculation is rather complicated for anything but the most trivial geometries. Abaqus offers an automatic view factor calculation capability for two- and three-dimensional cases as well as for axisymmetric situations. This capability can take into account general surface blocking (or shadowing) as well as the most common forms of radiation symmetry. The view factor calculation can also be automatically repeated a number of times throughout the analysis history (this is user-controlled) if cavity surfaces are moved in space causing the view factors to change.

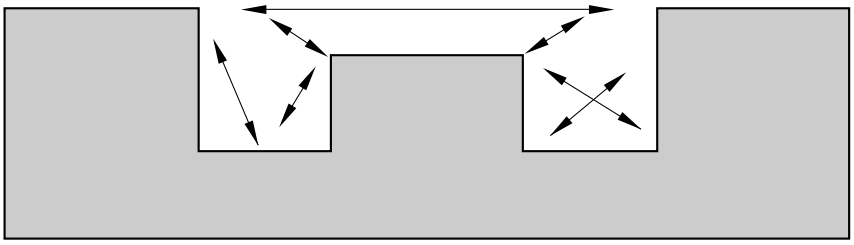


Figure 2.11.5–1 Heat exchange between surfaces by radiation.

SIMULIA is pleased to acknowledge that the view factor calculation technology implemented in Abaqus was derived from technology originally developed by the Atomic Energy Authority of the United Kingdom; see, for example, Johnson (1987). The remainder of this section contains a general description of this technology.

View factor between elementary areas

The dimensionless view factor F_{ij} between two elementary areas, A_i and A_j, satisfies the relation

$$A_i F_{ij} = \int_{A_i} \int_{A_j} \frac{cos\phi_i \, cos\phi_j}{\pi R_{ij}^2} \, dA_i \, dA_j, \tag{2.11.5–1}$$

where R_{ij} is the distance between the two areas and ϕ_i, ϕ_j are the angles between R_{ij} and the normals to the surfaces of the areas (Figure 2.11.5–2). The view factor also satisfies the reciprocity relation

$$A_i F_{ij} = A_j F_{ji}.$$

When the areas A_i and A_j are small compared with the distance R_{ij}, we can compute view factors in an area-lumped fashion; so the integral Equation 2.11.5–1 would become

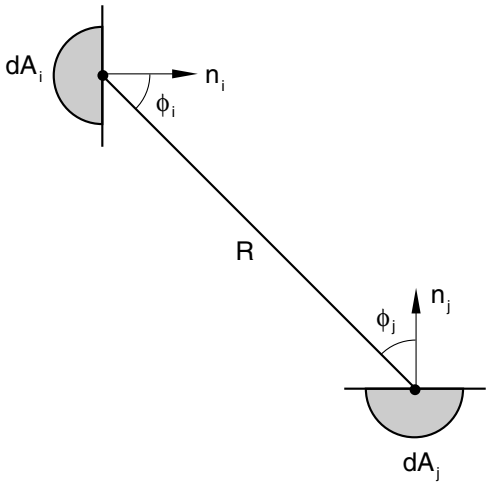


Figure 2.11.5–2 Schematic of view factor calculation.

$$A_i F_{ij} = A_j F_{ji} = \frac{A_i cos\phi_i A_j cos\phi_j}{\pi R_{ij}^2}. \tag{2.11.5–2}$$

However, this expression is singular in R_{ij}, so it is not well behaved for general surfaces. Thus, we use

$$A_i F_{ij} = A_j F_{ji} = \frac{4\sqrt{A_i A_j}}{\pi^2}\, atan\left[\frac{\sqrt{\pi A_i} cos\phi_i}{2R_{ij}}\right] atan\left[\frac{\sqrt{\pi A_j} cos\phi_j}{2R_{ij}}\right]. \tag{2.11.5–3}$$

The limit of Equation 2.11.5–3 for large R_{ij} is Equation 2.11.5–2.

If the areas A_i and A_j are not small compared with the distance R_{ij}, but one of the areas is much larger than the other (say $A_i << A_j$), we can use the formula

$$A_i F_{ij} = A_j F_{ji} = \frac{1}{2\pi} \sum_p \mathbf{g}_p \cdot \mathbf{n}_i, \tag{2.11.5–4}$$

where $\mathbf{n}_i$ is the unit normal to the smaller area A_j. The vector $\mathbf{g}_p$ is normal to the triangle defined by the centroid of area A_i and the edge p of area A_j, and $\mathbf{g}_p$ has magnitude equal to the angle opposite to edge p. See, for example, Hottel and Sarofim (1967).

For all other configurations Abaqus uses the method developed by Mitalas and Stephenson (1966), where Stokes theorem is used in Equation 2.11.5–1 and one of the contour integrals is solved analytically. The remaining contour integral is evaluated numerically,

$$A_i F_{ij} = A_j F_{ji} = \frac{1}{2\pi} \sum_p \sum_q \Phi_{pq} \sum_k (t \log t \cos\phi + s \log s \cos\theta + u\omega - r) dv_k, \tag{2.11.5–5}$$

where Φ_{pq} is the dot product between edge p of area A_j and edge q of area A_i. The angles ϕ, θ, and ω are the angles of the triangle defined by edge q and the integration point k. The scalars s, t, and r are the lengths of the sides opposed to ϕ, θ, and ω, respectively. The height of the triangle perpendicular to side r is u. The integration point length is v_k along the edge p.

Discretization

Cavities are composed of surfaces; and surfaces, in turn, are made up of finite element faces. For the purpose of view factor calculations, one can then think of a cavity as a collection of element faces (or facets) corresponding to the finite element discretization around the cavity.In the two- and three-dimensional cases the element faces composing cavities can be treated as elementary areas and, thus, Equation 2.11.5–1 applies. In axisymmetric cases the element faces represent rings so that the view factors involve two ring surfaces looking at each other. This requires an integration over 2π where it is important to account for "horizon" effects (Johnson, 1987).

In so far as the view factor calculations are concerned, first- and second-order element faces are treated similarly in the sense that the midside nodes of the faces in the second-order elements are ignored. This means that a pair of four-noded faces looking at each other will produce the same view factor as a pair of eight-noded faces with corner nodes coinciding with the nodes of the four-noded faces.

Radiation blocking

Radiation within a cavity implies that every surface exchanges heat with every other surface. The problem is made more complex when solid bodies are interposed between radiating surfaces blocking (or shadowing) off some but not all the possible paths along which heat can be radiated from the facets of one surface to the facets of another surface (Figure 2.11.5–3).

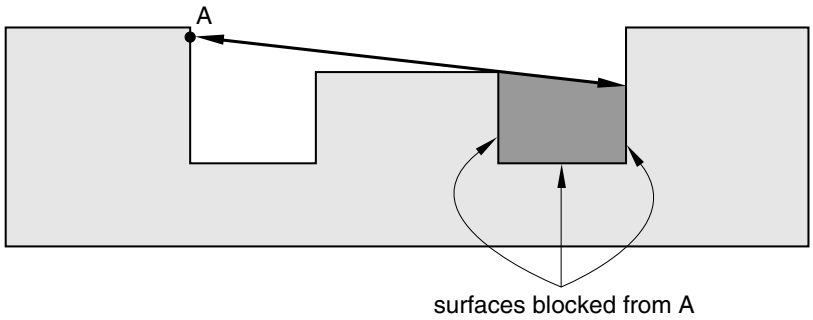


Figure 2.11.5–3 Blocking or shadowing example.

It is inconceivable that the user could handle this complexity in all but the simplest situations. Therefore, by default, Abaqus automatically checks if blocking takes place for every possible radiation path in a cavity. This requires that the program check if the ray joining the centers of each pair of facets intersects any other facet. For cavities with a large number of facets this can be very time consuming. For this reason Abaqus allows the user to guide its blocking algorithm by accepting input of which

surfaces cause blocking, thus significantly reducing the computational effort required. If a ray between two facets intersects any other facet, then in the two- and three-dimensional cases that ray is eliminated and no radiative heat transfer takes place between the facets. In the axisymmetric case blocking is much more complicated since each element face in the finite element model represents a ring. This is handled automatically and requires that the program calculate which part of the 2π extent of the ring is blocked.

Radiation symmetries

Use of symmetry can greatly reduce the size of a problem, but—in the case of cavity radiation—it requires that special facilities for definition and handling of symmetries be available. Abaqus provides capabilities for three different kinds of symmetries: simple reflection symmetry, periodic symmetry, and cyclic symmetry. Reflection symmetry allows one additional image of the model to be created by reflection through a line in two dimensions or reflection through a plane in three dimensions. Periodic symmetry can be used to create multiple images of the model by periodic repetition in two- or three-dimensional space according to a periodic distance vector. Cyclic symmetry creates multiple images of the model by cyclic repetition about a point in two dimensions or by cyclic repetition about an axis in three dimensions. Combinations of the different types of symmetry are supported.

To illustrate the handling of symmetries during view factor calculation, consider the case of a simple reflection symmetry in two-dimensional space (Figure 2.11.5–4).

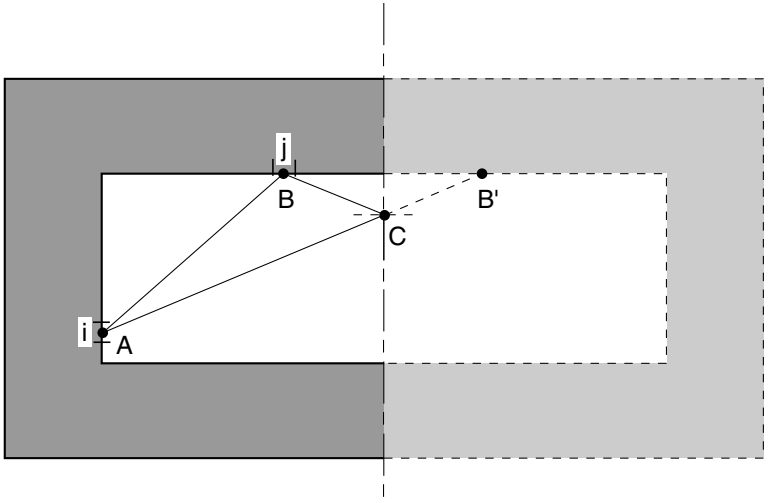


Figure 2.11.5–4 Reflection symmetry example.

Radiation between facet i (with its centroid at point A) and facet j (with its centroid at point B) has two contributions: one arising from the ray between points A and B and the other coming from the ray between points A and B', where B' is the mirror image of B. The length of ray AB is defined directly in the model. The definition of the length of ray AB' requires that point C on the reflection symmetry line

be located such that AC and BC make equal angles to it; ray AB' then has length $AC + BC$. Similar logic can be extended to the three-dimensional case.

In axisymmetric cases symmetry about the axis of symmetry of the model is always implied, and the only other symmetries allowed are simple reflection through a plane normal to the axis of symmetry or periodic repetition in the direction of the axis of symmetry.

View factor checking

The dimensionless view factor is a purely geometrical quantity, and it has some special properties. One property that allows us to check the accuracy of the calculation is that, for a completely enclosed cavity

$$\sum_j F_{ij} = 1. \tag{2.11.5–6}$$

This sum is equal to one as a result of the fact that all rays from surface i must strike some other surface j in an enclosed cavity. For an open cavity this sum is always less than one, indicating radiation to the ambient.

The quantity in Equation 2.11.5–6 is calculated for every facet of each cavity, and its value is used to provide a check to control the accuracy of the view factor calculation.

Radiation to ambient

The quantity calculated in Equation 2.11.5–6 can deviate from unity so long as the cavity is not fully enclosed. The user can define such an open cavity by giving the value of the ambient temperature in the cavity definition. In this case the difference between one and the quantity calculated in Equation 2.11.5–6 for each facet of the open cavity is considered to be the fraction radiating from that facet to the surrounding medium.

References

- "Thermal contact properties," Section 37.2.1 of the Abaqus Analysis User's Guide
- "Cavity radiation," Section 41.1.1 of the Abaqus Analysis User's Guide

2.12 Coupled thermal-electrical analysis

- "Coupled thermal-electrical analysis," Section 2.12.1

2.12.1 COUPLED THERMAL-ELECTRICAL ANALYSIS

Product: Abaqus/Standard

Joule heating arises when the energy dissipated by an electrical current flowing through a conductor is converted into thermal energy. Abaqus/Standard provides a fully coupled thermal-electrical procedure for analyzing this type of problem. Coupling arises from two sources: the conductivity in the electrical problem is temperature dependent, and the internal heat generated in the thermal problem is a function of electrical current. The thermal part of the problem includes all the heat conduction and heat storage (specific and latent heat) features described in "Uncoupled heat transfer analysis," Section 2.11.1. (Forced heat convection caused by fluid flowing through the mesh is not considered.)

The thermal-electrical elements have both temperature and electrical potential as nodal variables.

This section describes the governing equilibrium equations, the constitutive model, boundary conditions, the surface interaction model, finite element discretization, and the components of the Jacobian used.

Governing equations

The electric field in a conducting material is governed by Maxwell's equation of conservation of charge. Assuming steady-state direct current, the equation reduces to

$$\int_S \mathbf{J} \cdot \mathbf{n}\, dS = \int_V r_c\, dV, \tag{2.12.1–1}$$

where V is any control volume whose surface is S, $\mathbf{n}$ is the outward normal to S, $\mathbf{J}$ is the electrical current density (current per unit area), and r_c is the internal volumetric current source per unit volume.

The divergence theorem is used to convert the surface integral into a volume integral:

$$\int_V \left[\frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{J} - r_c \right] dV = 0,$$

and since the volume is arbitrary, this provides the pointwise differential equation

$$\frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{J} - r_c = 0.$$

The equivalent weak form is obtained by introducing an arbitrary, variational, electrical potential field, $\delta\varphi$, and integrating over the volume:

$$\int_V \delta\varphi \left[\frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{J} - r_c \right] dV = 0.$$

Using first the chain rule and then the divergence theorem, this statement can be rewritten as

$$-\int_V \frac{\partial \delta\varphi}{\partial \mathbf{x}} \cdot \mathbf{J}\, dV = \int_S \delta\varphi\, J\, dS + \int_V \delta\varphi\, r_c\, dV,$$

where $J \stackrel{\text{def}}{=} -\mathbf{J} \cdot \mathbf{n}$ is the current density entering the control volume across S.

Constitutive behavior

The flow of electrical current is described by Ohm's law:

$$\mathbf{J} = \boldsymbol{\sigma}^E \cdot \mathbf{E},$$

where $\boldsymbol{\sigma}^E(\theta, f^\alpha)$ is the electrical conductivity matrix; θ is the temperature; and $f^\alpha, \alpha = 1, 2...$ are any predefined field variables. The conductivity can be isotropic, orthotropic, or fully anisotropic. $\mathbf{E}(\mathbf{x})$ is the electrical field intensity defined as

$$\mathbf{E} = -\frac{\partial \varphi}{\partial \mathbf{x}}. \tag{2.12.1–2}$$

Since a potential rise occurs when a charged particle moves against the electrical field, the direction of the gradient is opposite to that of the electrical field. Using this definition of the electrical field, Ohm's law is rewritten as

$$\mathbf{J} = -\boldsymbol{\sigma}^E \cdot \frac{\partial \varphi}{\partial \mathbf{x}}. \tag{2.12.1–3}$$

The constitutive relation is linear; that is, it assumes that the electrical conductivity is independent of the electrical field.

Introducing Ohm's law, the governing conservation of charge equation becomes

$$\int_V \frac{\partial \delta\varphi}{\partial \mathbf{x}} \cdot \boldsymbol{\sigma}^E \cdot \frac{\partial \varphi}{\partial \mathbf{x}}\, dV = \int_V \delta\varphi\, r_c\, dV + \int_S \delta\varphi\, J\, dS. \tag{2.12.1–4}$$

Thermal energy balance

The heat conduction behavior is described by the basic energy balance relation

$$\int_V \rho \dot{U} \delta\theta\, dV + \int_V \frac{\partial \delta\theta}{\partial \mathbf{x}} \cdot \mathbf{k} \cdot \frac{\partial \theta}{\partial \mathbf{x}} dV = \int_V \delta\theta\, r\, dV + \int_S \delta\theta\, q\, dS, \tag{2.12.1–5}$$

where V is a volume of solid material, with surface area S; ρ is the density of the material; U is the internal energy; $\mathbf{k}$ is the thermal conductivity matrix; q is the heat flux per unit area of the body, flowing into the body; and r is the heat generated within the body. The thermal problem is discussed in detail in "Uncoupled heat transfer analysis," Section 2.11.1.

Equation 2.12.1–4 and Equation 2.12.1–5 describe the electrical and thermal problems, respectively. Coupling arises from two sources: the conductivity in the electrical problem is temperature dependent, $\boldsymbol{\sigma}^E = \boldsymbol{\sigma}^E(\theta)$, and the internal heat generation in the thermal problem is a function of electrical current, $r = r_{ec}(\mathbf{J})$, as described below.

Thermal energy due to electrical current

Joule's law describes the rate of electrical energy, P_{ec}, dissipated by current flowing through a conductor as

$$P_{ec} = \mathbf{E} \cdot \mathbf{J}.$$

Using Equation 2.12.1–2 and Equation 2.12.1–3, Joule's law is rewritten as

$$P_{ec} = \mathbf{E} \cdot \boldsymbol{\sigma}^E \cdot \mathbf{E}.$$

In a steady-state analysis P_{ec} is evaluated at time $t + \Delta t$. In a transient analysis an averaged value of P_{ec} is obtained over the increment

$$\begin{aligned} P_{ec} &= \frac{1}{\Delta t} \int_{\Delta t} P_{ec}\, dt \\ &= \mathbf{E} \cdot \boldsymbol{\sigma}^E \cdot \mathbf{E} - \mathbf{E} \cdot \boldsymbol{\sigma}^E \cdot \Delta\mathbf{E} + \frac{1}{3}\Delta\mathbf{E} \cdot \boldsymbol{\sigma}^E \cdot \Delta\mathbf{E}, \end{aligned}$$

where $\mathbf{E}$ and $\boldsymbol{\sigma}^E$ are values at time $t + \Delta t$. The amount of this energy released as internal heat is

$$r = \eta_v\, P_{ec},$$

where η_v is an energy conversion factor.

Surface conditions

The surface—S—of the body consists of parts on which boundary conditions can be prescribed—S_p—and parts that can interact with nearby surfaces of other bodies—S_i. Prescribed boundary conditions include the electrical potential, $\varphi = \varphi(\mathbf{x}, t)$; temperature, $\theta = \theta(\mathbf{x}, t)$; electrical current density, $J = J(\mathbf{x}, t)$; heat flux, $q = q(\mathbf{x}, t)$; and surface convection and radiation conditions. The surface interaction model includes heat conduction and radiation effects between the interface surfaces and electrical current flowing across the interface. Heat conduction and radiation are modeled by

$$q_c = k_g(\theta_B - \theta),$$

and

$$q_r = F_B(\theta_B - \theta^z)^4 - F(\theta - \theta^z)^4,$$

respectively, where θ is the temperature on the surface of the body under consideration, θ_B is the temperature on the surface of the other body, θ^z is the value of absolute zero on the temperature scale being used, $k_g(\bar{\theta}, \bar{f}^\alpha)$ is the gap thermal conductance, $\bar{\theta} = \frac{1}{2}(\theta + \theta_B)$ is the average interface

temperature, $\bar{f}^{\alpha} = \frac{1}{2}(f_A^{\alpha} + f_B^{\alpha})$ is the average of any predefined field variables at A and B, and $\boldsymbol{F}$ and F_B are constants.

The electrical current flowing between the interface surfaces is modeled as

$$J = \sigma_g(\varphi_B - \varphi),$$

where φ is the electrical potential on the surface of the body under consideration, φ_B is the electrical potential on the surface of the other body, and $\sigma_g(\bar{\theta}, \bar{f}^{\alpha})$ is the gap electrical conductance. The electrical energy dissipated by the current flowing across the interface,

$$P_{ec} = J(\varphi_B - \varphi) = \sigma_g(\varphi_B - \varphi)^2,$$

is released as heat on the surfaces of the bodies:

$$q_{ec} = f\eta_g P_{ec}, \qquad \text{and} \qquad q_{ec}^B = (1-f)\eta_g P_{ec},$$

where η_g is an energy conversion factor and $\boldsymbol{f}$ specifies how the total heat is distributed between the interface surfaces. P_{ec} is evaluated at the end of the time increment in a steady-state analysis, and an averaged value over the time increment is used in a transient analysis. This is described in detail in "Heat generation caused by electrical current," Section 5.2.6.

Introducing the surface interaction effects and electrical energy released as thermal energy, the governing electric and thermal equations become

$$\int_V \frac{\partial \delta\varphi}{\partial \mathbf{x}} \cdot \boldsymbol{\sigma}^E \cdot \frac{\partial \varphi}{\partial \mathbf{x}}\, dV = \int_V \delta\varphi\, r_c\, dV + \int_{S_p} \delta\varphi\, J\, dS + \int_{S_i} \delta\varphi\, \sigma_g\, (\varphi_B - \varphi)\, dS, \tag{2.12.1–6}$$

and

$$\begin{aligned} \int_V \rho \dot{U} \delta\theta\, dV + \int_V \frac{\partial \delta\theta}{\partial \mathbf{x}} \cdot \mathbf{k} \cdot \frac{\partial \theta}{\partial \mathbf{x}} dV &= \int_V \delta\theta\, r\, dV + \int_V \delta\theta\, \eta_v P_{ec}\, dV \\ &+ \int_{S_p} \delta\theta\, q\, dS + \int_{S_i} \delta\theta\, (q_c + q_r + q_{ec})\, dS. \end{aligned} \tag{2.12.1–7}$$

Spatial discretization

In a finite element model equilibrium is approximated as a finite set of equations by introducing interpolation functions. Discretized quantities are indicated by uppercase superscripts (for example, φ^N). The summation convention is adopted for the superscripts. The discretized quantities represent nodal variables, with nodes shared between adjacent elements and appropriate interpolation chosen to provide adequate continuity of the assumed variation.

The virtual electrical potential field is interpolated by

$$\delta\varphi = \mathrm{N}^N \delta\varphi^N,$$

where N^N are the interpolation functions. The discretized electrical equation is then written as

$$\delta\varphi^N \left\{ \int_V \frac{\partial N^N}{\partial \mathbf{x}} \cdot \boldsymbol{\sigma}^E \cdot \frac{\partial \varphi}{\partial \mathbf{x}}\, dV = \int_V N^N r_c\, dV + \int_{S_p} N^N J\, dS + \int_{S_i} N^N \sigma_g \left(\varphi_B - \varphi\right) dS \right\}.$$

Since $\delta\varphi$ is arbitrary,

$$I_\varphi^N = \int_V \frac{\partial N^N}{\partial \mathbf{x}} \cdot \boldsymbol{\sigma}^E \cdot \frac{\partial \varphi}{\partial \mathbf{x}}\, dV - \int_V N^N r_c\, dV - \int_{S_p} N^N J\, dS - \int_{S_i} N^N \sigma_g \left(\varphi_B - \varphi\right) dS = 0. \quad (2.12.1\text{–}8)$$

The temperature field in the thermal problem is approximated by the same set of interpolation functions:

$$\delta\theta = \mathrm{N}^P \delta\theta^P.$$

Using these interpolation functions and a backward difference operator to integrate the internal energy rate, $\dot{U}$, the thermal energy balance relation is obtained:

$$\begin{aligned} I_\theta^P = {} & \frac{1}{\Delta t} \int_V N^P \rho (U_{t+\Delta t} - U_t)\, dV + \int_V \frac{\partial N^P}{\partial \mathbf{x}} \cdot \mathbf{k} \cdot \frac{\partial \theta}{\partial \mathbf{x}} dV - \int_V N^P r\, dV \\ & - \int_V N^P \eta_v P_{ec}\, dV - \int_{S_p} N^P q\, dS - \int_{S_i} N^P \left(q_c + q_r + q_{ec}\right) dS = 0. \end{aligned} \quad (2.12.1\text{–}9)$$

Jacobian contributions

The Jacobian contributions are obtained by taking variations of Equation 2.12.1–8 and Equation 2.12.1–9 with respect to the electrical potential, φ, and the temperature, θ, at time $t + \Delta t$. This yields

$$\mathbf{K}_{\varphi\varphi}^{NM} = \frac{\partial I_\varphi^N}{\partial \varphi^M} = \int_V \frac{\partial N^N}{\partial \mathbf{x}} \cdot \boldsymbol{\sigma}^E \cdot \frac{\partial N^M}{\partial \mathbf{x}}\, dV + \int_{S_i} N^N \sigma_g N^M\, dS,$$

$$\mathbf{K}_{\varphi\theta}^{NQ} = \frac{\partial I_\varphi^N}{\partial \theta^Q} = -\int_V \frac{\partial N^N}{\partial \mathbf{x}} \cdot \frac{\partial \boldsymbol{\sigma}^E}{\partial \theta} \cdot \mathbf{E}\, N^Q\, dV - \frac{1}{2} \int_{S_i} N^N \frac{\partial \sigma_g}{\partial \bar{\theta}} (\varphi_B - \varphi)\, N^Q\, dS,$$

$$\mathbf{K}_{\theta\varphi}^{PM} = \frac{\partial I_\theta^P}{\partial \varphi^M} = \int_V N^P \eta_v \left(\mathbf{J} - \frac{1}{3}\Delta\mathbf{J} \right) \cdot \frac{\partial N^M}{\partial \mathbf{x}}\, dV \; + \; 2 \int_{S_i} N^P \, f \eta_g \sigma_g (\varphi_B - \varphi)\, N^M \, dS,$$

$$\begin{aligned}
\mathbf{K}_{\theta\theta}^{PQ} = \frac{\partial I_\theta^P}{\partial \theta^Q} = {} & \frac{1}{\Delta t} \int_V N^P \rho \frac{dU}{d\theta} N^Q \, dV \; + \; \int_V \frac{\partial N^P}{\partial \mathbf{x}} \cdot \mathbf{k} \cdot \frac{\partial N^Q}{\partial \mathbf{x}} \, dV \; + \; \int_V \frac{\partial N^P}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{k}}{\partial \theta} \cdot \frac{\partial \theta}{\partial \mathbf{x}} N^Q \, dV \\
& - \int_V N^P \eta_v \left(\mathbf{E} \cdot \frac{\partial \boldsymbol{\sigma}^E}{\partial \theta} \cdot \mathbf{E} - \mathbf{E} \cdot \frac{\partial \boldsymbol{\sigma}^E}{\partial \theta} \cdot \Delta\mathbf{E} + \frac{1}{3} \Delta\mathbf{E} \cdot \frac{\partial \boldsymbol{\sigma}^E}{\partial \theta} \cdot \Delta\mathbf{E} \right) N^Q \, dV \\
& - \int_{S_p} N^P \frac{\partial q}{+\partial \theta} N^Q \, dS - \int_{S_i} N^P \left(\frac{\partial q_c}{\partial \theta} + \frac{\partial q_r}{\partial \theta} + \frac{\partial q_{ec}}{\partial \theta} \right) N^Q \, dS.
\end{aligned}$$

The term $\partial q / \partial \theta$ in the $\mathbf{K}_{\theta\theta}^{PQ}$ component includes prescribed surface convection and radiation conditions. The surface interaction terms $\partial q_c / \partial \theta$, $\partial q_r / \partial \theta$, and $\partial q_{ec} / \partial \theta$ are evaluated in "Heat generation caused by electrical current," Section 5.2.6.

The Jacobian contributions give rise to an unsymmetric system of equations, requiring the use of the nonsymmetric matrix storage and solution scheme.

Reference

- "Coupled thermal-electrical analysis," Section 6.7.3 of the Abaqus Analysis User's Guide

2.13 Mass diffusion

- “Mass diffusion analysis,” Section 2.13.1

2.13.1 MASS DIFFUSION ANALYSIS

Product: Abaqus/Standard

Abaqus/Standard provides for the modeling of the transient or steady-state diffusion of one material through another, such as the diffusion of hydrogen through a metal (Crank (1956), deGroot and Mazur (1962)). The governing equations are an extension of Fick's equations, to allow for nonuniform solubility of the diffusing substance in the base material.

The basic solution variable (used as the degree of freedom at the nodes of the mesh) is the "normalized concentration" (often referred to as the "activity" of the diffusing material), $\phi \stackrel{\text{def}}{=} c/s$, where c is the mass concentration of the diffusing material and s is its solubility in the base material. This means that when the mesh includes dissimilar materials that share nodes, the normalized concentration is continuous across the interface between the different materials. Since ϕ is the square root of the partial pressure of the diffusing phase, the partial pressure is the same on both sides of the interface; Sievert's law is assumed to hold at the interface.

Governing equations

The diffusion problem is defined from the requirement of mass conservation for the diffusing phase:

$$\int_V \frac{dc}{dt}\, dV + \int_S \mathbf{n} \cdot \mathbf{J}\, dS = 0, \tag{2.13.1–1}$$

where V is any volume whose surface is S, $\mathbf{n}$ is the outward normal to S, $\mathbf{J}$ is the flux of concentration of the diffusing phase, and $\mathbf{n} \cdot \mathbf{J}$ is the flux of concentration leaving S.

Using the divergence theorem,

$$\int_V \left(\frac{dc}{dt} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{J} \right) dV = 0.$$

Because the volume is arbitrary, this provides the pointwise equation

$$\frac{dc}{dt} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{J} = 0.$$

The equivalent weak form is

$$\int_V \delta\phi \left(\frac{dc}{dt} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{J} \right) dV = 0,$$

where $\delta\phi$ is an arbitrary, suitably continuous, scalar field.

This statement can be rewritten as

$$\int_V \left[\delta\phi \left(\frac{dc}{dt} \right) + \frac{\partial}{\partial \mathbf{x}} \cdot (\delta\phi\, \mathbf{J}) - \mathbf{J} \cdot \frac{\partial\, \delta\phi}{\partial \mathbf{x}} \right] dV = 0.$$

Using the divergence theorem again yields

$$\int_V \left[\delta\phi \left(\frac{dc}{dt} \right) - \frac{\partial\, \delta\phi}{\partial \mathbf{x}} \cdot \mathbf{J} \right] dV + \int_S \delta\phi\, \mathbf{n} \cdot \mathbf{J}\, dS = 0. \tag{2.13.1–2}$$

Constitutive behavior

The diffusion is assumed to be driven by the gradient of a chemical potential, which gives the general behavior

$$\mathbf{J} = -s\mathbf{D} \cdot \left[\frac{\partial \phi}{\partial \mathbf{x}} + \kappa_s \frac{\partial}{\partial \mathbf{x}} \left(\ln(\theta - \theta^Z) \right) + \kappa_p \frac{\partial p}{\partial \mathbf{x}} \right], \tag{2.13.1–3}$$

where $\mathbf{D}(c, \theta, \mathbf{f})$ is the diffusivity; $s(\theta, \mathbf{f})$ is the solubility; $\kappa_s(c, \theta, \mathbf{f})$ is the "Soret effect" factor, providing diffusion because of the temperature gradient; θ is the temperature; θ^Z is the absolute zero on the temperature scale used; $\kappa_p(c, \theta, \mathbf{f})$ is the pressure stress factor, providing diffusion driven by the gradient of the equivalent pressure stress, $p \stackrel{\text{def}}{=} -\text{trace}(\boldsymbol{\sigma})/3$; and $\mathbf{f}$ are any predefined field variables.

An example of a particular form of this constitutive model is the assumption made for hydrogen diffusion in a metal:

$$\mathbf{J} = -\frac{\mathbf{D}c}{R(\theta - \theta^Z)} \cdot \frac{\partial \mu}{\partial \mathbf{x}},$$

with the chemical potential, μ, defined as

$$\mu = \mu^0 + R(\theta - \theta^Z) \ln \phi + p\overline{V}_H,$$

where μ^0 is a fixed datum, $\boldsymbol{R}$ is the universal gas constant, and $\overline{V}_H$ is the partial molar volume of hydrogen in the solid solution. This form is similar to that used by Sofronis and McMeeking (1989) and results in a constitutive expression of the form

$$\mathbf{J} = -s\mathbf{D} \cdot \left[\frac{\partial \phi}{\partial \mathbf{x}} + \phi \ln \phi \frac{\partial}{\partial \mathbf{x}} \left(\ln(\theta - \theta^Z) \right) + \phi \frac{\overline{V}_H}{R(\theta - \theta^Z)} \frac{\partial p}{\partial \mathbf{x}} \right].$$

To implement this particular form, data for κ_s and κ_p must be calculated from the equations

$$\kappa_s = \phi \ln \phi \quad \text{and} \quad \kappa_p = \phi \frac{\overline{V}_H}{R(\theta - \theta^Z)}.$$

Changing variables ($c = \phi s$) and introducing the constitutive assumption of Equation 2.13.1–3 into Equation 2.13.1–2 yields

$$\int_V \left[\delta\phi \left(s \frac{d\phi}{dt} + \phi \frac{ds}{d\theta} \frac{d\theta}{dt} \right) + \frac{\partial\, \delta\phi}{\partial \mathbf{x}} \cdot s\mathbf{D} \cdot \left(\frac{\partial \phi}{\partial \mathbf{x}} + \frac{\kappa_s}{(\theta - \theta^Z)} \frac{\partial \theta}{\partial \mathbf{x}} + \kappa_p \frac{\partial p}{\partial \mathbf{x}} \right) \right] dV = \int_S \delta\phi\, q\, dS, \tag{2.13.1–4}$$

where

$$q \overset{\text{def}}{=} -\mathbf{n} \cdot \mathbf{J}$$

is the concentration flux entering the body across $\mathcal{S}$.

Discretization and time integration

Equilibrium in a finite element model is approximated by a finite set of equations through the introduction of appropriate interpolation functions. Discretized quantities are indicated by uppercase superscripts (for example, ϕ^N). The summation convention is adopted for the superscripts. These represent nodal variables, with nodes shared between adjacent elements and appropriate interpolation chosen to provide adequate continuity of the assumed variation. The interpolation is based on material coordinates S_i, $i = 1, 2, 3$.

The virtual normalized concentration field is interpolated by

$$\delta\phi = \mathrm{N}^N \delta\phi^N,$$

where $\mathrm{N}^N(S_i)$ are interpolation functions. Then, the discretized equations are written as

$$\int_V \left[\mathrm{N}^N \left(s\frac{d\phi}{dt} + \phi\frac{ds}{d\theta}\frac{d\theta}{dt} \right) + \frac{\partial \mathrm{N}^N}{\partial \mathbf{x}} \cdot s\mathbf{D} \cdot \left(\frac{\partial \phi}{\partial \mathbf{x}} + \frac{\kappa_s}{(\theta - \theta^Z)} \frac{\partial \theta}{\partial \mathbf{x}} + \kappa_p \frac{\partial p}{\partial \mathbf{x}} \right) \right] dV = \int_S \mathrm{N}^N \, q \, dS. \tag{2.13.1–5}$$

Time integration in transient problems utilizes the backward Euler method (the modified Crank-Nicholson operator). Adopting the convention that any quantity not explicitly associated with a point in time is taken at $t+\Delta t$, we can drop the subscript $t+\Delta t$ and write the integrated equations as

$$\int_V \left[\mathrm{N}^N \left(s\frac{(\phi - \phi_t)}{\Delta t} + \phi\frac{ds}{d\theta}\frac{d\theta}{dt} \right) + \frac{\partial \mathrm{N}^N}{\partial \mathbf{x}} \cdot s\mathbf{D} \cdot \left(\frac{\partial \phi}{\partial \mathbf{x}} + \frac{\kappa_s}{(\theta - \theta^Z)} \frac{\partial \theta}{\partial \mathbf{x}} + \kappa_p \frac{\partial p}{\partial \mathbf{x}} \right) \right] dV = \int_S \mathrm{N}^N \, q \, dS. \tag{2.13.1–6}$$

Jacobian contribution

The Jacobian contribution from the conservation equation is obtained from the variation of Equation 2.13.1–6 with respect to ϕ at time $t + \Delta t$. This yields

$$\int_V \left[\mathrm{N}^N \left(\frac{s}{\Delta t} d\phi + \frac{ds}{d\theta}\frac{d\theta}{dt} d\phi \right) + \frac{\partial \mathrm{N}^N}{\partial \mathbf{x}} \cdot s\frac{\partial \mathbf{D}}{\partial \phi} \cdot \left(\frac{\partial \phi}{\partial \mathbf{x}} + \frac{\kappa_s}{(\theta - \theta^Z)} \frac{\partial \theta}{\partial \mathbf{x}} + \kappa_p \frac{\partial p}{\partial \mathbf{x}} \right) d\phi \right.$$

$$\left. + \frac{\partial \mathrm{N}^N}{\partial \mathbf{x}} \cdot s\mathbf{D} \cdot \left(\frac{\partial d\phi}{\partial \mathbf{x}} + \frac{1}{(\theta - \theta^Z)} \frac{\partial \theta}{\partial \mathbf{x}} \frac{\partial \kappa_s}{\partial \phi} d\phi + \frac{\partial p}{\partial \mathbf{x}} \frac{\partial \kappa_p}{\partial \phi} d\phi \right) \right] dV.$$

Rearranging and using the interpolation $d\phi = \mathrm{N}^N d\phi^N$, we obtain

$$\int_V \left[\left(\frac{s}{\Delta t} + \frac{ds}{d\theta}\frac{d\theta}{dt} \right) \mathrm{N}^N \mathrm{N}^M + \frac{\partial \mathrm{N}^N}{\partial \mathbf{x}} \cdot s\mathbf{D} \cdot \frac{\partial \mathrm{N}^M}{\partial \mathbf{x}} \right.$$

$$\left. + s\frac{\partial \mathrm{N}^N}{\partial \mathbf{x}} \cdot \left\{ \frac{\partial \mathbf{D}}{\partial \phi} \cdot \left(\frac{\partial \phi}{\partial \mathbf{x}} + \frac{\kappa_s}{(\theta - \theta^Z)} \frac{\partial \theta}{\partial \mathbf{x}} + \kappa_p \frac{\partial p}{\partial \mathbf{x}} \right) + \mathbf{D} \cdot \left(\frac{1}{(\theta - \theta^Z)} \frac{\partial \theta}{\partial \mathbf{x}} \frac{\partial \kappa_s}{\partial \phi} + \frac{\partial p}{\partial \mathbf{x}} \frac{\partial \kappa_p}{\partial \phi} \right) \right\} \mathrm{N}^M \right] \, dV.$$

Inspecting the above equation, we observe that the Jacobian becomes unsymmetric whenever the diffusivity, $\mathbf{D}$; the temperature-driven diffusion coefficient, κ_s; or the pressure-driven diffusion coefficient, κ_p, is defined as a function of concentration.

Reference

- "Mass diffusion analysis," Section 6.9.1 of the Abaqus Analysis User's Guide

2.14 Substructuring

- “Substructuring and substructure analysis,” Section 2.14.1

2.14.1 SUBSTRUCTURING AND SUBSTRUCTURE ANALYSIS

Product: Abaqus/Standard

The basic substructuring idea is to consider a "substructure" (a part of the model) separately and eliminate all but the degrees of freedom needed to connect this part to the rest of the model so that the substructure appears in the model as a "substructure": a collection of finite elements whose response is defined by the stiffness (and mass) of these retained degrees of freedom denoted by the vector, $\{u^R\}$.

In Abaqus/Standard the response within a substructure, once it has been reduced to a substructure, is considered to be a linear perturbation about the state of the substructure at the time it is made into a substructure. Thus, the substructure is in equilibrium with stresses $\boldsymbol{\sigma}_0$, displacements $\mathbf{u}_0$, and other state variables h_0 when it is made into a substructure. Then, whenever it responds as a substructure, the total value of a displacement or stress component at some point within the substructure is

$$u = u_0 + \lfloor L_u^R \rfloor \{\Delta u^R\}$$
$$\sigma = \sigma_0 + \lfloor L_\sigma^R \rfloor \{\Delta u^R\},$$

where $\lfloor L_u^R(\mathbf{x}) \rfloor$ and $\lfloor L_\sigma^R(\mathbf{x}) \rfloor$ are linear transformations between the retained degrees of freedom of the substructure and the component of displacement or stress under consideration. The substructure must be in a self-equilibriating state when it is made into a substructure (except for reaction forces at prescribed boundary conditions that are applied to internal degrees of freedom in the substructure). If the substructure has been loaded to a nonzero state with some of its retained degrees of freedom fixed, these fixities are released at the time the substructure is created and any reaction forces at them converted into concentrated loads that are part of the preload state. This means that the contribution of the substructure to the overall equilibrium of the model is defined entirely by its linear response. Since the purpose of the substructuring technique is to have the substructure contribute terms only to the retained degrees of freedom, we need to define its external load vector $\{\overline{P}^R\}$, formed from the nonzero substructure load cases applied to the substructure, and its internal force vector, $\{\overline{I}^R\}$, as a sum of linear transformations of the retained variables $\{\Delta u^R\}$ and their velocities and accelerations:

$$\{\overline{I}^R\} = [\overline{M}]\{\ddot{u}^R\} + [\overline{C}]\{\dot{u}^R\} + [\overline{K}]\{\Delta u^R\}.$$

We refer to $[\overline{M}]$ as the reduced mass matrix for the substructure, $[\overline{C}]$ as its reduced damping matrix, and $[\overline{K}]$ as its reduced stiffness. These "reduced" mass, damping, and stiffness matrices connect the retained degrees of freedom only.

The reduced stiffness matrix is easily derived when only static response is considered. Since the response of a substructure is entirely linear, its contribution to the virtual work equation for the model of which it is a part is

$$\delta W = \lfloor \delta u^R \quad \delta u^E \rfloor \left(\left\{ \begin{array}{c} \Delta P^R \\ \Delta P^E \end{array} \right\} - \left[\begin{array}{cc} K^{RR} & K^{RE} \\ K^{ER} & K^{EE} \end{array} \right] \left\{ \begin{array}{c} \Delta u^R \\ \Delta u^E \end{array} \right\} \right),$$

where $\{\Delta P^R\}$ and $\{\Delta P^E\}$ are consistent nodal forces applied to the substructure during its loading as a substructure (they do not include the self-equilibriating preloading of the substructure) and

$$[K] = \begin{bmatrix} K^{RR} & K^{RE} \\ K^{ER} & K^{EE} \end{bmatrix}$$

is its tangent stiffness matrix.

Since the internal degrees of freedom in the substructure, $\{u^E\}$, appear only within the substructure, the equilibrium equations conjugate to $\{\delta u^E\}$ in the contribution to the virtual work equation given above are complete within the substructure, so that

$$\{\Delta P^E\} - [K^{ER}]\{\Delta u^R\} - [K^{EE}]\{\Delta u^E\} = 0.$$

These equations can be rewritten to define Δu^E as

$$\{\Delta u^E\} = [K^{EE}]^{-1}\left(\{\Delta P^E\} - [K^{ER}]\{\Delta u^R\}\right). \tag{2.14.1–1}$$

The substructure's contribution to the static equilibrium equations is, therefore,

$$\delta W = \left\lfloor \delta u^R \right\rfloor \left(\left(\{\Delta P^R\} - [K^{RE}][K^{EE}]^{-1}\{\Delta P^E\}\right) - \left([K^{RR}] - [K^{RE}][K^{EE}]^{-1}[K^{ER}]\right)\{\Delta u^R\}\right).$$

Thus, for static analysis the substructure's reduced stiffness is

$$[\overline{K}] = [K^{RR}] - [K^{RE}][K^{EE}]^{-1}[K^{ER}],$$

and the contribution of the substructure load cases applied to the substructure is the load vector

$$\{\overline{P}^R\} = \{\Delta P^R\} - [K^{RE}][K^{EE}]^{-1}\{\Delta P^E\}.$$

The static modes defined by Equation 2.14.1–1 may not be sufficient to define the dynamic response of the substructure accurately. The substructure's dynamic representation may be improved by retaining additional degrees of freedom not required to connect the substructure to the rest of the model; that is, some of the u^E can be moved into u^R. This technique is known as Guyan reduction. An additional, and generally more effective, technique is to augment the response within the substructure by including some generalized degrees of freedom, q^α, associated with natural modes of the substructure. The simplest such approach is to extract some natural modes from the substructure with all retained degrees of freedom constrained, so that Equation 2.14.1–1 is augmented to be

$$\{\Delta u^E\} = [K^{EE}]^{-1}\left(\{\Delta P^E\} - [K^{ER}]\{\Delta u^R\}\right) + \{\phi^E\}^\alpha q^\alpha,$$

with the variation

$$\{\delta u^E\} = -[K^{EE}]^{-1}[K^{ER}]\{\delta u^R\} + \{\phi^E\}^\alpha \delta q^\alpha$$

and the time derivatives

$$\{\dot{u}^E\} = -[K^{EE}]^{-1}[K^{ER}]\{\dot{u}^R\} + \{\phi^E\}^\alpha \dot{q}^\alpha$$
$$\{\ddot{u}^E\} = -[K^{EE}]^{-1}[K^{ER}]\{\ddot{u}^R\} + \{\phi^E\}^\alpha \ddot{q}^\alpha.$$

The $\{\phi^E\}^\alpha$ are the eigenmodes of the substructure, obtained with all retained degrees of freedom constrained, and the q^α are the generalized displacements—the magnitudes of the response in these normal modes.

The contribution of the substructure to the virtual work equation for the dynamic case is

$$\{\,\delta u^R \quad \delta u^E\,\}\left(\begin{Bmatrix}\Delta P^R\\ \Delta P^E\end{Bmatrix} - \begin{bmatrix}M^{RR} & M^{RE}\\ M^{ER} & M^{EE}\end{bmatrix}\begin{Bmatrix}\ddot{u}^R\\ \ddot{u}^E\end{Bmatrix} - \begin{bmatrix}C^{RR} & C^{RE}\\ C^{ER} & C^{EE}\end{bmatrix}\begin{Bmatrix}\dot{u}^R\\ \dot{u}^E\end{Bmatrix} - \begin{bmatrix}K^{RR} & K^{RE}\\ K^{ER} & K^{EE}\end{bmatrix}\begin{Bmatrix}\Delta u^R\\ \Delta u^E\end{Bmatrix}\right),$$

where

$$[M] = \begin{bmatrix}M^{EE} & M^{ER}\\ M^{RE} & M^{RR}\end{bmatrix}$$

is the substructure's mass matrix,

$$[C] = \begin{bmatrix}C^{EE} & C^{ER}\\ C^{RE} & C^{RR}\end{bmatrix}$$

is its damping matrix, and

$$\{P\} = \begin{Bmatrix}\Delta P^R\\ \Delta P^E\end{Bmatrix}$$

is the nodal force vector in the substructure.

With the assumed dynamic response within the substructure, the internal degrees of freedom in this contribution (Δu^E and its time derivatives) can be transformed to the retained degrees of freedom and the normal mode amplitudes, reducing the system to

$$\lfloor\,\delta u^R \quad \delta q\,\rfloor\left([T]^T\{P\} - [T]^T[M][T]\begin{Bmatrix}\ddot{u}^R\\ \ddot{q}\end{Bmatrix} - [T]^T[C][T]\begin{Bmatrix}\dot{u}^R\\ \dot{q}\end{Bmatrix} - [T]^T[K][T]\begin{Bmatrix}\Delta u^R\\ \Delta q\end{Bmatrix}\right),$$

where

$$[T] = \begin{bmatrix}[I] & [0]\\ -[K^{EE}]^{-1}[K^{ER}] & [\phi^E]\end{bmatrix},$$

in which $[\phi^E]$ is the matrix of eigenvectors, $\{q\}$ is the vector of generalized degrees of freedom, $[I]$ is a unit matrix, and $[0]$ is a null matrix.

Large-rotation substructures

Large-rotation substructures require first the computation of an equivalent rigid body rotation matrix $\mathbf{R}$ associated with the substructure's motion. Since the substructure exhibits only small deformations, one can use the original and current positions of two nodes in two-dimensional analyses or three nodes in three-dimensional analyses to compute two rectangular local systems and then the rotation matrix. For example, in three dimensions Abaqus/Standard computes a reference (average) point using the three nodes in the original configuration. The first unit direction vector, E_j^1, points from this average point to the first node. The third direction, E_j^3, is taken to be the normal to the plane defined by the three nodes and the second direction, E_j^2, is simply the cross product of the third and first directions. The process is repeated in the current configuration to compute a local system, e_i^k. The rotation matrix can then be easily computed as $R_{ij} = e_i^k E_j^k$.

Abaqus/Standard automatically picks the two or three nodes used for the computation of the rotation matrix $\mathbf{R}$ from the substructure's retained nodes. In most cases only retained nodes with at least all translational degrees of freedom retained can be canditates. For example, in three-dimensional analyses the first node used for the equivalent rigid body calculation is chosen to be the node with the highest stiffness (largest diagonal value) in the substructure. The second node is chosen to be the retained node farthest apart from the first node with the provision that its nodal stiffness is high enough (at least 0.01% of the stiffness of the first node). The third node is picked to be the retained node for which the distance to the line defined by the first and second node is maximum (with the same stiffness requirement as for the second node). In the rare case when less than three (in three-dimensional analyses) valid candidate nodes are retained, the $\mathbf{R}$ matrix is computed directly from the nodal rotations of the stiffest node with all rotational degrees of freedom retained.

To compute internal forces associated with a substructure in large rotations, Abaqus/Standard computes strain-inducing displacements/rotations by "subtracting" the rigid body motion from the substructure's nodal displacements/rotations. For translational degrees of freedom the strain-inducing displacements $\mathbf{u}$ at a node can be computed using

$$\mathbf{u} = \mathbf{x} - \mathbf{x_0} - \mathbf{R}(\mathbf{X} - \mathbf{X_0}),$$

where $\mathbf{X}$ and $\mathbf{x}$ are the original and current positions of the node and $\mathbf{X_0}$ and $\mathbf{x_0}$ are the original and current coordinates of an average point (calculated as outlined above). For rotational degrees of freedom the total rotation matrix at a node ($\mathbf{R^{tot}}$) is the compound rotation between the strain-inducing rotation matrix and the rigid body rotation

$$\mathbf{R^{tot}} = \mathbf{exp}\left[\hat{\phi}\right]\mathbf{R}.$$

The strain-inducing rotations ϕ can then be easily extracted. The internal force in the substructure can thus be written as

$$\overline{\mathbf{F}} = \overline{\mathbf{K}}^{rot}\overline{\mathbf{u}},$$

where

$$\overline{\mathbf{K}}^{rot} = \mathbf{R}\overline{\mathbf{K}}\mathbf{R}^{\mathbf{T}}$$

is the rigid body rotated stiffness matrix and

$$\overline{\mathbf{u}} = \left\{ \begin{array}{c} \mathbf{u} \\ \phi \end{array} \right\}$$

is the strain-inducing displacement-rotation vector.

Similarly, in dynamics the reduced mass (including the coupling between nodal displacements/rotations and eigenmode contributions) is rigid body rotated before any mass contributions are included in the virtual work associated with the substructure.

Fixed direction gravity loading

When gravity loading that acts in a fixed direction is defined, Abaqus/Standard will create internally at the generation level a number of load cases (two in two-dimensional analyses and three in three-dimensional analyses) corresponding to unit gravity loads in the substructure's directions. At the usage level the total rotation matrix of the substructure $\mathbf{R}$ (includes both the user-specified rotation/mirroring and the rotation of the substructure in nonlinear geometry analyses) is first used to rotate back the user-specified unit direction $\mathbf{N}$ for the gravity load. Thus, the user-specified directions are now expressed in the local (rotating) system associated with the substructure ($\mathbf{n} = \mathbf{R}^{\mathbf{T}} \cdot \mathbf{N}$). The internally generated unit load cases are then scaled by the components of $\mathbf{n}$ and by the appropriate magnitude and amplitude and then added to the external force in the model.

Reference

- "Substructuring," Section 10.1 of the Abaqus Analysis User's Guide

2.15 Submodeling

- “Submodeling analysis,” Section 2.15.1

2.15.1 SUBMODELING ANALYSIS

Products: Abaqus/Standard Abaqus/Explicit

Submodeling is the technique of studying a local part of a model with a refined mesh, based on interpolation of the solution from an initial, global model onto appropriate parts of the boundary of the submodel. The method is most useful when it is necessary to obtain an accurate, detailed solution in the local region and the detailed modeling of that local region has negligible effect on the overall solution. The response at the boundary of the local region is defined by the solution for the global model and it, together with any loads applied to the local region, determines the solution in the submodel. The technique relies on the global model defining this submodel boundary response with sufficient accuracy.

Submodeling can be applied quite generally in Abaqus. With a few restrictions different element types can be used in the submodel compared to those used to model the corresponding region in the global model. Both the global model and the submodel can use solid elements, or they can both use shell elements. A special option is available to use a submodel consisting of solid elements with a global model consisting of shell elements. The material response defined for the submodel may also be different from that defined for the global model. Both the global model and the submodel can have nonlinear response and can be analyzed for any sequence of analysis procedures. The procedures do not have to be the same for both models.

The submodel is run as a separate analysis. The only link between the submodel and the global model is the transfer of the time-dependent values of variables to the relevant driven variables of the submodel. The only information in the global model available to the submodel analysis is the file output data written during the global model analysis. These data contain, by default, the undeformed coordinates of all global model nodes and element information for all elements in the global model (see "Results file output format," Section 5.1.2 of the Abaqus Analysis User's Guide). The user must have requested appropriate responses in the area where the submodel boundary is located.

Two forms of the submodeling technique are implemented in Abaqus. The more general node-based submodeling technique transfers node-located solution variables, most commonly displacements, from global model nodes to submodel nodes. A surface-based submodeling technique, which transfers material point stress results from the global model to surface load integration points in the submodel, is also available.

Node-based submodeling

Node-based submodeling is the most commonly used technique. With this technique global model responses are used to prescribe boundary conditions at the driven nodes in the submodel.

Interpolation procedure and tolerance checking

In the solid-to-solid case the positions of the submodel boundary nodes (the driven nodes) are determined with respect to the global model, and the appropriate element interpolation functions are used to obtain the values of the degrees of freedom at the driven nodes. An "exterior tolerance," which the user can specify, is used to check whether it is valid to extrapolate values from the global model. In cases where the submodel boundary nodes fall outside the global model, the extrapolation is valid if the distance between the driven nodes and the free surface of the global model falls within the specified tolerance.

A similar check is done along the global model boundaries for the shell-to-shell submodeling case. We also check whether the driven nodes of the submodel lie sufficiently close to the midsurface of the shell elements in the global model. To simplify the calculations, the closest point in the global model is approximated by measuring the distance in the direction normal to a flat approximation to each shell element in the global model, as shown in Figure 2.15.1–1.

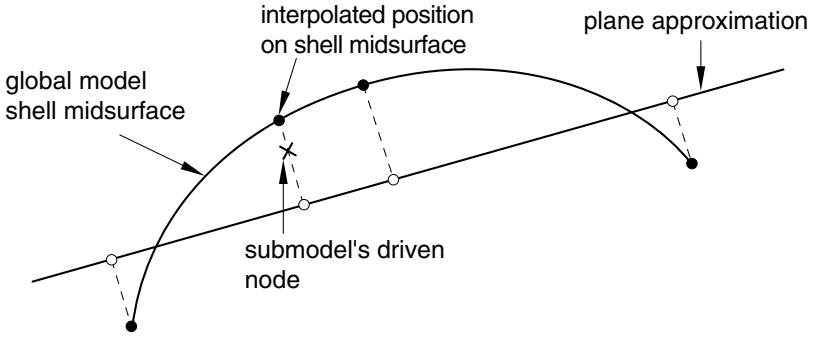


Figure 2.15.1–1 Flat surface approximation in shell-to-shell submodeling.

For the shell-to-solid case Abaqus uses two kinds of tolerances to determine the relation between the submodel and the global model. First, the closest point on the shell midsurface of the global model is determined. This point will subsequently be referred to as the "image node" of the driven node. The exterior tolerance parameter is used to check if the image node lies within the domain of the global model. Then the distance between the driven node and its image is checked against half of the maximum shell thickness specified by the user (see Figure 2.15.1–2).

If the node is within the half thickness plus the exterior tolerance, it is accepted. This check is only approximate if the global model has varying shell thickness, and in that case it will not protect the user in parts of the global model that have a small thickness compared to the maximum thickness specified by the user.

After the locations of the driven nodes (or image nodes for the shell-to-solid case) are determined, the prescribed values of the driven variables are interpolated from the values written to the file output for the global model. These must have been written with a sufficiently high frequency to obtain accurate values at the driven nodes. All components of displacements, temperatures, charges, and—for complex steady-state dynamic analysis—the phase angles as well as the amplitudes have to be written for the global model nodes from which the values for the driven nodes will be interpolated. For small global models responses will typically be written for all nodes. For large global models node sets can be created that contain the nodes in the regions around the submodel boundary.

For solid-to-solid and shell-to-shell submodeling, the interpolated values of displacements, rotations, temperatures, etc. are applied directly to the driven nodes. For these nodes the user can specify the individual degrees of freedom that are driven.

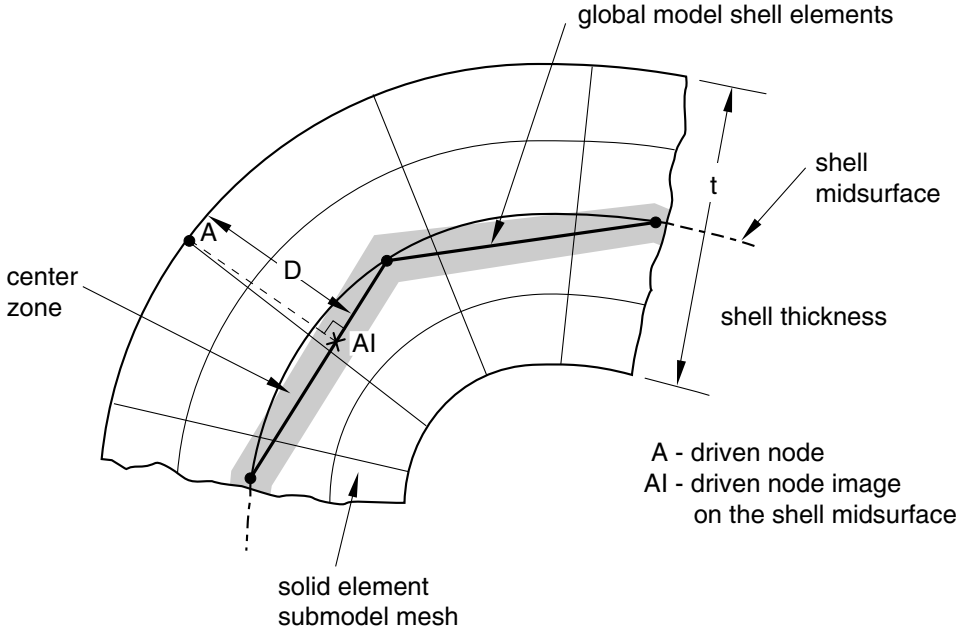


Figure 2.15.1–2 Center zone in shell-to-solid submodeling.

Driven variables for shell-to-solid submodeling

In the shell-to-solid case the driven degrees of freedom are chosen automatically, depending on the distance between the driven node and the midsurface of the shell. If the node lies within the center zone (specified by the user; see Figure 2.15.1–2), all displacement components are driven. If the node lies outside the center zone, only the displacement components parallel to the shell midsurface are driven. By default, the size of the center zone is taken as 10% of the maximum shell thickness. The procedure is described in detail below. The center zone should be large enough so that it contains at least one layer of nodes. If the transverse shear stresses at the submodel boundary are high and the submodel is highly refined in the thickness direction, this can result in high local stresses, since the shear force at the submodel boundary is only transferred at the driven nodes within the center zone. High transverse shear stresses occur only in regions where bending moments vary rapidly, and it is better not to locate the submodel boundary in such regions. It is best to locate the submodel boundary in areas of low transverse shear stress in the global model.

All displacement degrees of freedom are driven when the driven node lies within the center zone. For geometrically linear analysis these prescribed displacements are obtained from the displacements and rotations of the image node as

$$\mathbf{u}^A = \mathbf{u}^{AI} + \boldsymbol{\phi}^{AI} \times \mathbf{D}, \tag{2.15.1–1}$$

where $\mathbf{u}^A$ is the prescribed displacement of driven node A, $\mathbf{u}^{AI}$ and $\boldsymbol{\phi}^{AI}$ are the interpolated displacement and rotation of the image node, and $\mathbf{D}$ is the vector connecting the image node to the driven node:

$$\mathbf{D} = \mathbf{X}^A - \mathbf{X}^{AI}.$$

For large-displacement analysis finite rotations must be taken into account. The finite rotation equivalent of Equation 2.15.1–1 is

$$\mathbf{u}^A = \mathbf{u}^{AI} + (\mathbf{C} - \mathbf{I}) \cdot \mathbf{D} = \mathbf{u}^{AI} + \mathbf{d} - \mathbf{D}, \tag{2.15.1–2}$$

where $\mathbf{C}$ is the rotation matrix as defined in "Rotation variables," Section 1.3.1; $\mathbf{I}$ is the identity tensor; and $\mathbf{d}$ is the rotated vector connecting the image node to the driven node in the current configuration:

$$\mathbf{d} = \mathbf{C} \cdot \mathbf{D}.$$

For driven nodes outside the center zone only the displacement components parallel to the shell midsurface are driven. For the geometrically linear case this leads to the constraints

$$\begin{aligned} \mathbf{T}_1 \cdot \mathbf{u}^A &= \mathbf{T}_1 \cdot (\mathbf{u}^{AI} + \boldsymbol{\phi}^{AI} \times \mathbf{D}), \\ \mathbf{T}_2 \cdot \mathbf{u}^A &= \mathbf{T}_2 \cdot (\mathbf{u}^{AI} + \boldsymbol{\phi}^{AI} \times \mathbf{D}), \end{aligned} \tag{2.15.1–3}$$

where $\mathbf{T}_1$ and $\mathbf{T}_2$ are two (unit) vectors orthogonal to $\mathbf{D}$. The equivalent expressions for the geometrically nonlinear case are

$$\begin{aligned} \mathbf{t}_1 \cdot \mathbf{u}^A &= \mathbf{t}_1 \cdot (\mathbf{u}^{AI} + \mathbf{d} - \mathbf{D}), \\ \mathbf{t}_2 \cdot \mathbf{u}^A &= \mathbf{t}_2 \cdot (\mathbf{u}^{AI} + \mathbf{d} - \mathbf{D}), \end{aligned} \tag{2.15.1–4}$$

where $\mathbf{t}_1$ and $\mathbf{t}_2$ are two (unit) vectors orthogonal to $\mathbf{d}$.

Since the submodeling capability in Abaqus is quite general and allows the use of different procedure types in both analyses, there are several possibilities for the evaluation of the values at driven nodes as follows. In all cases Abaqus assumes that the global model and the submodel both use small- or large-displacement theory.

In the schemes listed below the first procedure type applies to the global analysis and the second to the submodel analysis.

1. General procedure to general procedure for small-displacement theory: Equation 2.15.1–1 is used inside the center zone, and Equation 2.15.1–3 is used outside the center zone.
2. General procedure to general procedure for large-displacement theory: Equation 2.15.1–2 is used inside the center zone, and Equation 2.15.1–4 outside the center zone.
3. General procedure to linear perturbation procedure for small-displacement theory:

$$\boldsymbol{\Delta}\mathbf{u}^A = \mathbf{u}^{AI} + \boldsymbol{\phi}^{AI} \times \mathbf{D} - \mathbf{u}_0^A, \tag{2.15.1–5}$$

inside the center zone, and

$$\begin{aligned}\mathbf{T}_1 \cdot \boldsymbol{\Delta}\mathbf{u}^A &= \mathbf{T}_1 \cdot (\mathbf{u}^{AI} + \boldsymbol{\phi}^{AI} \times \mathbf{D} - \mathbf{u}_0^A),\\ \mathbf{T}_2 \cdot \boldsymbol{\Delta}\mathbf{u}^A &= \mathbf{T}_2 \cdot (\mathbf{u}^{AI} + \boldsymbol{\phi}^{AI} \times \mathbf{D} - \mathbf{u}_0^A),\end{aligned} \tag{2.15.1–6}$$

outside the center zone; $\mathbf{u}_0^A$ denotes the base state in the submodel.

4. General procedure to linear perturbation procedure for large-displacement theory:

$$\boldsymbol{\Delta}\mathbf{u}^A = \mathbf{u}^{AI} + \mathbf{d} - \mathbf{D} - \mathbf{u}_0^A \tag{2.15.1–7}$$

inside the center zone, and

$$\begin{aligned}\mathbf{t}_1 \cdot \boldsymbol{\Delta}\mathbf{u}^A &= \mathbf{t}_1 \cdot (\mathbf{u}^{AI} + \mathbf{d} - \mathbf{D} - \mathbf{u}_0^A),\\ \mathbf{t}_2 \cdot \boldsymbol{\Delta}\mathbf{u}^A &= \mathbf{t}_2 \cdot (\mathbf{u}^{AI} + \mathbf{d} - \mathbf{D} - \mathbf{u}_0^A)\end{aligned} \tag{2.15.1–8}$$

outside the center zone, where $\mathbf{t}_\alpha$ denotes the tangent vector. The exact formulation would require the use of the base state normal vector $\mathbf{d}^0$ and the base state tangent vector $\mathbf{t}_\alpha^0$. Since they are not available, Abaqus approximates them with the current normal vector $\boldsymbol{d}$ and current tangent vector $\mathbf{t}_\alpha$.

5. Linear perturbation procedure to general procedure for small-displacement theory:

$$\mathbf{u}^A = \boldsymbol{\Delta}\mathbf{u}^{AI} + \boldsymbol{\Delta}\boldsymbol{\phi}^{AI} \times \mathbf{D} + \mathbf{u}_0^A \tag{2.15.1–9}$$

inside the center zone, and

$$\begin{aligned}\mathbf{T}_1 \cdot \mathbf{u}^A &= \mathbf{T}_1 \cdot (\boldsymbol{\Delta}\mathbf{u}^{AI} + \boldsymbol{\Delta}\boldsymbol{\phi}^{AI} \times \mathbf{D} + \mathbf{u}_0^A),\\ \mathbf{T}_2 \cdot \mathbf{u}^A &= \mathbf{T}_2 \cdot (\boldsymbol{\Delta}\mathbf{u}^{AI} + \boldsymbol{\Delta}\boldsymbol{\phi}^{AI} \times \mathbf{D} + \mathbf{u}_0^A)\end{aligned} \tag{2.15.1–10}$$

outside the center zone; $\mathbf{u}_0^A$ denotes the base state in the submodel.

6. Linear perturbation procedure to general procedure for large-displacement theory:

$$\mathbf{u}^A = \boldsymbol{\Delta}\mathbf{u}^{AI} + \boldsymbol{\Delta}\boldsymbol{\phi}^{AI} \times \mathbf{D} + \mathbf{u}_0^A \tag{2.15.1–11}$$

inside the center zone, and

$$\begin{aligned}\mathbf{T}_1 \cdot \mathbf{u}^A &= \mathbf{T}_1 \cdot (\boldsymbol{\Delta}\mathbf{u}^{AI} + \boldsymbol{\Delta}\boldsymbol{\phi}^{AI} \times \mathbf{D} + \mathbf{u}_0^A),\\ \mathbf{T}_2 \cdot \mathbf{u}^A &= \mathbf{T}_2 \cdot (\boldsymbol{\Delta}\mathbf{u}^{AI} + \boldsymbol{\Delta}\boldsymbol{\phi}^{AI} \times \mathbf{D} + \mathbf{u}_0^A)\end{aligned} \tag{2.15.1–12}$$

outside the center zone. Since the base state is not available, an approximate form is used, where $\mathbf{D}$ is used in place of $\mathbf{d}^0$ and $\mathbf{T}_\alpha$ is used for $\mathbf{t}_\alpha^0$. With the above assumptions cases 5 and 6 are

governed by the same equations. The approximation will give good results for cases with a small base state rotation field in the global analysis.

7. Linear perturbation procedure to linear perturbation procedure for small-displacement theory:

$$\boldsymbol{\Delta}\mathbf{u}^{A} = \boldsymbol{\Delta}\mathbf{u}^{AI} + \boldsymbol{\Delta}\boldsymbol{\phi}^{AI} \times \mathbf{D} \tag{2.15.1–13}$$

inside the center zone, and

$$\begin{aligned} \mathbf{T}_1 \cdot \boldsymbol{\Delta}\mathbf{u}^{A} &= \mathbf{T}_1 \cdot (\boldsymbol{\Delta}\mathbf{u}^{AI} + \boldsymbol{\Delta}\boldsymbol{\phi}^{AI} \times \mathbf{D}), \\ \mathbf{T}_2 \cdot \boldsymbol{\Delta}\mathbf{u}^{A} &= \mathbf{T}_2 \cdot (\boldsymbol{\Delta}\mathbf{u}^{AI} + \boldsymbol{\Delta}\boldsymbol{\phi}^{AI} \times \mathbf{D}) \end{aligned} \tag{2.15.1–14}$$

outside the center zone.

8. Linear perturbation procedure to linear perturbation procedure for large-displacement theory:

$$\boldsymbol{\Delta}\mathbf{u}^{A} = \boldsymbol{\Delta}\mathbf{u}^{AI} + \boldsymbol{\Delta}\boldsymbol{\phi}^{AI} \times \mathbf{D} \tag{2.15.1–15}$$

inside the center zone, and

$$\begin{aligned} \mathbf{T}_1 \cdot \boldsymbol{\Delta}\mathbf{u}^{A} &= \mathbf{T}_1 \cdot (\boldsymbol{\Delta}\mathbf{u}^{AI} + \boldsymbol{\Delta}\boldsymbol{\phi}^{AI} \times \mathbf{D}), \\ \mathbf{T}_2 \cdot \boldsymbol{\Delta}\mathbf{u}^{A} &= \mathbf{T}_2 \cdot (\boldsymbol{\Delta}\mathbf{u}^{AI} + \boldsymbol{\Delta}\boldsymbol{\phi}^{AI} \times \mathbf{D}) \end{aligned} \tag{2.15.1–16}$$

outside the center zone. Since the base state is not available, $\boldsymbol{D}$ is used in place of $\mathbf{d}^0$ and $\mathbf{T}_\alpha$ in place of $\mathbf{t}^0_\alpha$. With the above assumptions cases 7 and 8 are governed by the same equations. The approximation will give good results for cases with a small base state rotation field in the global analysis.

Surface-based submodeling

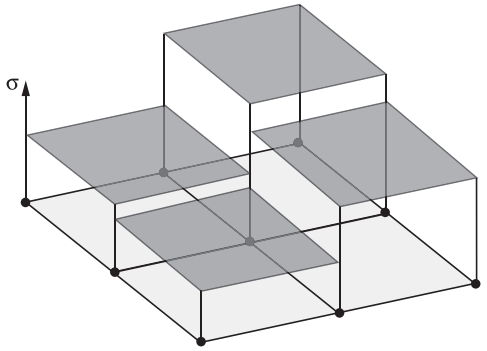
With the surface-based submodeling technique global model responses are used to prescribe fluxes on driven surfaces in the submodel. Currently this technique is limited to use with stresses, and the flux is a surface-applied traction.

Interpolation procedure and tolerance checking

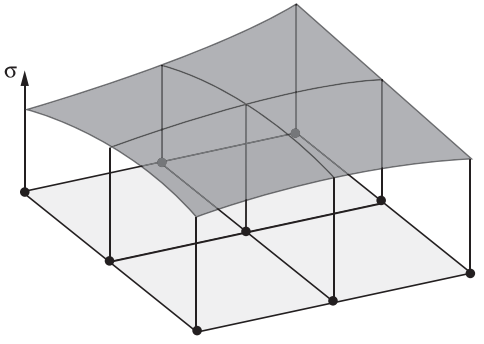
The interpolation procedure resembles that for node-based submodeling in solids. In the surface-based case, though, the positions of the submodel boundary surface integration points (the driven integration points) are determined with respect to the global model, and the appropriate element interpolation functions are used to obtain the values of the stress tensor at the given integration point. An “exterior tolerance,” which the user can specify, is used to check whether it is valid to extrapolate values from the global model. In cases where the submodel falls outside the global model, the extrapolation is valid if the distance between the driven integration points and the free surface of the global model falls within the specified tolerance.

Stress solution smoothing

After the locations of the driven integration points in the global mesh are determined, a prescribed stress at the integration point is interpolated from node-located stress values from the global model. These driving-node-located stresses are determined from the global model material point stress values through a patch recovery technique. In this technique the driving node stress is determined from a polynomial curve fit of stress results in adjacent elements. The effect of this recovery technique is a smoothing of the stress solution, as shown in Figure 2.15.1–3.



Original stress via FEM



Stress modified by patch recovery

Figure 2.15.1–3 Relation between global model element stress results and the patch recovery calculated stress field used for node-based interpolation.

Because the driving node stress result is a function of the neighboring element stress results, the elements in the global model that contribute to the driving stress at a particular integration point extend

beyond the global element encompassing the driven integration point. Consider the submodel driven surface shown in Figure 2.15.1–4.

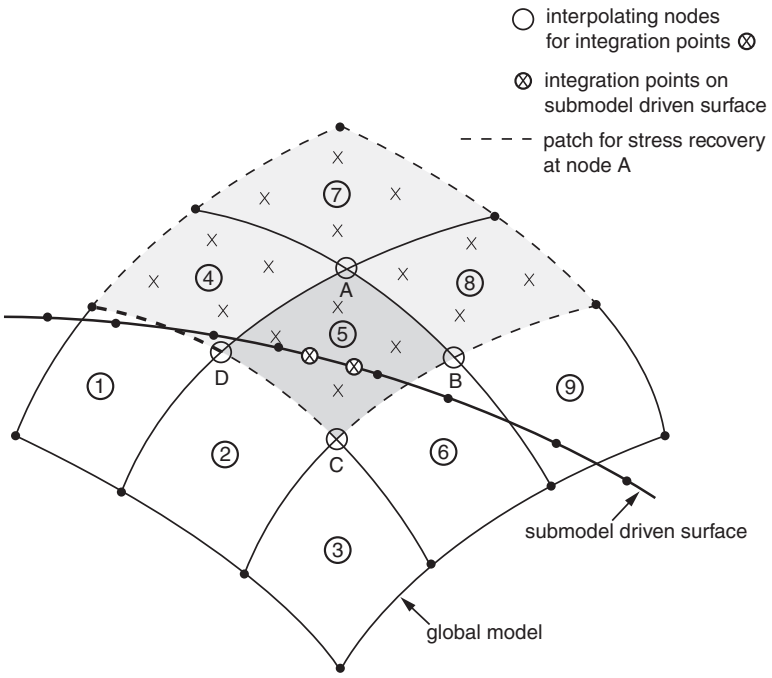


Figure 2.15.1–4 The extent of global elements contributing to a driving stress for integration points lying within a single global model element.

Stresses at driven integration points in the center of the figure, those inside the darkly shaded element number 5, are interpolated from global model nodes A, B, C, and D. The figure illustrates, through the lighter shading of elements 1, 4, 5, and 8, the global element contribution to the stress calculated at node A. Considering the additional contributions of nodes B, C, and D, the complete set of elements contributing to the driven integration points found inside global element 5 is, therefore, all the elements shown in the figure.

Surface traction determination

The submodel interpolation procedure locates global model stress results, $\boldsymbol{\sigma}_g$, at the driven surface integration points. These stress results then define submodel tractions, $\mathbf{t}_s$, based on the current submodel surface normal, $\mathbf{n}_s$:

$$\mathbf{t}_s = \mathbf{n}_s \cdot \boldsymbol{\sigma}_g. \tag{2.15.1–17}$$

In cases where inertia relief is employed in the submodel, a common case when the submodel is driven exclusively by global model stress results, the time evolution of the submodel surface normal will generally differ from the global model by a rigid body rotation. When this rotation discrepancy is large, this traction calculation can introduce significant errors in the submodel solution. See "Surface-based submodeling," Section 10.2.3 of the Abaqus Analysis User's Guide, for more information on identifying and addressing these cases where solution error may result.

References

- "Submodeling: overview," Section 10.2.1 of the Abaqus Analysis User's Guide
- "Node-based submodeling," Section 10.2.2 of the Abaqus Analysis User's Guide
- "Surface-based submodeling," Section 10.2.3 of the Abaqus Analysis User's Guide

2.16 Fracture mechanics

- "J-integral evaluation," Section 2.16.1
- "Stress intensity factor extraction," Section 2.16.2
- "T-stress extraction," Section 2.16.3
- "Prediction of the direction of crack propagation," Section 2.16.4

2.16.1 J-INTEGRAL EVALUATION

Product: Abaqus/Standard

The *J*-integral is widely accepted as a fracture mechanics parameter for both linear and nonlinear material response. It is related to the energy release associated with crack growth and is a measure of the intensity of deformation at a notch or crack tip, especially for nonlinear materials. If the material response is linear, it can be related to the stress intensity factors. Because of the importance of the *J*-integral in the assessment of flaws, its accurate numerical evaluation is vital to the practical application of fracture mechanics in design calculations. Abaqus/Standard provides a procedure for such evaluations of the *J*-integral, based on the virtual crack extension/domain integral methods (Parks, 1977, and Shih, Moran, and Nakamura, 1986). The method is particularly attractive because it is simple to use, adds little to the cost of the analysis, and provides excellent accuracy, even with rather coarse meshes.

J-integral in two dimensions

In the context of quasi-static analysis the *J*-integral is defined in two dimensions as

$$J = \lim_{\Gamma \to 0} \int_{\Gamma} \mathbf{n} \cdot \mathbf{H} \cdot \mathbf{q} \, d\Gamma, \tag{2.16.1–1}$$

where Γ is a contour beginning on the bottom crack surface and ending on the top surface, as shown in Figure 2.16.1–1; the limit $\Gamma \to 0$ indicates that Γ shrinks onto the crack tip; $\mathbf{q}$ is a unit vector in the virtual crack extension direction; and $\mathbf{n}$ is the outward normal to Γ. $\mathbf{H}$ is given by

$$\mathbf{H} = W\mathbf{I} - \boldsymbol{\sigma} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{x}}.$$

For elastic material behavior W is the elastic strain energy; for elastic-plastic or elastic-viscoplastic material behavior W is defined as the elastic strain energy density plus the plastic dissipation, thus representing the strain energy in an "equivalent elastic material." This implies that the *J*-integral calculation is suitable only for monotonic loading of elastic-plastic materials.

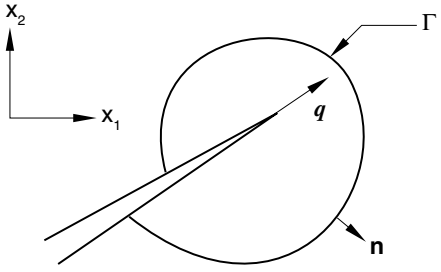


Figure 2.16.1–1 Contour for evaluation of the *J*-integral.

Following Shih et al. (1986), we rewrite Equation 2.16.1–1 in the form

$$J = -\oint_{C+C_+ +\Gamma+C_-} \mathbf{m}\cdot\mathbf{H}\cdot\bar{\mathbf{q}}\,d\Gamma - \int_{C_+ +C_-} \mathbf{t}\cdot\frac{\partial\mathbf{u}}{\partial\mathbf{x}}\cdot\bar{\mathbf{q}}\,d\Gamma, \tag{2.16.1–2}$$

where $\bar{\mathbf{q}}$ is a sufficiently smooth weighting function within the region enclosed by the closed contour $C + C_+ + \Gamma + C_-$ and has the value $\bar{\mathbf{q}} = \mathbf{q}$ on Γ and $\bar{\mathbf{q}} = 0$ on C; and $\mathbf{m}$ is the outward normal to the domain enclosed by the closed contour, as shown in Figure 2.16.1–2. $\mathbf{m} = -\mathbf{n}$ on Γ; and $\mathbf{t} = \mathbf{m}\cdot\boldsymbol{\sigma}$ is the surface traction on the crack surfaces C_+ and C_-.

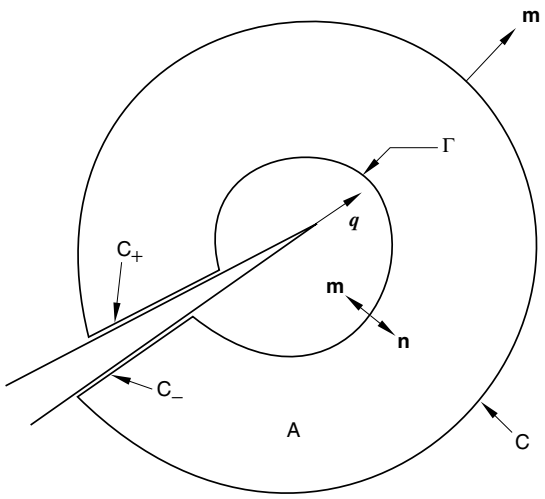


Figure 2.16.1–2 Closed contour $C + C_+ + \Gamma + C_-$ encloses a domain A that includes the crack-tip region as $\Gamma \to 0$.

Using the divergence theorem, we convert the closed contour integral into the domain integral

$$J = -\int_A \left(\frac{\partial}{\partial\mathbf{x}}\right)\cdot(\mathbf{H}\cdot\bar{\mathbf{q}})\;d\Gamma - \int_{C_+ +C_-} \mathbf{t}\cdot\frac{\partial\mathbf{u}}{\partial\mathbf{x}}\cdot\bar{\mathbf{q}}\,d\Gamma, \tag{2.16.1–3}$$

where A is the domain enclosed by the closed contour $C + C_+ + \Gamma + C_-$. It is worth noting that the domain A includes the crack-tip region as $\Gamma \to 0$.

If equilibrium is satisfied and W is a function of the mechanical strain—i.e., $W = W(\varepsilon^m)$—we have

$$\left(\frac{\partial}{\partial\mathbf{x}}\right)\cdot\boldsymbol{\sigma} + \mathbf{f} = 0 \quad \text{and} \quad \frac{\partial W}{\partial\mathbf{x}} = \frac{\partial W}{\partial\varepsilon^m} : \frac{\partial\varepsilon^m}{\partial\mathbf{x}} = \boldsymbol{\sigma} : \left(\frac{\partial\varepsilon}{\partial\mathbf{x}} - \frac{\partial\varepsilon^{th}}{\partial\mathbf{x}}\right),$$

where $\mathbf{f}$ is the body force per unit volume and $\boldsymbol{\varepsilon}^{th}$ is the thermal strain. Substituting the above two equations into Equation 2.16.1–3 gives

$$J = -\int_A \left[\mathbf{H} : \frac{\partial \bar{\mathbf{q}}}{\partial \mathbf{x}} + \left(\mathbf{f} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{x}} - \boldsymbol{\sigma} : \frac{\partial \boldsymbol{\varepsilon}^{th}}{\partial \mathbf{x}} \right) \cdot \bar{\mathbf{q}} \right] d\Gamma - \int_{C_+ + C_-} \mathbf{t} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{x}} \cdot \bar{\mathbf{q}} \, d\Gamma. \qquad (2.16.1\text{–}4)$$

To evaluate these integrals, Abaqus defines the domain in terms of rings of elements surrounding the crack tip. Different "contours" (domains) are created. The first contour consists of those elements directly connected to crack-tip nodes. The next contour consists of the ring of elements that share nodes with the elements in the first contour as well as the elements in the first contour. Each subsequent contour is defined by adding the next ring of elements that share nodes with the elements in the previous contour. $\bar{\mathbf{q}}$ is chosen to have a magnitude of zero at the nodes on the outside of the contour and to be one (in the crack direction) at all nodes inside the contour except for the midside nodes (if they exist) in the outer ring of elements. These midside nodes are assigned a value between zero and one according to the position of the node on the side of the element.

J-integral in three dimensions

The *J*-integral can be extended to three dimensions by considering a crack with a tangentially continuous front, as shown in Figure 2.16.1–3. The local direction of virtual crack extension is again given by $\mathbf{q}$,

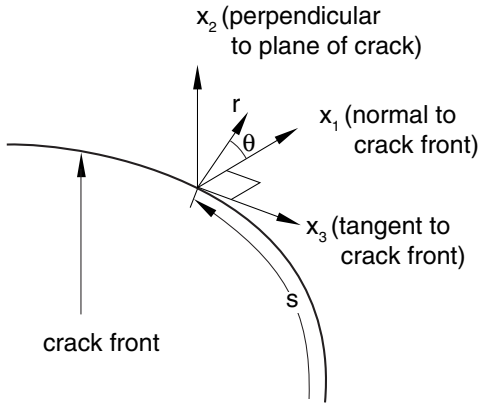


Figure 2.16.1–3 Definition of local orthogonal Cartesian coordinates at the point *s* on the crack front; the crack is in the x_1–x_3 plane.

which is perpendicular to the local crack front and lies in the crack plane. Asymptotically, as $r \to 0$, the conditions for path independence apply on any contour in the x_1–x_2 plane, which is perpendicular to the crack front at *s*. Hence, the *J*-integral defined in this plane can be extended to represent the pointwise energy release rate along the crack front as

$$J(s) = \lim_{\Gamma \to 0} \int_{\Gamma} \mathbf{n} \cdot \mathbf{H} \cdot \mathbf{q} \, d\Gamma. \tag{2.16.1–5}$$

For a virtual crack advance $\lambda(s)$ in the plane of a three-dimensional crack, the energy release rate is given by

$$\bar{J} = \int_{L} J(s)\lambda(s) ds = \lim_{\Gamma \to 0} \int_{A_t} \lambda(s) \mathbf{n} \cdot \mathbf{H} \cdot \mathbf{q} dA \,, \tag{2.16.1–6}$$

where L denotes the crack front under consideration; dA is a surface element on a vanishingly small tubular surface enclosing the crack tip (i.e., $dA = ds d\Gamma$); and $\mathbf{n}$ is the outward normal to dA. $\bar{J}$ can be calculated by the domain integral method similar to that used in two dimensions. To do so, we first convert the surface integral in Equation 2.16.1–6 to a volume integral by introducing a contour surface A_o, outside surface A_t, external surfaces A_{ends} at the ends of the crack front (the surfaces A_{ends} vanish for the crack whose front forms a closed loop), and the crack faces A_{cracks}, as shown in Figure 2.16.1–4.

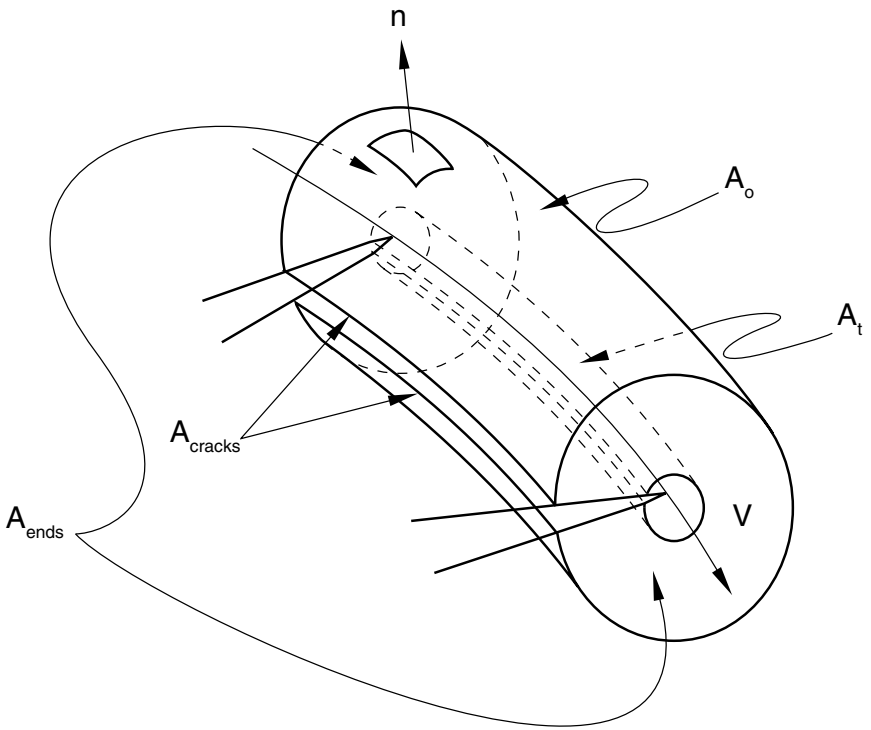


Figure 2.16.1–4 Surface $A = A_t + A_o + A_{ends} + A_{cracks}$ encloses a domain volume V that includes the crack-front region as $\Gamma \to 0$.

It can be seen that $A = A_t + A_o + A_{ends} + A_{cracks}$ encloses a volume V. A weighting function $\bar{\mathbf{q}}$ is defined such that it has a magnitude of zero on A_o and $\bar{\mathbf{q}} = \lambda(s)\mathbf{q}$ on A_t. $\bar{\mathbf{q}}$ is assumed to vary smoothly between these values within A. On the external surfaces A_{ends} where $\mathbf{q}$ is not tangential to the surfaces, it must be made so. This can be done in Abaqus by defining the surface normals explicitly. Then, we can rewrite Equation 2.16.1–6 as

$$\bar{J} = -\oint_A \mathbf{m}\cdot\mathbf{H}\cdot\bar{\mathbf{q}}\,dA - \int_{A_{ends}+A_{cracks}} \mathbf{t}\cdot\frac{\partial\mathbf{u}}{\partial\mathbf{x}}\cdot\bar{\mathbf{q}}\,dA, \tag{2.16.1–7}$$

where $\mathbf{m}$ is the outward normal to A (and $\mathbf{m} = -\mathbf{n}$ on A_t). $\mathbf{t} = \mathbf{m}\cdot\boldsymbol{\sigma}$ is the surface traction on surfaces A_{ends} and the crack surfaces A_{cracks}.

Using the divergence theorem, we obtain

$$\bar{J} = -\int_V \left[\mathbf{H} : \frac{\partial\bar{\mathbf{q}}}{\partial\mathbf{x}} + \left(\mathbf{f}\cdot\frac{\partial\mathbf{u}}{\partial\mathbf{x}} - \boldsymbol{\sigma} : \frac{\partial\boldsymbol{\varepsilon}^{th}}{\partial\mathbf{x}}\right)\cdot\bar{\mathbf{q}}\right] dV - \int_{A_{ends}+A_{cracks}} \mathbf{t}\cdot\frac{\partial\mathbf{u}}{\partial\mathbf{x}}\cdot\bar{\mathbf{q}}\,dA. \tag{2.16.1–8}$$

To obtain $J(s)$ at each node set P along the crack front line, $\lambda(s)$ is discretized with the same interpolation functions as those used in the finite elements along the crack front:

$$\lambda(s) = N^Q(s)\lambda^Q,$$

where $\lambda^Q = 1$ at the node set P and all other λ^Q are zero. This expression for $\lambda(s)$ is substituted into Equation 2.16.1–8. Finally, the J-integral value at each node set P along the crack front can be calculated as

$$J^P = \bar{J}^P \Big/ \int_L N^P\,ds. \tag{2.16.1–9}$$

Reference

- "Contour integral evaluation," Section 11.4.2 of the Abaqus Analysis User's Guide

2.16.2 STRESS INTENSITY FACTOR EXTRACTION

Product: Abaqus/Standard

The stress intensity factors K_I, K_{II}, and K_{III} play an important role in linear elastic fracture mechanics. They characterize the influence of load or deformation on the magnitude of the crack-tip stress and strain fields and measure the propensity for crack propagation or the crack driving forces. Furthermore, the stress intensity can be related to the energy release rate (the J-integral) for a linear elastic material through

$$J = \frac{1}{8\pi}\mathbf{K}^T \cdot \mathbf{B}^{-1} \cdot \mathbf{K},$$

where $\mathbf{K} = \lfloor K_I, K_{II}, K_{III} \rfloor^T$ and $\mathbf{B}$ is called the pre-logarithmic energy factor matrix (Shih and Asaro, 1988; Barnett and Asaro, 1972; Gao, Abbudi, and Barnett, 1991; Suo, 1990). For homogeneous, isotropic materials $\mathbf{B}$ is diagonal and the above equation simplifies to

$$J = \frac{1}{\bar{E}}(K_I^2 + K_{II}^2) + \frac{1}{2G}K_{III}^2,$$

where $\bar{E} = E$ for plane stress and $\bar{E} = E/(1-\nu^2)$ for plane strain, axisymmetry, and three dimensions. For an interfacial crack between two dissimilar isotropic materials with Young's moduli E_1 and E_2, Poisson's ratios ν_1 and ν_2, and shear moduli $G_1 = E_1/2(1+\nu_1)$ and $G_2 = E_2/2(1+\nu_2)$,

$$J = \frac{1-\beta^2}{E^*}(K_I^2 + K_{II}^2) + \frac{1}{2G^*}K_{III}^2,$$

where

$$\frac{1}{E^*} = \frac{1}{2}\Big(\frac{1}{\bar{E}_1} + \frac{1}{\bar{E}_2}\Big), \qquad \frac{1}{G^*} = \frac{1}{2}\Big(\frac{1}{G_1} + \frac{1}{G_2}\Big)$$

$$\beta = \frac{G_1(\kappa_2 - 1) - G_2(\kappa_1 - 1)}{G_1(\kappa_2 + 1) + G_2(\kappa_1 + 1)},$$

and $\kappa = 3 - 4\nu$ for plane strain, axisymmetry, and three dimensions; and $\kappa = (3-\nu)/(1+\nu)$ for plane stress. Unlike their analogues in a homogeneous material, K_I and K_{II} are no longer the pure Mode I and Mode II stress intensity factors for an interfacial crack. They are simply the real and imaginary parts of a complex stress intensity factor, whose physical meaning can be understood from the interface traction expressions:

$$(\sigma_{22} + i\sigma_{12})_{\theta=0} = \frac{(K_I + iK_{II})r^{i\varepsilon}}{\sqrt{2\pi r}}, \qquad (\sigma_{23})_{\theta=0} = \frac{K_{III}}{\sqrt{2\pi r}},$$

where r and θ are polar coordinates centered at the crack tip. The bimaterial constant ε is defined as

$$\varepsilon = \frac{1}{2\pi}\ln\frac{1-\beta}{1+\beta}.$$

In this section we describe an interaction integral method (Shih and Asaro, 1988) to extract the individual stress intensity factors for a crack under mixed-mode loading. The method is applicable to cracks in isotropic and anisotropic linear materials.

Interaction integral method

In general, the ***J***-integral for a given problem can be written as

$$J = \frac{1}{8\pi}[K_I B_{11}^{-1} K_I + 2K_I B_{12}^{-1} K_{II} + 2K_I B_{13}^{-1} K_{III} + (\text{terms not involving } K_I)].$$

where I, II, III correspond to $1, 2, 3$ when indicating the components of $\boldsymbol{B}$. We define the ***J***-integral for an auxiliary, pure Mode I, crack-tip field with stress intensity factor k_I, as

$$J_{aux}^I = \frac{1}{8\pi} k_I \cdot B_{11}^{-1} \cdot k_I.$$

Superimposing the auxiliary field onto the actual field yields

$$J_{tot}^I = \frac{1}{8\pi}[(K_I + k_I)B_{11}^{-1}(K_I + k_I) + 2(K_I + k_I)B_{12}^{-1}K_{II} + 2(K_I + k_I)B_{13}^{-1}K_{III} + (\text{terms not involving } K_I \text{ or } k_I)].$$

Since the terms not involving K_I or k_I in J_{tot}^I and ***J*** are equal, the interaction integral can be defined as

$$J_{int}^I = J_{tot}^I - J - J_{aux}^I = \frac{k_I}{4\pi}(B_{11}^{-1}K_I + B_{12}^{-1}K_{II} + B_{13}^{-1}K_{III}).$$

If the calculations are repeated for Mode II and Mode III, a linear system of equations results:

$$J_{int}^\alpha = \frac{k_\alpha}{4\pi} B_{\alpha\beta}^{-1} K_\beta, \quad (\text{no sum on } \alpha = I, II, III),$$

If the k_α are assigned unit values, the solution of the above equations leads to

$$\mathbf{K} = 4\pi \mathbf{B} \cdot \mathbf{J}_{int}\,,$$

where $\mathbf{J}_{int} = \lfloor J_{int}^I, J_{int}^{II}, J_{int}^{III} \rfloor^T$. The calculation of this integral is discussed next.

Based on the definition of the ***J***-integral, the interaction integrals J_{int}^α can be expressed as

$$J_{int}^\alpha = \lim_{\Gamma \to 0} \int_\Gamma \mathbf{n} \cdot \mathbf{M}^\alpha \cdot \mathbf{q}\, d\Gamma$$

with $\mathbf{M}^\alpha$ given as

$$\mathbf{M}^{\alpha} = \boldsymbol{\sigma}\colon\boldsymbol{\varepsilon}^{\alpha}_{aux}\mathbf{I} - \boldsymbol{\sigma}\cdot\left(\frac{\partial\mathbf{u}}{\partial\mathbf{x}}\right)^{\alpha}_{aux} - \boldsymbol{\sigma}^{\alpha}_{aux}\cdot\frac{\partial\mathbf{u}}{\partial\mathbf{x}}.$$

The subscript aux represents three auxiliary pure Mode I, Mode II, and Mode III crack-tip fields for $\alpha = I, II, III$, respectively. Γ is a contour that lies in the normal plane at position ***s*** along the crack front, beginning on the bottom crack surface and ending on the top surface (see Figure 2.16.2–1). The limit $\Gamma \to 0$ indicates that Γ shrinks onto the crack tip.

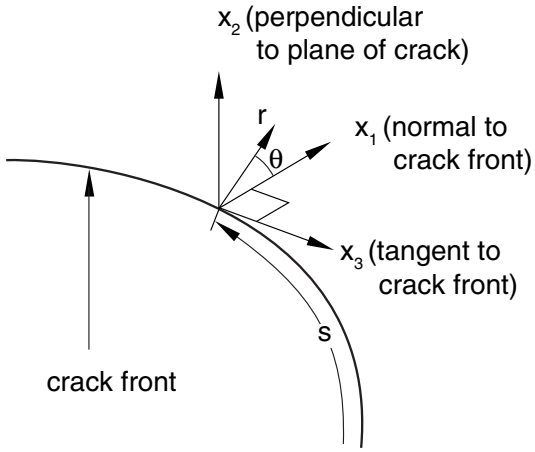


Figure 2.16.2–1 Definition of local orthogonal Cartesian coordinates at the point ***s*** on the crack front; the crack is in the x_1–x_3 plane.

Following the domain integral procedure used in Abaqus/Standard for calculating the ***J***-integral, we define an interaction integral for a virtual crack advance $\lambda(s)$:

$$\bar{J}^{\alpha}_{int} = \int_L J^{\alpha}_{int}(s)\lambda(s)ds = \int_A \lambda(s)\mathbf{n}\cdot\mathbf{M}^{\alpha}\cdot\mathbf{q}dA\,,$$

where ***L*** denotes the crack front under consideration; dA is a surface element on a vanishingly small tubular surface enclosing the crack tip (i.e., $dA = dsd\Gamma$); $\mathbf{n}$ is the outward normal to dA; and $\mathbf{q}$ is the local direction of virtual crack propagation. The integral $\bar{J}^{\alpha}_{int}$ can be calculated by the same domain integral method as that used for calculating the ***J***-integral.

To obtain J^{α}_{int} at each node set ***P*** along the crack front line, λ is discretized with the same interpolation functions as those used in the finite elements along the crack front:

$$\lambda(s) = N^{Q}(s)\lambda^{Q},$$

where $\lambda^{Q} = 1$ at the node set ***P*** and all other λ^{Q} are zero. The result is substituted into the expression for $\bar{J}^{\alpha}_{int}$. Finally, the interaction integral value at each node set ***P*** along the crack front can be calculated as

$$J_{int}^{\alpha P} = \bar{J}_{int}^{\alpha P} / \int_L N^P \, ds.$$

Reference

- “Contour integral evaluation,” Section 11.4.2 of the Abaqus Analysis User’s Guide

2.16.3 T-STRESS EXTRACTION

Product: Abaqus/Standard

The asymptotic expansion of the stress field near a sharp crack in a linear elastic body with respect to r, the distance from the crack tip, is

$$\sigma_{ij} = \frac{K_I}{\sqrt{2\pi r}} f_{ij}^I(\theta) + \frac{K_{II}}{\sqrt{2\pi r}} f_{ij}^{II}(\theta) + \frac{K_{III}}{\sqrt{2\pi r}} f_{ij}^{III}(\theta) + T\delta_{1i}\delta_{1j} + t_1^o\delta_{1i}\delta_{2j} + t_2^o\delta_{2i}\delta_{2j} + t_3^o\delta_{2i}\delta_{3j} + \\ + \sigma_{13}^o\delta_{1i}\delta_{3j} + (\nu T + E\varepsilon_{33})\delta_{3i}\delta_{3j} + O(r^{1/2})$$

(Williams, 1957), where r and θ are the in-plane polar coordinates centered at the crack tip. The local axes are defined so that the 1-axis lies in the plane of the crack at the point of interest on the crack front and is perpendicular to the crack front at this point; the 2-axis is normal to the plane of the crack (and thus is perpendicular to the crack front); and the 3-axis lies tangential to the crack front. $\mathbf{t}^o$ is the surface traction on the crack surfaces at the crack tip, and σ_{13}^o is a constant stress term for σ_{13}. ε_{33} is the extensional strain along the crack front. In plane strain $\varepsilon_{33} = 0$; in plane stress the term $(\nu T + E\varepsilon_{33})\delta_{3i}\delta_{3j}$ vanishes.

The T-stress represents a stress parallel to the crack faces. It is a useful quantity, not only in linear elastic crack analysis but also in elastic-plastic fracture studies.

The T-stress usually arises in the discussions of crack stability and kinking for linear elastic materials. For small amounts of crack growth under Mode I loading, a straight crack path has been shown to be stable when $T < 0$, whereas the path will be unstable and, therefore, will deviate from being straight when $T > 0$ (Cotterell and Rice, 1980). A similar trend has been found in three-dimensional crack propagation studies by Xu, Bower, and Ortiz (1994). Hutchinson and Suo (1992) also showed how the advancing crack path is influenced by the T-stress once cracking initiates under mixed-mode loading. (The direction of crack initiation can be otherwise predicted using the criteria discussed in "Prediction of the direction of crack propagation," Section 2.16.4.)

The T-stress also plays an important role in elastic-plastic fracture analysis, even though the T-stress is calculated from the linear elastic material properties of the same solid containing the crack. The early study of Larsson and Carlsson (1973) demonstrated that the T-stress can have a significant effect on the plastic zone size and shape and that the small plastic zones in actual specimens can be predicted adequately by including the T-stress as a second crack-tip parameter. Some recent investigations (Bilby et al., 1986; Al-Ani and Hancock, 1991; Betegón and Hancock, 1991; Du and Hancock, 1991; Parks, 1992; and Wang, 1991) further indicate that the T-stress can correlate well with the tensile stress triaxiality of elastic-plastic crack-tip fields. The important feature observed in these works is that a negative T-stress can reduce the magnitude of the tensile stress triaxiality (also called the hydrostatic tensile stress) ahead of a crack tip; the more negative the T-stress becomes, the greater the reduction of tensile stress triaxiality. In contrast, a positive T-stress results only in modest elevation of the stress triaxiality. It was found that when the tensile stress triaxiality is high, which is indicated by a positive T-stress, the crack-tip field can be described adequately by the HRR solution (Hutchinson, 1968; Rice and Rosengren, 1968), scaled by a single parameter: the J-integral; that is, J-dominance will exist. When the tensile stress triaxiality is reduced (indicated by the T-stress becoming more negative), the crack-tip fields will quickly deviate from the HRR solution, and J-dominance will be lost

(the asymptotic fields around the crack tip cannot be well characterized by the HRR fields). Thus, using the T-stress (calculated based on the load level and linear elastic material properties) to characterize the triaxiality of the crack-tip stress state and using the J-integral (calculated based on the actual elastic-plastic deformation field) to measure the scale of the crack-tip deformation provides a two-parameter fracture mechanics theory to describe the Mode I elastic-plastic crack-tip stresses and deformation in plane strain or three dimensions accurately over a wide range of crack configurations and loadings.

To extract the T-stress, we use an auxiliary solution of a line load, with magnitude f, applied in the plane of crack propagation and along the crack line:

$$\sigma_{11}^L = \frac{f}{\pi r}\cos^3\theta, \qquad \sigma_{22}^L = \frac{f}{\pi r}\cos\theta\sin^2\theta, \qquad \sigma_{12}^L = \frac{f}{\pi r}\sin\theta\cos^2\theta,$$

$$\sigma_{33}^L = \frac{f}{\pi r}\nu\cos\theta, \qquad \sigma_{13}^L = \sigma_{23}^L = 0.$$

The term $\sigma_{33}^L = 0$ for plane stress.

The interaction integral used is exactly the same as that for extracting the stress intensity factors:

$$I_{int} = \lim_{\Gamma\to 0}\int_\Gamma \mathbf{n}\cdot\mathbf{M}\cdot\mathbf{q}\,d\Gamma,$$

with $\mathbf{M}$ as

$$\mathbf{M} = \boldsymbol{\sigma}\colon\boldsymbol{\varepsilon}_{aux}^L\mathbf{I} - \boldsymbol{\sigma}\cdot\left(\frac{\partial\mathbf{u}}{\partial\mathbf{x}}\right)_{aux}^L - \boldsymbol{\sigma}_{aux}^L\cdot\frac{\partial\mathbf{u}}{\partial\mathbf{x}}.$$

In the limit as $r\to 0$, using the local asymptotic fields,

$$T = \bar{E}\Big[-\frac{I_{int}(s)}{f} + \nu\varepsilon_{33}(s) - \xi\alpha\Delta\Theta\Big] + t_2^o,$$

where $\bar{E} = E$ and $\xi = 1$ for plane stress; $\bar{E} = E/(1-\nu^2)$ and $\xi = 1+\nu$ for plane strain, axisymmetry, and three dimensions; ε_{33} is zero for plane strain and plane stress; α is the thermal expansion coefficient; and $\Delta\Theta$ is the temperature difference.

$I_{int}(s)$ can be calculated by means of the same domain integral method used for J-integral calculation and the stress intensity factor extraction, which has been described in "J-integral evaluation," Section 2.16.1, and "Stress intensity factor extraction," Section 2.16.2. $I_{int}(s)$ is doubled if only half the structure is modeled.

Reference

- "Contour integral evaluation," Section 11.4.2 of the Abaqus Analysis User's Guide

2.16.4 PREDICTION OF THE DIRECTION OF CRACK PROPAGATION

Product: Abaqus/Standard

Various criteria have been proposed to predict the angle at which a pre-existing crack will propagate. Among these criteria are the maximum tangential stress criterion (Erdogan and Sih, 1963), the maximum principal stress criterion (Maiti and Smith, 1983), the maximum energy release rate criterion (Palaniswamy and Knauss, 1978, and Hussain, Pu, and Underwood, 1974), the minimum elastic energy density criterion (Sih, 1973), and the T-criterion (Theocaris, 1982). These criteria predict slightly different angles for the initial crack propagation, but they all have the implication that $K_{II} = 0$ at the crack tip as the crack extends (Cotterell and Rice, 1980). In Abaqus/Standard we provide three criteria for homogeneous, isotropic linear elastic materials: the maximum tangential stress criterion, the maximum energy release rate criterion, and the $K_{II} = 0$ criterion. K_{III} is not taken into account in what follows, since a generally accepted theory for crack propagation with $K_{III} \neq 0$ remains to be developed.

Maximum tangential stress criterion

The near-crack-tip stress field for a homogeneous, isotropic linear elastic material is given by

$$\sigma_{\theta\theta} = \frac{1}{\sqrt{2\pi r}} \cos\frac{1}{2}\theta (K_I \cos^2\frac{1}{2}\theta - \frac{3}{2} K_{II} \sin\theta),$$
$$\tau_{r\theta} = \frac{1}{2\sqrt{2\pi r}} \cos\frac{1}{2}\theta [K_I \sin\theta + K_{II}(3\cos\theta - 1)],$$

where r and θ are polar coordinates centered at the crack tip in a plane orthogonal to the crack front.

The direction of crack propagation can be obtained using either the condition $\partial\sigma_{\theta\theta}/\partial\theta = 0$ or $\tau_{r\theta} = 0$; i.e.,

$$\hat{\theta} = \cos^{-1}\left(\frac{3K_{II}^2 + \sqrt{K_I^4 + 8K_I^2 K_{II}^2}}{K_I^2 + 9K_{II}^2}\right),$$

where the crack propagation angle $\hat{\theta}$ is measured with respect to the crack plane. $\hat{\theta} = 0$ represents the crack propagation in the "straight-ahead" direction. $\hat{\theta} < 0$ if $K_{II} > 0$, while $\hat{\theta} > 0$ if $K_{II} < 0$.

Maximum energy release rate criterion

Consider a crack segment of length a kinking out the plane of the crack at an angle $\hat{\theta}$, as shown in Figure 2.16.4–1. When a is infinitesimally small compared with all other geometric lengths (including the length of the parent crack), the stress intensity factors K_I^k and K_{II}^k at the tip of the putative crack can be expressed as linear combinations of K_I and K_{II}, the stress intensity factors existing prior to kinking for the parent crack:

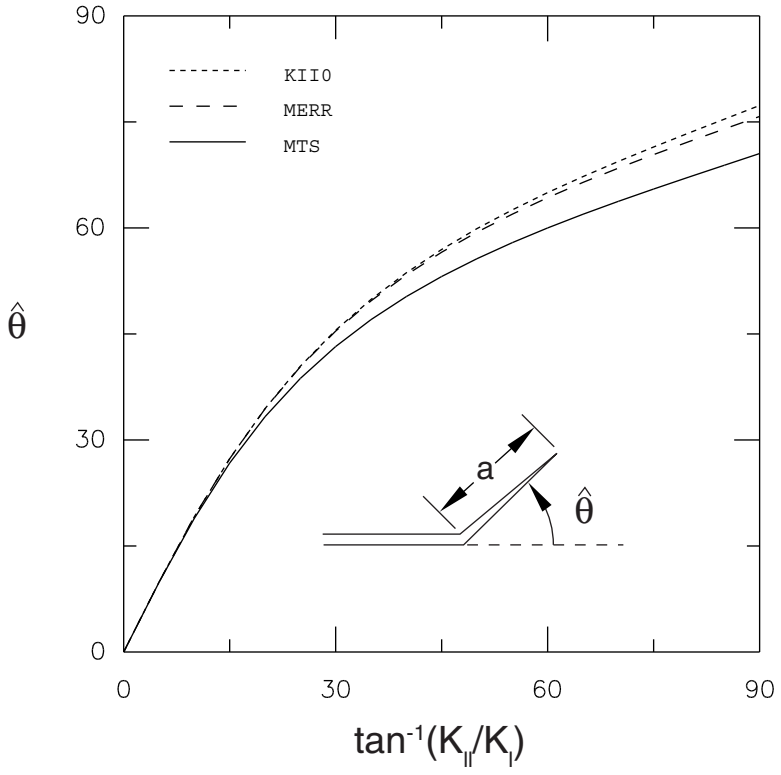


Figure 2.16.4–1 Contour for evaluation of the *J*-integral.

$$K_I^k = c_{11}K_I + c_{12}K_{II},$$
$$K_{II}^k = c_{21}K_I + c_{22}K_{II}.$$

The $\hat{\theta}$-dependences of the coefficients c_{ij} are given by Hayashi and Nemat-Nasser (1981) and by He and Hutchinson (1989).

For the crack segment we also have the relation

$$G^k = \frac{1}{\bar{E}}(K_I^{k\,2} + K_{II}^{k\,2}).$$

The maximum energy release rate criterion postulates that the parent crack initially propagates in the direction that maximizes G^k.

$K_{II} = 0$ criterion

This criterion simply postulates that a crack will initially propagate in the direction that makes $K_{II}^k = 0$.

It can be seen from Figure 2.16.4–1 that the maximum energy release rate criterion and the $K_{II} = 0$ criterion predict nearly coincident crack propagation angles. By comparison, the maximum tangential stress criterion predicts smaller crack propagation angles.

Reference

- "Contour integral evaluation," Section 11.4.2 of the Abaqus Analysis User's Guide

2.17 Stress linearization

- “Stress linearization,” Section 2.17.1

2.17.1 STRESS LINEARIZATION

Product: Abaqus/CAE

Stress linearization is the separation of stresses through a section into constant membrane, linear bending, and nonlinearly varying peak stresses. The capability for calculating linearized stresses is available in the Visualization module of Abaqus/CAE; it is most commonly used for two-dimensional axisymmetric models. See Chapter 52, "Calculating linearized stresses," of the Abaqus/CAE User's Guide for detailed information on how to obtain linearized stresses; the computational methods used by Abaqus are discussed here.

Computing the stress components

Linearized stress components are computed using user-defined sections traversing a finite element model structure. Stress values are extracted at regular intervals along the defined section, and integration is performed numerically using the extracted stress values. Membrane, bending, and peak stress values are computed. These stresses are defined as follows:

Membrane stress

The constant portion of the normal stress such that a pure moment acts on a plane after the membrane stress is subtracted from the total stress.

Bending stress

The variable portion of the normal stress equal to the equivalent linear stress or, where no peak stresses exist, equal to the total stress minus the membrane stress.

Peak stress

The portion of the normal stress that exists after the membrane and bending stresses are subtracted from the total stress.

For the best results, the endpoints of the section should be chosen so that the section is normal to the interior and exterior surfaces of the model. This orientation minimizes problems with shear stresses since they will be approximately zero at the ends of the line (Kroenke, 1973).

Three-dimensional structures

The membrane values of the stress components are computed using the following equation:

$$\boldsymbol{\sigma}^m = \frac{1}{t} \int_{-t/2}^{t/2} \boldsymbol{\sigma} \, dx,$$

where

$\boldsymbol{\sigma}^m$ is the membrane value of stress,

t is the thickness of the section,

σ is the stress along the path, and

x is the coordinate along the path.

The linear bending values of the stress components are computed using the following equations:

$$\sigma_B^b = -\sigma_A^b = \frac{6}{t^2} \int_{-t/2}^{t/2} \sigma \cdot x \cdot dx,$$

where σ_A^b and σ_B^b are the bending values of the stress at point A and point B (the endpoints of the section; see Figure 2.17.1–1).

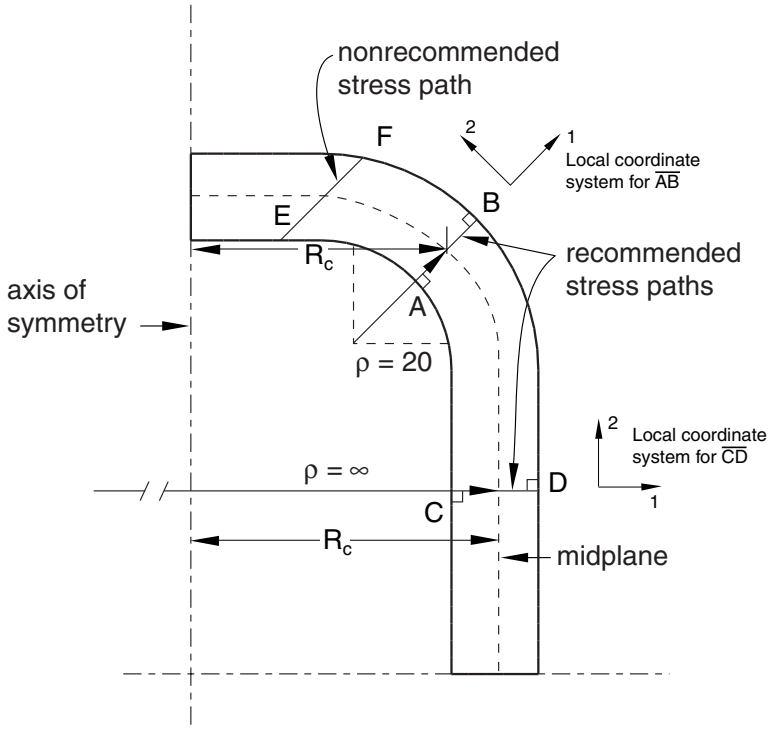


Figure 2.17.1–1 Recommended stress paths.

The integration is performed numerically. Assuming the path between point A and point B is divided uniformly into n intervals, the integrals are evaluated as follows:

$$\sigma^m = \frac{1}{2n} \sum_{j=1}^{n} (\sigma_j + \sigma_{j+1})$$

and

$$\boldsymbol{\sigma}_A^b = -\frac{3}{nt}\sum_{j=1}^{n}(\boldsymbol{\sigma}_j x_j + \boldsymbol{\sigma}_{j+1} x_{j+1}),$$

where $\boldsymbol{\sigma}_j$ is the stress at point $\boldsymbol{j}$ along the path.

Axisymmetric structures

The derivation of the above equations is similar for the axisymmetric case, except for the fact that the neutral axis is shifted radially outward. Separate expressions are obtained for the stresses in the thickness, meridional, and hoop directions. In Abaqus/CAE these are represented as local directions 1, 2, and 3, respectively (see Figure 2.17.1–2).

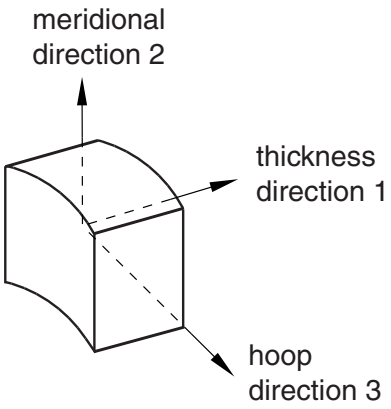


Figure 2.17.1–2 Stress directions.

Meridional stress

The meridional stresses are computed using the following relations. The meridional force per unit circumferential length is

$$F_y = \int_{-t/2}^{t/2} \sigma_y \frac{R}{R_c} dx,$$

where

σ_y is the stress in the meridional direction,
$\boldsymbol{R}$ is the radius of the point being integrated, and
$R_c = \frac{1}{2}(R_A + R_B)$ is the mean circumferential radius.

The meridional membrane stress σ_y^m is obtained by dividing F_y through the thickness:

$$\sigma_y^m = \frac{F_y}{t} = \frac{1}{R_c t} \int_{-t/2}^{t/2} \sigma_y R dx.$$

The numerical scheme used to compute the meridional membrane stress is

$$\sigma_y^m = \frac{1}{2nR_c} \sum_{j=1}^{n} (\sigma_{yj} R_j + \sigma_{y(j+1)} R_{j+1}),$$

where

σ_{yj} is the meridional stress component at point ***j*** along the path and

R_j is the radius at point ***j*** along the path.

To compute the meridional bending stresses, we first need to compute the distance from the center surface to the neutral surface for meridional bending. That distance, x_f, is equal to

$$x_f = \frac{t^2 \cdot \cos\phi}{12 \cdot R_c},$$

where ϕ is the angle between the thickness direction and the radial direction.

The meridional bending moment per unit circumferential length is defined as

$$M_y = \int_{-t/2}^{t/2} (x - x_f) \sigma_y \frac{R}{R_c} dx,$$

and the moment of inertia for meridional bending is given by

$$I_y = \int_{-t/2}^{t/2} (x - x_f)^2 \frac{R}{R_c} dx = \frac{1}{12} t^3 - x_f^2 t.$$

Hence, the meridional bending stresses at the endpoints A and B are obtained with

$$\sigma_{yA}^b = \frac{M_y (x_A - x_f)}{I_y}, \quad \sigma_{yB}^b = \frac{M_y (x_B - x_f)}{I_y}.$$

The numerical scheme used to compute the meridional bending moment is

$$M_y = \frac{t}{2nR_c} \sum_{j=1}^{n} \Big[\sigma_{yj} R_j (x_j - x_f) + \sigma_{y(j+1)} R_{j+1} (x_{j+1} - x_f) \Big].$$

Hoop stress

The hoop membrane stress σ_h^m is obtained with

$$\sigma_h^m = \frac{F_h}{t} = \frac{1}{t}\int_{-t/2}^{t/2} \sigma_h \left(1 + \frac{x}{\rho}\right) dx,$$

where

σ_h is the hoop stress and

ρ is the radius of curvature of the midsurface of the section in the meridional plane.

The hoop bending stresses at the endpoints are obtained using the relations

$$\sigma_{hA}^b = \frac{M_h(x_A - x_h)}{I_h}, \quad \sigma_{hB}^b = \frac{M_h(x_B - x_h)}{I_h},$$

where

$$x_h = \frac{t^2}{12\rho}$$

is the distance from the center surface to the neutral surface for hoop bending,

$$M_h = \int_{-t/2}^{t/2} (x - x_h)\sigma_h \left(1 + \frac{x}{\rho}\right) dx$$

is the hoop bending moment per unit meridional length, and

$$I_h = \int_{-t/2}^{t/2} (x - x_h)^2 \left(1 + \frac{x}{\rho}\right) dx = \frac{1}{12}t^3 - x_h^2 t$$

is the moment of inertia for circumferential bending.

The numerical scheme to compute the circumferential membrane stress is

$$\sigma_h^m = \frac{1}{2n\rho}\sum_{j=1}^{n}\Big[\sigma_{hj}(\rho + x_j) + \sigma_{h(j+1)}(\rho + x_{j+1})\Big],$$

and the bending moment is computed with the summation

$$M_h = \frac{t}{2n\rho}\sum\Big[\sigma_{h(j)}(\rho + x_j)(x_j - x_h) + \sigma_{h(j+1)}(\rho + x_{j+1})(x_{j+1} - x_h)\Big].$$

Thickness stress

The thickness stress does not transfer any forces or moments. Typically, the stress arises due to applied external pressures and thermal expansion effects, and there is no obvious preferred method for determining "membrane" and "bending" stresses. Hence, we choose the thickness "membrane" stress as the average thickness stress:

$$\sigma_x^m = \frac{1}{t}\int_{-t/2}^{t/2} \sigma_x dx = \frac{1}{2n}\sum_{j=1}^{n}(\sigma_{x(j)} + \sigma_{x(j+1)}).$$

We choose the thickness "bending" stress such that the sum of the thickness membrane and bending stresses at the endpoints A and B is equal to the total stress at these points:

$$\sigma_{xA}^b = \sigma_{xA} - \sigma_x^m, \quad \sigma_{xB}^b = \sigma_{xB} - \sigma_x^m,$$

and interpolate the bending stress linearly between these values. This is a reasonable assumption, since the thickness surface stresses are determined by the applied pressure and should not have a strong "peak" stress contribution.

Shear stress

The membrane shear stress in the meridional plane is computed in the same way as the meridional membrane stress:

$$\sigma_{xy}^m = \frac{1}{t}\int_{-t/2}^{t/2} \sigma_{xy}\frac{R}{R_c}dx,$$

where σ_{xy} is the shear stress along the path.

The shear stress distribution is assumed to be parabolic and equal to zero at the ends. Hence, the bending shear stresses are set to 0.0. The numerical scheme used to compute the membrane shear stress is:

$$\sigma_{xy}^m = \frac{1}{2nR_c}\sum_{j=1}^{n}\Big(\sigma_{xy(j)}R_j + \sigma_{xy(j+1)}R_{j+1}\Big).$$

Curvature correction

The equations used when performing stress linearization in axisymmetric structures include the in-plane and out-of-plane radius of curvature of the stress line section. By default, "curvature correction" is turned on for axisymmetric structures and turned off for nonaxisymmetric structures. Abaqus/CAE allows the inclusion of these curvature correction terms in the equations for nonaxisymmetric structures when

computing the S22, S33, and S12 components. The numerical scheme is identical to that used when performing stress linearization in axisymmetric structures. The user must select a coordinate system in which to specify the curvature correction terms. An error message is generated when the *x*-axis of the coordinate system is normal to the stress line.

The user can turn off the curvature correction for axisymmetric structures, in which case the linear stress components are computed using the equations for three-dimensional structures.

Computing the local coordinate system

Stress linearization requires the results to be transformed from the global coordinate system to a local coordinate system defined by the stress line.

Axisymmetric case

The computation of the local coordinate system is a trivial procedure when performed for axisymmetric stress linearization. The transformation matrix in this case will be

$$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Three-dimensional case

When performing three-dimensional stress linearization, the local *x*-axis will be defined by the stress line. The local *y*- and *z*-axes are computed by a series of cross products. This procedure is shown below.

The vector between points (X_1, Y_1, Z_1) and (X_2, Y_2, Z_2) is

$$\mathbf{x} = \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix} = \begin{bmatrix} X_2 - X_1 \\ Y_2 - Y_1 \\ Z_2 - Z_1 \end{bmatrix}.$$

Assuming the local *y*-axis lies in the plane of the local *x*-axis and the global *Y*-axis, the local *z*-axis is defined by

$$\begin{aligned} \mathbf{z} &= \mathbf{x} \times \mathbf{y} \\ &= \det \begin{bmatrix} i & j & k \\ \Delta X & \Delta Y & \Delta Z \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} -\Delta Z \\ 0 \\ \Delta X \end{bmatrix}. \end{aligned}$$

Therefore, the local *y*-axis will be

$$\begin{aligned} \mathbf{y} &= \mathbf{z} \times \mathbf{x} \\ &= \det \begin{bmatrix} i & j & k \\ -\Delta Z & 0 & \Delta X \\ \Delta X & \Delta Y & \Delta Z \end{bmatrix} = \begin{bmatrix} -\Delta X \Delta Y \\ \Delta X^2 + \Delta Z \\ -\Delta Y \Delta Z \end{bmatrix}. \end{aligned}$$

After normalization, the above three vectors can be combined to create the transformation matrix.

Reference

- Chapter 52, “Calculating linearized stresses,” of the Abaqus/CAE User’s Guide

2.18 Design sensitivity analysis

- "Design sensitivity analysis," Section 2.18.1

2.18.1 DESIGN SENSITIVITY ANALYSIS

Product: Abaqus/Design

Abaqus/Design supports design sensitivity analysis (DSA) for static stress and frequency problems. DSA provides derivatives of certain response quantities with respect to specified input quantities. These derivatives are known as *sensitivities*. The responses available for DSA are a subset of the list of Abaqus output variables and are known as *design responses*; the specified input quantities are known as *design parameters*. Quantities that are functions of design parameters are referred to as being design dependent. The DSA theory is presented from the perspective of computing the required derivatives analytically, first for static stress analysis and then for frequency analysis. At the end of each section an alternative numerical approach based on this theory is discussed.

DSA for static stress analysis

The equations pertaining to DSA can be derived based on a total displacement formulation or an incremental displacement formulation. The total displacement formulation is intended for history-independent problems, where the current state of the problem depends only on the total displacements. The incremental formulation is intended for history-dependent problems, where the current state of the problem depends on the state at the beginning of the increment and the incremental displacements.

Total displacement DSA formulation for nonlinear equilibrium problems

Let $\boldsymbol{R}$ and $\boldsymbol{P}$ be the numbers of design responses and design parameters, respectively. Let each response ϕ_r, $r = 1, \ldots, R$, be a function of design parameters h_p, $p = 1, \ldots, P$ and depend on them both explicitly and via the displacement field represented here by the nodal displacement vector u^N (see the definition of finite element interpolation in "Procedures: overview and basic equations," Section 2.1.1),

$$\phi_r = \phi_r\big(u^N(h_p), h_p\big).$$

The dependence $u^N(h_p)$ is only implicit; i.e., it is implied only by the design dependence of coefficients in the equilibrium equation system whose solution is u^N.

Assume that we have solved an equilibrium problem defined by Equation 2.1.1–2 at the end of an increment and that we have the converged solution u^N as well as values of all responses. Sensitivity of a response ϕ_r with respect to design parameter h_p is defined as

$$\frac{d\phi_r}{dh_p} = \frac{\partial \phi_r}{\partial h_p} + \frac{\partial \phi_r}{\partial u^N}\,\frac{du^N}{dh_p}\ . \tag{2.18.1–1}$$

All but one quantity in the above equation can be determined explicitly given the equilibrium solution. The only unknown is du^N/dh_p; to compute it, an additional system of equations has to be solved.

Rewrite Equation 2.1.1–2 in the form

$$F^N(u^M) = 0, \tag{2.18.1–2}$$

where

$$F^N = -\int_{V^0} \boldsymbol{\beta}_N : \boldsymbol{\tau}^c \, dV^0 + \int_S \mathbf{N}_N^T \cdot \mathbf{t} \, dS + \int_V \mathbf{N}_N^T \cdot \mathbf{f} \, dV.$$

All the quantities in the above equation are assumed to depend on design parameters h_p explicitly or via displacement field u^N. Differentiation of the above two equations with respect to design parameters leads to the following equation:

$$K_{MN} \frac{du^N}{dh_p} = -\frac{\partial F^M}{\partial h_p}, \tag{2.18.1–3}$$

in which

$$K_{MN} = \frac{\partial F^M}{\partial u^N}$$

is the tangent stiffness (Jacobian) matrix defined in Equation 2.1.1–4 and $\partial F^M / \partial h_p$ is an explicitly determinable quantity. Substituting Equation 2.18.1–3 into Equation 2.18.1–1, we obtain

$$\frac{d\phi_r}{dh_p} = \frac{\partial \phi_r}{\partial h_p} - \frac{\partial \phi_r}{\partial u^N} K_{NM}^{-1} \frac{\partial F^M}{\partial h_p}, \tag{2.18.1–4}$$

which is the solution of the total displacement DSA problem.

The DSA algorithm used in Abaqus is known as the *direct differentiation method* (DDM) and consists of the following operations. After the converged equilibrium solution is obtained, the three arrays $\partial \phi_r / \partial h_p$, $\partial \phi_r / \partial u^M$, and $\partial F^M / \partial h_p$ have to be computed in an element-by-element manner. $\partial F^M / \partial h_p$ is often called the pseudoload since it becomes the right-hand side of the DSA problem. The final DSA solution is obtained by solving the system of Equation 2.18.1–3 for each $p = 1, \ldots, P$ with respect to the unknown vectors of nodal displacement sensitivity du^N / dh_p. The displacement sensitivities are then substituted into Equation 2.18.1–1 to compute $d\phi_r / dh_p$.

The coefficient matrix K_{MN} used in the DSA computations is simply the last tangent stiffness matrix used in the equilibrium iterative algorithm. At the stage of the DSA computations this matrix is still available in the decomposed form and can be retrieved easily to perform the back substitutions for the DSA right-hand-side vectors. This makes the DSA module a very efficient add-on to the equilibrium analysis enabling sensitivity computations at a relatively low cost.

Incremental displacement DSA formulation for history-dependent equilibrium problems

The formulation of DSA presented above provides a brief introduction to the way DSA is implemented in Abaqus; however, due to some simplifications, the discussion is not relevant to a large number of nonlinear mechanical problems, especially those involving history-dependent behavior of the structure

modeled. The main difficulty in such problems is that many quantities necessary to compute the residual F^N in Equation 2.18.1–2 or to define design responses do not lend themselves to be expressed as functions of total displacement u^N. Rather, at each time increment they are functions of certain state variables at the beginning of the increment (referred to as the time instant t) and of the incremental displacements, Δu^N:

$$F^N = F^N\big(\boldsymbol{\alpha}^t(h_p), \Delta u^N(h_p), h_p\big),$$

$$\phi_r = \phi_r\big(\boldsymbol{\alpha}^t(h_p), \Delta u^N(h_p), h_p\big);$$

see, for example, Kleiber et al. (1997). The notation $\boldsymbol{\alpha}^t$ stands for a set of state variables $\boldsymbol{\alpha}$ that may include tensors (stress, back stress, etc.) as well as scalar quantities (equivalent plastic strain, etc.) defined for a particular material point at time t. Some responses may also depend directly on the displacement u^N, and the beginning-of-the-increment value of u^N will, generally, also enter into the set $\boldsymbol{\alpha}^t$.

In such a case Equation 2.18.1–4 takes the following form:

$$\frac{d\phi_r}{dh_p} = \frac{D\phi_r}{Dh_p} - \frac{\partial \phi_r}{\partial \Delta u^M} K^{-1}_{MN} \frac{DF^N}{Dh_p}, \tag{2.18.1–5}$$

where

$$\frac{D(\cdot)}{Dh_p} \overset{\text{def}}{=} \frac{\partial(\cdot)}{\partial h_p} + \frac{\partial(\cdot)}{\partial \boldsymbol{\alpha}^t} \frac{d\boldsymbol{\alpha}^t}{dh_p}$$

denotes the *explicit design derivative* of a quantity $(\cdot)$.

The fundamental difference, from the point of view of the DSA solution algorithm, between the total and incremental approach is that in the latter case all state variables $\boldsymbol{\alpha}$ effectively become additional, or *internal*, design responses, whose sensitivities must be computed and updated at the end of each time increment to proceed with the DSA in the next increment. The number of such internal responses may be significant with obvious effects both on the computational time and memory requirement.

The DSA solution procedure is similar to that in the total displacement approach. After the equilibrium computations are complete, the arrays of explicit design derivatives $D\phi_r/Dh_p$, DF^M/Dh_p (the pseudoload), and the derivatives with respect to displacements $\partial\phi_r/\partial\Delta u^M$ are assembled in the element loop. The set of design responses ϕ_r, $r = 1, \ldots, R$, includes in this case all the scalars and tensor components of $\boldsymbol{\alpha}$. In the direct differentiation method the following system of equations is solved for each design parameter h_p:

$$K_{MN} \frac{d\Delta u^N}{dh_p} = -\frac{DF^M}{Dh_p},$$

and the solution vectors are substituted into Equation 2.18.1–5.

Computational approach

The derivatives required for DSA can be computed analytically or numerically. In the analytical approach the finite element equations are differentiated exactly, following the theory described in the previous sections. This approach is difficult to implement, but it is efficient and yields exact sensitivities. In the numerical approach some or all of the required derivatives are computed using the finite difference technique. The numerical approach can be further subdivided into the *overall* or *global* finite difference approach and the *semi-analytic* approach. In the global finite difference approach the response sensitivities with respect to a particular design parameter are obtained by perturbing that design parameter a number of times (depending on the finite difference technique) and performing an entire equilibrium analysis for each perturbation. The responses are retained for each analysis and then differenced to obtain the response sensitivities. This approach is computationally expensive since an entire equilibrium problem must be solved for each perturbation, but it is easily implemented. The semi-analytic approach is used in Abaqus and can be viewed as a compromise between the analytic and global finite difference approaches. In the semi-analytic approach the DSA element vectors are obtained by differencing; but, like the analytic approach, the DSA solution is obtained by back-substitution against K_{MN}. The advantage of the semi-analytic approach is that it is much easier to implement than the analytic approach and much more efficient than the global finite difference approach. The details of this method are described in the following paragraphs.

The objective of the semi-analytic approach is to compute the DSA vectors DF^M/Dh_p and $d\phi_r/dh_p$ numerically by finite differencing. For simplicity, assume that the finite difference technique is central difference such that for a given function $A(x)$, the derivative of $\boldsymbol{A}$ with respect to $\boldsymbol{x}$ is

$$\frac{dA}{dx} = \frac{A(x+\delta x) - A(x-\delta x)}{2\delta x},$$

where δx is the perturbation of $\boldsymbol{x}$.

For generality, consider the history-dependent case. To approximate the explicit design derivatives of F^M, the incremental displacement is held constant while a positive perturbation δh_p is applied to each design parameter h_p. In this way perturbed values of F^M are obtained as

$$F^M + \delta F^M = F^M(\boldsymbol{\alpha}^t(h_p + \delta h_p), \Delta u(h_p), h_p + \delta h_p).$$

The change in the state corresponding to a perturbation in the design parameters is approximated by

$$\boldsymbol{\alpha}^t(h_p + \delta h_p) = \boldsymbol{\alpha}^t(h_p) + \frac{d\boldsymbol{\alpha}^t}{dh_p}\delta h_p.$$

The above process is repeated for a negative perturbation ($-\delta h_p$), after which the results are differenced to arrive at the explicit design derivative DF^M/Dh_p.

Once the (incremental) displacement sensitivities are found, the response sensitivities $d\phi_r/dh_p$ can be obtained using

$$\phi_r + \delta\phi_r = \phi_r(\boldsymbol{\alpha}^t(h_p + \delta h_p), \Delta u^N(h_p + \delta h_p), h_p + \delta h_p),$$

where

$$\Delta u^N(h_p + \delta h_p) = \Delta u^N(h_p) + \frac{d\Delta u^N}{dh_p}\delta h_p.$$

The process is repeated for a negative perturbation of h_p, and the results are differenced.

The finite difference interval must be chosen carefully. If the interval is too small, round-off or *cancellation* errors occur due to loss of precision during the differencing operations. On the other hand, if the interval is too large, *truncation* errors may occur. Truncation errors arise from the fact that differencing formulas are based on truncated Taylor series expansions. Abaqus will automatically choose a perturbation size that provides the best compromise between cancellation and truncation errors.

DSA for frequency analysis

This discussion will build upon the concepts and terminology described above, so it is recommended that the previous section be read first. The discussion below is divided into two sections depending on the characteristics of the eigenvalue problem: distinct eigenvalues and repeated eigenvalues.

Distinct eigenvalue case

The theory presented below assumes that all the eigenvalues are distinct (i.e., no repeated eigenvalues). If this is not the case, further manipulations are required to obtain the eigenvalue and eigenvector sensitivities corresponding to the repeated eigenvalues, and the following equations for the sensitivities will be incorrect. The repeated eigenvalue case is considered in the next section.

Performing a frequency analysis means solving the following eigenvalue problem (see "Eigenvalue extraction," Section 2.5.1):

$$(K^{NM} - \lambda M^{NM})\phi_N = 0, \tag{2.18.1–6}$$

where λ represents the eigenvalues, ϕ_N represents the eigenvectors, and M^{NM} is the mass matrix. In addition, the eigenvectors are scaled such that either

$$\phi^N_{\alpha_{\max}} = 1 \tag{2.18.1–7}$$

or

$$\phi^N_\alpha M^{NM} \phi^M_\alpha = 1 \tag{2.18.1–8}$$

for each mode. The default is the first scaling scheme. To obtain eigenvalue and eigenvector sensitivities, first differentiate Equation 2.18.1–6 with respect to design parameter h_p to obtain the following equation:

$$\left(K^{NM} - \lambda_\alpha M^{NM}\right)\frac{d\phi^M_\alpha}{dh_p} = \left(\frac{DK^{NM}}{Dh_p} - \lambda_\alpha \frac{DM^{NM}}{Dh_p}\right)\phi^M_\alpha - \frac{d\lambda_\alpha}{dh_p} M^{NM}\phi^M_\alpha, \tag{2.18.1–9}$$

where α represents a particular mode number.

Pre-multiplying by ϕ_α^N, making use of Equation 2.18.1–6, and manipulating the result gives the eigenvalue sensitivities:

$$\frac{d\lambda_\alpha}{dh_p} = \frac{\phi_\alpha^N \left(\frac{DK^{NM}}{Dh_p} - \lambda_\alpha \frac{DM^{NM}}{Dh_p} \right) \phi_\alpha^M}{\phi_\alpha^N M^{NM} \phi_\alpha^M}. \tag{2.18.1–10}$$

Except for the mass and stiffness derivatives, all quantities in this equation are known once the eigenvalue problem has been solved.

Repeated eigenvalue case

This section outlines the formulation used to obtain eigenvalue sensitivities for repeated eigenvalues. Further information can be found in the papers by Mills-Curran (1988) and Shaw (1991). When an eigenvalue λ repeats R times, the eigenvectors $\mathbf{\Phi} = \lfloor \phi_1 \dots \phi_R \rfloor$ associated with λ are linearly independent but not unique—any linear combination of these eigenvectors is also an eigenvector. Because of this non-uniqueness, the eigenvectors may not be continuous or differentiable in the design parameter. However, a set of R eigenvectors $\mathbf{Z} = \lfloor \mathbf{z}_1 \dots \mathbf{z}_R \rfloor$ that are continuous and differentiable can be obtained by an orthogonal transformation:

$$\mathbf{Z} = \mathbf{\Phi A}, \tag{2.18.1–11}$$

where $\mathbf{A}^T\mathbf{A} = \mathbf{I}$, and $\mathbf{A}$ is to be determined. Replacing the eigenvectors ϕ with the eigenvectors $\mathbf{z}$ in Equation 2.18.1–9 and premultiplying by $\mathbf{\Phi}^T$ yields in matrix notation:

$$\left(\mathbf{B} - \frac{d\mathbf{\Lambda}}{dh_p} \mathbf{C} \right) \mathbf{A} = \mathbf{0}, \tag{2.18.1–12}$$

where $\mathbf{\Lambda} = \lambda \mathbf{I}$, $d\mathbf{\Lambda}/dh_p$ is a diagonal matrix,

$$\mathbf{B} = \mathbf{\Phi}^T \left(\frac{D\mathbf{K}}{Dh_p} - \lambda \frac{D\mathbf{M}}{Dh_p} \right) \mathbf{\Phi}, \tag{2.18.1–13}$$

and

$$\mathbf{C} = \mathbf{\Phi}^T \mathbf{M} \mathbf{\Phi}.$$

Equation 2.18.1–12 is recognized as an $R \times R$ reduced eigenvalue problem whose R eigenvalues $d\mathbf{\Lambda}/dh_p$ are the eigenvalue sensitivities associated with the repeated eigenvalue λ and whose eigenvectors are $\mathbf{A}$. N sensitivities are obtained for the single eigenvalue λ. The physical interpertation of this is that a perturbation in the design parameter may cause the original repeated mode to branch into as many as N distinct modes (imagine a beam with a circular cross-section perturbed into an elliptical section;

the repeated modes associated with the original symmetry of the section now split into distinct modes associated with the minor and major axes of the ellipse).

Computational approach

A semi-analytic approach is used to compute the eigenvalue sensitivities. The basic idea of this approach, as outlined in the section on static DSA, is to compute some of the required intermediate derivatives using finite differencing. In the context of DSA for frequency procedures this means that the derivatives of the mass and stiffness matrices are computed using finite differencing.

Choosing the finite difference interval

Abaqus uses a heuristic algorithm to automatically determine the perturbation sizes to be used in the differencing scheme. The objective of the algorithm is to select the perturbation sizes that lead to accurately computed derivatives. This is done on an element to element basis, so it is possible for a given design parameter that the perturbation size for one element will be different from that of another element. The selection of the perturbation sizes is based on the behavior of a scalar metric s_p for each design parameter h_p. This metric must be both convenient and representative of the terms that are numerically differenced. It is chosen as follows:

- *Static steps.* For static steps the scalar metric is chosen as the norm of the element pseudoload:

$$s_p = ||(dF_{el}^M/dh_p)|_{\Delta u_{el}^N \neq \Delta u_{el}^N(h_p)}||.$$

  Since an accurate pseudoload is necessary to obtain an accurate sensitivity solution, the norm of the pseudoload is an appropriate choice for s_p. This choice is also convenient since the pseudoload has to be computed in any case.

- *Frequency steps.* For frequency steps the scalar metric is chosen as the projection of the matrix $\left(\frac{DK^{NM}}{Dh_p} - \lambda_\alpha \frac{DM^{NM}}{Dh_p}\right)$ onto an eigenvector $\bar{\phi}_\alpha^N$:

$$s_p = \bar{\phi}_\alpha^N \left(\frac{DK^{NM}}{Dh_p} - \lambda_\alpha \frac{DM^{NM}}{Dh_p}\right) \bar{\phi}_\alpha^M.$$

  The choice of $\bar{\phi}_\alpha^N$ depends on whether a given mode α has a distinct eigenvalue or is associated with a repeated eigenvalue.

  If mode α has a distinct eigenvalue, $\bar{\phi}_\alpha^N$ is taken as ϕ_α^N. Consequently, s_p becomes simply the numerator of Equation 2.18.1–10. Therefore, s_p is a direct measure of the magnitude of the eigenvalue sensitivity and is also convenient since this term already must be calculated to obtain the eigenvalue sensitivity.

  Unlike the distinct eigenvalue case, the sensitivities of a repeated eigenvalue cannot be treated independently. The sensitivities of a repeated eigenvalue are extracted simultaneously from the same reduced eigenvalue system, and this system is obtained by numerical differencing (recall Equation 2.18.1–12 and Equation 2.18.1–13). Consequently a single perturbation size (for each design parameter) must be used to obtain all sensitivities of a repeated eigenvalue. To calculate the single perturbation size, a single scalar metric is obtained by taking $\bar{\phi}_\alpha^N$ as the sum of the

eigenvectors associated with the repeated eigenvalue. The calculation of s_p is similar to the calculation of the matrix $\mathbf{B}$ (the only term in the reduced eigenvalue system obtained by numerical differencing); therefore, this choice s_p is both representative of the repeated eigenvalue case and involves differencing calculations that are already being done to obtain the reduced eigenvalue system.

The basic idea of the algorithm is compute the scalar metric s_p for a range of perturbation sizes δh_p^I, $I = 1, 2, \ldots$, which vary by orders of magnitude. For each δh_p^I, a corresponding s_p^I is computed. The relative change in s_p between consecutive perturbation sizes, calculated as $e_p^I = |s_p^I - s_p^{I-1}|/s_p^I$, is used to measure how close a perturbation size is to optimum. In this range of perturbation sizes the one that yields the smallest relative change, denoted as e_p^a, is identified as the best perturbation size, δh_p^a. If e_p^a is not less than a specified tolerance, the range of perturbation sizes is expanded (up to a certain limit), and the testing continues. It is important to realize that e_p^I is not used directly in assessing the accuracy of the numerical differentiation but rather is intended as a means for determining the optimum perturbation size. Thus, a tighter tolerance on e_p^a causes the algorithm to expend more effort in finding an optimum perturbation size but does not guarantee more accurate sensitivities.

Reference

- "Design sensitivity analysis," Section 19.1.1 of the Abaqus Analysis User's Guide

3. Elements

Overview 3.1
Continuum elements 3.2
Infinite elements 3.3
Membrane and truss elements 3.4
Beam elements 3.5
Shell elements 3.6
Rebar 3.7
Hydrostatic fluid calculations 3.8
Special-purpose elements 3.9

3.1 Overview

- “Element library: overview,” Section 3.1.1

3.1.1 ELEMENT LIBRARY: OVERVIEW

The Abaqus element library provides a complete geometric modeling capability. For this reason any combination of elements can be used to make up the model. Sometimes multi-point constraints are required for application of the necessary kinematic relations to form the model (for example, to model part of a shell surface with solid elements and part with shell elements or to model a pipe elbow with a mixture of beam and shell elements).

All elements use numerical integration to allow complete generality in material behavior. Shell and beam element properties can be defined as general section behaviors, or each cross-section of the element can be integrated numerically, so that nonlinear response can be tracked accurately when needed. A composite layered section can be specified, with different materials at different heights through the section. Some special elements (such as line springs) use an approximate analytical solution to model nonlinear behavior.

All of the elements in Abaqus are formulated in a global Cartesian coordinate system except the axisymmetric elements, which are formulated in terms of r–z coordinates. In almost all elements, primary vector quantities (such as displacements $\mathbf{u}$ and rotations ϕ) are defined in terms of nodal values with scalar interpolation functions. For example, in elements with a two-dimensional topology the interpolation can be written as

$$\mathbf{u}(g,h) = N^N(g,h)\mathbf{u}^N,$$

where the interpolation functions $N^N(g,h)$ are written in terms of the parametric coordinates g and h. In most element types the same parametric interpolation is used for the coordinate vector $\mathbf{x}$:

$$\mathbf{x}(g,h) = N^N(g,h)\mathbf{x}^N.$$

Such *isoparametric* elements are guaranteed to be able to represent all rigid body modes and homogeneous deformation modes exactly, a necessary condition for convergence to the exact solution as the mesh is refined.

All elements in Abaqus are integrated numerically. Hence, the virtual work integral as described in "Nonlinear solution methods in Abaqus/Standard," Section 2.2.1, will be replaced by a summation:

$$\int_V \boldsymbol{\sigma} : \delta\mathbf{D}\, dV \rightarrow \sum_{i=1}^{n} \boldsymbol{\sigma}_i : \delta\mathbf{D}_i\, V_i,$$

where n is the number of integration points in the element and V_i is the volume associated with integration point i. Abaqus will use either "full" or "reduced" integration. For full integration the number of integration points is sufficient to integrate the virtual work expression exactly, at least for linear material behavior. All triangular and tetrahedral elements in Abaqus use full integration. Reduced integration can be used for quadrilateral and hexahedral elements; in this procedure the number of integration points is sufficient to

integrate exactly the contributions of the strain field that are one order less than the order of interpolation. The (incomplete) higher-order contributions to the strain field present in these elements will not be integrated.

The advantage of the reduced integration elements is that the strains and stresses are calculated at the locations that provide optimal accuracy, the so-called Barlow points (Barlow, 1976). A second advantage is that the reduced number of integration points decreases CPU time and storage requirements. The disadvantage is that the reduced integration procedure can admit deformation modes that cause no straining at the integration points. These zero-energy modes make the element rank-deficient and cause a phenomenon called "hourglassing," where the zero energy mode starts propagating through the mesh, leading to inaccurate solutions. This problem is particularly severe in first-order quadrilaterals and hexahedra. To prevent these excessive deformations, an additional artificial stiffness is added to the element. In this so-called hourglass control procedure, a small artificial stiffness is associated with the zero-energy deformation modes. This procedure is used in many of the solid and shell elements in Abaqus.

Most fully integrated solid elements are unsuitable for the analysis of (approximately) incompressible material behavior. The reason for this is that the material behavior forces the material to deform (approximately) without volume changes. Fully integrated solid element meshes, and in particular lower-order element meshes, do not allow such deformations (other than purely homogeneous deformation). For that reason Abaqus uses "selectively reduced" integration in these elements: reduced integration is used for the volume strain and full integration for the deviatoric strains. As a consequence the lower-order elements give an acceptable performance for approximately incompressible behavior. For fully incompressible material behavior, another complication occurs: the bulk modulus and, hence, the stiffness matrix becomes infinitely large. For this case a mixed (hybrid) formulation is required, where the displacement field is augmented with a hydrostatic pressure field. In this formulation only the inverse of the bulk modulus appears, and, consequently, the contribution to the operator matrix vanishes. The hydrostatic pressure field plays the role of a Lagrange multiplier field enforcing the incompressibility constraints.

Abaqus/Standard also provides elements for multifield problems. Examples are the pore pressure elements used for the analysis of porous solids with fluid diffusion, coupled temperature-displacement elements that couple heat transfer with stress analysis, and piezoelectric elements that couple electrical conduction with stress analysis. In these multifield elements the scalar variable (such as the temperature) is usually interpolated with different scalar functions as the displacement field; i.e.,

$$T(g,h) = M^N(g,h)\,T^N,$$

where $M^N(g,h)$ may differ from $N^N(g,h)$. The coupling of the fields will generally occur at the integration points; for example, in coupled temperature-displacement elements the coupling is due to temperature-dependent mechanical properties and heat generation is due to inelastic work. Finally, Abaqus offers a complete set of diffusion elements to analyze conductive and convective heat transfer. In these elements only temperatures appear as nodal degrees of freedom. The temperatures are interpolated with essentially the same interpolation function, $M^N(g,h)$, as used in the coupled temperature-displacement elements.

Reference

- "Element library: overview," Section 27.1.1 of the Abaqus Analysis User's Guide

3.2 Continuum elements

- “Solid element overview,” Section 3.2.1
- “Solid element formulation,” Section 3.2.2
- “Hybrid incompressible solid element formulation,” Section 3.2.3
- “Solid isoparametric quadrilaterals and hexahedra,” Section 3.2.4
- “Continuum elements with incompatible modes,” Section 3.2.5
- “Triangular, tetrahedral, and wedge elements,” Section 3.2.6
- “Generalized plane strain elements,” Section 3.2.7
- “Axisymmetric elements,” Section 3.2.8
- “Axisymmetric elements allowing nonlinear bending,” Section 3.2.9

3.2.1 SOLID ELEMENT OVERVIEW

Abaqus contains a library of solid elements for two-dimensional and three-dimensional applications. The two-dimensional elements allow modeling of plane and axisymmetric problems and include extensions to generalized plane strain (when the model exists between two planes that may move with respect to each other, providing thickness direction strain that may vary with position in the plane of the model but is constant with respect to thickness position). The material description of three-dimensional solid elements may include several layers of different materials, in different orientations, for the analysis of laminated composite solids. A set of nonlinear elements for asymmetric loading of axisymmetric models is also available, and linear infinite elements in two and three dimensions can be used to model unbounded domains.

The solid element library includes isoparametric elements: quadrilaterals in two dimensions and "bricks" (hexahedra) in three dimensions. These isoparametric elements are generally preferred for most cases because they are usually the more cost-effective of the elements that are provided in Abaqus. They are offered with first- and second-order interpolation and are described in detail in "Solid isoparametric quadrilaterals and hexahedra," Section 3.2.4. For practical reasons it is sometimes not possible to use isoparametric elements throughout a model; for example, some commercial mesh generators use automatic meshing techniques that rely on triangulation to fill arbitrarily shaped regions. Because of these needs Abaqus includes triangular, tetrahedron, and wedge elements. For most cases it is recommended that these elements be only used to fill in awkward parts of the mesh and, in particular, that well-shaped isoparametric elements be used in any critical region (such as an area where the strain must be predicted accurately). The isoparametric elements can also be degenerated to make simpler shapes. Generally the elements written for those particular geometries are preferred to this method. The exception to this rule occurs in cases where singularities are to be modeled (such as in fracture mechanics applications), since the degenerate second-order isoparametric elements can provide a $1/\sqrt{r}$ singularity through the use of the "quarter point" technique (placing the midside nodes 1/4 of the distance along the side from the node at the singularity instead of at the middle point of the side).

Solid elements are provided with first-order (linear) and second-order (quadratic) interpolation, and the user must decide which approach is more appropriate for the application. Some guidelines are as follows. Standard first-order elements are essentially constant strain elements: the isoparametric forms can provide more than constant strain response, but the higher-order content of the solutions they give is generally not accurate and, thus, of little value. The "incompatible mode" elements, described in "Continuum elements with incompatible modes," Section 3.2.5, are from the user's perspective lower-order elements but have internal degrees of freedom that enable the element to represent almost all linear strain patterns. These elements can represent certain important linear strain fields exactly: the most important field is the one due to bending. The second-order elements are capable of representing all possible linear strain fields. Thus, in the case of elliptic problems—problems for which the governing partial differential equations are elliptic in character, such as elasticity, heat conduction, acoustics, in which smoothness of the solution is assured—much higher solution accuracy per degree of freedom is usually available with the higher-order elements. Therefore, it is generally recommended that the highest-order elements available be used for such cases: in Abaqus this means second-order elements. This observation logically leads to the use of the "hierarchical" finite element technique or "p"-method—refining the model by increasing the interpolation order in the elements in critical regions: this approach is as yet not available in Abaqus.

A case where both incompatible mode elements and second-order elements can be used effectively is the stress analysis of relatively thin members subjected to bending: such problems are often encountered in practical applications. In such cases the strain variation through the thickness must be at least linear, and constant strain (first-order) elements do a poor job of representing this variation. Fully integrated first-order isoparametric elements also suffer from "shear locking" in these geometries: they cannot provide the pure bending solution because they must shear at the numerical integration points to respond with an appropriate kinematic behavior corresponding to the bending. This shearing then locks the element—the response is far too stiff. For the isoparametric elements reduced integration provides a cure for these problems, but at the cost of allowing spurious singular modes ("hourglassing"). The use of second-order elements is a more reliable alternative, because the second-order interpolation naturally contains the linear strain field—one element through the thickness is enough to represent the behavior of a thin component subjected to bending loads quite accurately. Another alternative is formed by the incompatible mode elements: the linear strain field in these elements contains the modes required to solve the bending problem exactly if the elements are rectangular in shape. For a detailed discussion of the performance of Abaqus continuum elements in bending problems, see "Performance of continuum and shell elements for linear analysis of bending problems," Section 2.3.5 of the Abaqus Benchmarks Guide. (It should be remembered, however, that Abaqus offers shell and beam elements that are specifically written for thin geometries: the use of solid elements for such cases should only be considered when beam or shell elements are not practical.)

For all of these reasons the second-order elements are preferred in elliptic applications. The argument is readily extended to higher-order interpolation (cubic, quartic, etc), but the rapid increase in cost per element for higher-order forms means that—even though the accuracy per degree of freedom is higher—the accuracy per computational cost may not be increasing. Practical experience suggests that—except in special cases—little is gained by going beyond the second-order elements, so Abaqus does not offer any higher-order forms.

Many problems of practical interest are not elliptic: localizations arise in one form or another. Plasticity applications are an example—as the solution approaches the limit load, most plasticity models tend toward hyperbolic behavior. This allows discontinuities to occur in the solution—the slip line solutions of classical perfect plasticity theory are plots of the characteristic lines of velocity discontinuities in the hyperbolic equations of the problem. If the finite element solution is to exhibit accuracy, these discontinuities in the gradient field of the solution should be reasonably well modeled. With a fixed mesh that does not use special elements that admit discontinuities in their formulation, this suggests that the lowest-order elements—the first-order elements—are likely to be the most successful, because, for a given number of nodes, they provide the most locations at which some component of the gradient of the solution can be discontinuous (the element edges). This argument is hardly rigorous, but it is, nevertheless, true that first-order elements tend to be preferred for such cases. The incompatible mode elements can represent discontinuities particularly well. They are also able to represent strain localization such as occurs in shear bands. One should realize, however, that better defined shear localization increases the strain magnitude and, hence, tends to increase the number of increments and iterations required for the analysis.

All of the solid elements in Abaqus, except the infinite elements, are written to include finite-strain effects. When these elements are used with a hyperelastic (elastomeric) material definition, the constitutive behavior is calculated directly from the deformation gradient matrix, $\mathbf{F}$. When the elements are used for geometrically nonlinear analysis with any other material definition (at finite strain this means the material

has some inelastic behavior, since all of the elasticity definitions in Abaqus except the hyperelasticity models assume that the elastic strains are small), the strains are calculated as the integral of the rate of deformation,

$$\mathbf{D} = \text{sym}\left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right),$$

with the effects of material rotation with respect to the coordinate system taken into consideration. In all cases the solid elements report stress as the "true" (Cauchy) stress. Unless a local orientation is specified for an element, stress and strain components are given as physical components referred to the global spatial directions. When a local orientation is defined for a solid element, the stress and strain components are given in the user-defined local system: this system rotates with the average material rotation calculated at each material point.

Reference

- "General-purpose continuum elements," Section 28.1 of the Abaqus Analysis User's Guide

3.2.2 SOLID ELEMENT FORMULATION

Products: Abaqus/Standard Abaqus/Explicit

All the solid elements in Abaqus allow for finite strain and rotation in large-displacement analysis. For kinematically linear analysis the strain is defined as

$$\boldsymbol{\varepsilon} = \text{sym}\left(\frac{\partial \mathbf{u}}{\partial \mathbf{X}}\right),$$

where $\mathbf{u}$ is the total displacement and $\mathbf{X}$ is the spatial position of the point under consideration in the original configuration. As discussed in Chapter 1, "Introduction and Basic Equations," this measure of strain is useful only if the strains and rotations are small (all components of the strain and rotation matrices are negligible compared to unity).

For cases where the strains and/or rotations are no longer small, two ways of measuring strain are used in the solid elements in Abaqus. When the hyperelastic or hyperfoam material definition is used with an element, Abaqus internally uses the stretch values calculated directly from the deformation gradient matrix, $\mathbf{F}$, to compute the material behavior. With any other material behavior it is assumed that any elastic strains are small compared to unity, so the appropriate reference configuration for the elasticity is only infinitesimally different from the current configuration and the appropriate stress measure is, therefore, the Cauchy ("true") stress. (More precisely, the appropriate stress measure should be the Kirchhoff stress defined with respect to the elastic reference configuration, but the assumption that this reference configuration and the current configuration are only infinitesimally different makes the Kirchhoff and Cauchy stress measures almost the same: the differences are on the order of the elastic strains compared to unity). The conjugate strain rate to Cauchy stress is the rate of deformation,

$$\mathbf{D} = \text{sym}\left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right),$$

where $\mathbf{v}$ is the velocity at a point and $\mathbf{x}$ are the current spatial coordinates of the point. The strain is, therefore, defined as the integral of the rate of deformation. This integration is nontrivial, particularly in the general case where the principal axes of strain rotate during the deformation. In Abaqus the total strain is constructed by integrating the strain rate approximately over the increment by the central difference algorithm; and, when the strain components are referred to a fixed coordinate basis, the strain at the start of the increment must also be rotated to account for the rigid body rotation that occurs in the increment. This is also done approximately, using the Hughes-Winget (1980) method. This integration algorithm defines the integration of a tensor associated with the material behavior as

$$\mathbf{a}_{t+\Delta t} = \Delta\mathbf{R} \cdot \mathbf{a}_t \cdot \Delta\mathbf{R}^T + \Delta\check{\mathbf{a}}(\Delta\mathbf{D}),$$

where $\mathbf{a}$ is the tensor; $\Delta\check{\mathbf{a}}$ is the increment in the tensor associated with the material's constitutive behavior, and, therefore, dependent on the strain increment, $\Delta\mathbf{D}$, defined by the central difference formula as

$$\Delta\mathbf{D} = \text{sym}\left(\frac{\partial\Delta\mathbf{u}}{\partial\mathbf{x}_{t+\Delta t/2}}\right),$$

where $\mathbf{x}_{t+\Delta t/2} = (1/2)(\mathbf{x}_t + \mathbf{x}_{t+\Delta t})$; and $\Delta\mathbf{R}$ is the increment in rotation, defined by Hughes and Winget as

$$\Delta\mathbf{R} = \left(\mathbf{I} - \frac{1}{2}\Delta\mathbf{W}\right)^{-1} \cdot \left(\mathbf{I} + \frac{1}{2}\Delta\mathbf{W}\right),$$

where $\Delta\mathbf{W}$ is the central difference integration of the rate of spin:

$$\Delta\mathbf{W} = \text{asym}\left(\frac{\partial\Delta\mathbf{u}}{\partial\mathbf{x}_{t+\Delta t/2}}\right).$$

A somewhat different algorithm to calculate $\Delta\mathbf{R}$ is used for the Green-Naghdi rate in Abaqus/Explicit.

For example, the stress is integrated by this method as

$$\boldsymbol{\sigma}_{t+\Delta t} = \Delta\mathbf{R} \cdot \boldsymbol{\sigma}_t \cdot \Delta\mathbf{R}^T + \Delta\check{\boldsymbol{\sigma}}(\Delta\mathbf{D}),$$

where $\Delta\check{\boldsymbol{\sigma}}(\Delta\mathbf{D})$ is the stress increment caused by the straining of the material during this time increment and $\boldsymbol{\sigma}$ is the Cauchy stress. The subscripts t and $t + \Delta t$ refer to the beginning and the end of the increment, respectively.

As shown in "Procedures: overview and basic equations," Section 2.1.1, the contribution of the internal work terms to the Jacobian of the Newton method that is often used in Abaqus/Standard is

$$\int_V \left(d\boldsymbol{\sigma} : \delta\mathbf{D} + \boldsymbol{\sigma} : d\delta\mathbf{D}\right)\, dV, \tag{3.2.2–1}$$

where $d\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}$ are evaluated at the end of the increment.

Using the integration definition above, it can be shown that

$$d\boldsymbol{\sigma}_{t+\Delta t} = d\Delta\mathbf{R} \cdot \Delta\mathbf{R}^T \cdot (\boldsymbol{\sigma}_{t+\Delta t} - \Delta\check{\boldsymbol{\sigma}}) + (\boldsymbol{\sigma}_{t+\Delta t} - \Delta\check{\boldsymbol{\sigma}}) \cdot \Delta\mathbf{R} \cdot d\Delta\mathbf{R}^T + \mathbf{C} : d\Delta\mathbf{D},$$

where $\mathbf{C}$ is the Jacobian matrix of the constitutive model:

$$\mathbf{C} = \frac{\partial\Delta\check{\boldsymbol{\sigma}}}{\partial\Delta\mathbf{D}}.$$

However, rather than computing the tangent matrix for the Newton method on this basis, we approximate this by using

$$d\boldsymbol{\sigma}_{t+\Delta t} = d\mathbf{W} \cdot \boldsymbol{\sigma}_{t+\Delta t} + \boldsymbol{\sigma}_{t+\Delta t} \cdot d\mathbf{W}^T + \mathbf{C} : d\mathbf{D},$$

which yields the Jacobian

$$\int_V \left(\delta\mathbf{D} : \mathbf{C} : d\mathbf{D} - \boldsymbol{\sigma} : \left(2\delta\mathbf{D} \cdot d\mathbf{D} - \frac{\partial \delta\mathbf{v}}{\partial \mathbf{x}}^T \cdot \frac{\partial d\mathbf{v}}{\partial \mathbf{x}} \right) \right) dV.$$

This Jacobian is the tangent stiffness of the rate form of the problem. Experience with practical cases suggests that this approximation provides an acceptable rate of convergence in the Newton iterations in most applications with real materials.

The strain and rotation measures described above are approximations. Probably the most limiting aspect of these approximations is the definition of the rotation increment $\Delta\mathbf{R}$. While this measure does give a representation of the rotation of the material at a point in some average sense (both in Abaqus/Standard and Abaqus/Explicit), it is clear that each of the individual material fibers at a point has a different rotation (unless the material point undergoes rigid body motion only or, as an approximate extension, if the strains at the point are small). This suggests that the formulation described above will not be suitable for applications where the strains and rotations are large and where the material exhibits some form of anisotropic behavior. A common example of such cases is the induction of anisotropy through straining, as in "kinematic hardening" plasticity models. The integration methods described above are not suitable for such material models at large strains (for practical purposes with typical material parameters this means that the solutions will be quite wrong when the strains are greater than 20%–30%). Therefore, the use of the kinematic hardening model in Abaqus at such strain levels is not recommended. There is extensive literature on this subject; for example, see Agah-Tehrani et al. (1986).

Reference

- "Solid (continuum) elements," Section 28.1.1 of the Abaqus Analysis User's Guide

3.2.3 HYBRID INCOMPRESSIBLE SOLID ELEMENT FORMULATION

Products: Abaqus/Standard Abaqus/Explicit

Many problems involve the prediction of the response of almost incompressible materials. This is especially true at large strains, since most solid materials show relatively incompressible behavior under large deformations. In this section we describe the augmented virtual work basis provided in Abaqus/Standard for such cases. The method is described in the context of incompressible elasticity theory, since that is where it is most likely to be used.

When the material response is incompressible, the solution to a problem cannot be obtained in terms of the displacement history only, since a purely hydrostatic pressure can be added without changing the displacements. The nearly incompressible case (that is, when the bulk modulus is much larger than the shear modulus or Poisson's ratio, ν, is greater than 0.4999999) exhibits behavior approaching this limit, in that a very small change in displacement produces extremely large changes in pressure, so that a purely displacement-based solution is too sensitive to be useful numerically (for example, round-off on the computer may cause the method to fail). We remove this singular behavior in the system by treating the pressure stress as an independently interpolated basic solution variable, coupled to the displacement solution through the constitutive theory and the compatibility condition, with this coupling implemented by a Lagrange multiplier. This independent interpolation of pressure stress is the basis of these "hybrid" elements. More precisely, they are "mixed formulation" elements, using a mixture of displacement and stress variables with an augmented variational principle to approximate the equilibrium equations and compatibility conditions. The hybrid elements also remedy the problem of volume strain "locking," which can occur at much lower values of ν (i.e., $\nu = 0.49$). Volume strain locking occurs if the finite element mesh cannot properly represent incompressible deformations. Volume strain locking can be avoided in regular displacement elements by fully or selectively reduced integration, as described in "Solid isoparametric quadrilaterals and hexahedra," Section 3.2.4.

We begin by writing the internal virtual work:

$$\delta W = \int_V \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon}\, dV, \tag{3.2.3–1}$$

where $\delta\boldsymbol{\varepsilon}$ is the virtual strain:

$$\delta\boldsymbol{\varepsilon} = \text{sym}\left(\frac{\partial \delta\mathbf{u}}{\partial \mathbf{x}}\right),$$

where $\delta\mathbf{u}$ is the virtual displacement field, $\boldsymbol{\sigma}$ is the true (Cauchy) stress, V is the current volume, and δW is the virtual work as defined by this equation. See "Equilibrium and virtual work," Section 1.5.1, for a detailed discussion of the virtual work concept.

In a displacement-based formulation the Cauchy stress, $\boldsymbol{\sigma}$, is obtained with the constitutive equations from the deformation, usually in rate form:

$$d\boldsymbol{\sigma} = \mathbf{C} : d\boldsymbol{\varepsilon} + d\mathbf{W} \cdot \boldsymbol{\sigma} - \boldsymbol{\sigma} \cdot d\mathbf{W}, \tag{3.2.3–2}$$

where $\mathbf{C}$ is the "material stiffness matrix" and $d\mathbf{W}$ is the rate of rotation (spin) of the material.

We modify the Cauchy stress by introducing an independent hydrostatic pressure field $\hat{p}$ as follows:

$$\overline{\boldsymbol{\sigma}} = \boldsymbol{\sigma} + (1-\rho)\mathbf{I}(p - \hat{p}), \tag{3.2.3–3}$$

where

$$p = -\frac{1}{3}\text{trace}(\boldsymbol{\sigma})$$

is the hydrostatic pressure stress and ρ is a small number. If ρ was set equal to zero, the hydrostatic component in $\overline{\boldsymbol{\sigma}}$ would be identical to the independent pressure field $\hat{p}$, corresponding to a pure "mixed" formulation. The small nonzero value (10^{-9}) is chosen to avoid equation solver difficulties. This relation is used in incremental form:

$$\overline{\boldsymbol{\sigma}} = \overline{\boldsymbol{\sigma}}^0 + \Delta\boldsymbol{\sigma} + (1-\rho)\mathbf{I}(\Delta p - \Delta\hat{p}), \tag{3.2.3–4}$$

where $\overline{\boldsymbol{\sigma}}^0$ is the modified Cauchy stress at the start of the increment. We use the modified Cauchy stress in the virtual work expression and augment the expression with the Lagrange multiplier enforced constraint $\Delta p - \Delta\hat{p} = 0$:

$$\delta\overline{W} = \int_V [\overline{\boldsymbol{\sigma}} : \delta\boldsymbol{\varepsilon} + J^{-1}\delta\lambda(\Delta p - \Delta\hat{p})]\, dV, \tag{3.2.3–5}$$

with J the volume change ratio (Jacobian) and $\delta\lambda$ a Lagrange multiplier whose interpolation must still be determined. $\Delta\hat{p}$ will be interpolated over each element so that the constraint is satisfied in an integrated (average) sense. Since Δp is the value of the equivalent pressure stress increment computed from the kinematic solution, Equation 3.2.3–4 does not make sense if the material is fully incompressible because then Δp cannot be computed. For the purpose of development we regard the bulk modulus as finite, and we will be able to show that the final formulation approaches a usable limit as we allow the bulk modulus to approach infinity.

For the formulation of the tangent stiffness (the Jacobian), we need to define the rate of change of $\delta\overline{W}$. Therefore, we rewrite the virtual work equation in terms of the reference volume V^0:

$$\delta\overline{W} = \int_{V^0} [J\overline{\boldsymbol{\sigma}} : \delta\boldsymbol{\varepsilon} + \delta\lambda(\Delta p - \Delta\hat{p})]\, dV^0. \tag{3.2.3–6}$$

The rate of change $d\delta\overline{W}$ is then readily obtained as

$$d\delta\overline{W} = \int_{V^0} [dJ\,\overline{\boldsymbol{\sigma}} : \delta\boldsymbol{\varepsilon} + J\,d\overline{\boldsymbol{\sigma}} : \delta\boldsymbol{\varepsilon} + J\,\overline{\boldsymbol{\sigma}} : d\delta\boldsymbol{\varepsilon} + \delta\lambda(dp - d\hat{p}) + d\delta\lambda(\Delta p - \Delta\hat{p})]\, dV^0. \tag{3.2.3–7}$$

We rewrite this expression in terms of the current volume:

$$d\delta\overline{W} = \int_V [d\overline{\boldsymbol{\sigma}} : \delta\boldsymbol{\varepsilon} + d\boldsymbol{\varepsilon} : \mathbf{I}\,\overline{\boldsymbol{\sigma}} : \delta\boldsymbol{\varepsilon} + \overline{\boldsymbol{\sigma}} : d\delta\boldsymbol{\varepsilon} + J^{-1}\delta\lambda(dp - d\hat{p}) + J^{-1}d\delta\lambda(\Delta p - \Delta\hat{p})]\, dV, \quad (3.2.3\text{–}8)$$

where we used the identity $J^{-1}dJ = \mathbf{I} : d\boldsymbol{\varepsilon}$.

The rate of the modified stress follows from Equation 3.2.3–4 and the constitutive equations:

$$d\overline{\boldsymbol{\sigma}} = \mathbf{C} : d\boldsymbol{\varepsilon} + (1-\rho)\mathbf{I}(dp - d\hat{p}) + d\mathbf{W}\cdot\overline{\boldsymbol{\sigma}} - \overline{\boldsymbol{\sigma}}\cdot d\mathbf{W}, \quad (3.2.3\text{–}9)$$

where

$$dp = -\frac{1}{3}\text{trace}(d\boldsymbol{\sigma}) = -\frac{1}{3}\mathbf{I} : \mathbf{C} : d\boldsymbol{\varepsilon};$$

and we used the fact that $d\mathbf{W}\cdot\overline{\boldsymbol{\sigma}} - \overline{\boldsymbol{\sigma}}\cdot d\mathbf{W} = d\mathbf{W}\cdot\boldsymbol{\sigma} - \boldsymbol{\sigma}\cdot d\mathbf{W}$, since $\overline{\boldsymbol{\sigma}}$ and $\boldsymbol{\sigma}$ differ only in the hydrostatic part. Substituting these expressions into the expression for the rate of virtual work yields

$$\begin{aligned} d\delta\overline{W} = \int_V \Big\{ & \delta\boldsymbol{\varepsilon} : \mathbf{C} : d\boldsymbol{\varepsilon} + \delta\boldsymbol{\varepsilon} : \overline{\boldsymbol{\sigma}}\,\mathbf{I} : d\boldsymbol{\varepsilon} - \frac{1}{3}(1-\rho)\delta\boldsymbol{\varepsilon} : \mathbf{I}\,\mathbf{I} : \mathbf{C} : d\boldsymbol{\varepsilon} \\ & - (1-\rho)\delta\boldsymbol{\varepsilon} : \mathbf{I}\, d\hat{p} - \frac{1}{3}J^{-1}\delta\lambda\,\mathbf{I} : \mathbf{C} : d\boldsymbol{\varepsilon} - J^{-1}\delta\lambda\, d\hat{p} \\ & + J^{-1}d\delta\lambda(\Delta p - \Delta\hat{p}) + \overline{\boldsymbol{\sigma}} : d\delta\boldsymbol{\varepsilon} + \delta\boldsymbol{\varepsilon} : (d\mathbf{W}\cdot\overline{\boldsymbol{\sigma}} - \overline{\boldsymbol{\sigma}}\cdot d\mathbf{W})\Big\} dV. \end{aligned} \quad (3.2.3\text{–}10)$$

It remains to choose $\delta\lambda$. To get a symmetric expression for the rate of virtual work, we choose

$$\delta\lambda = (1-\rho)J\left(\frac{1}{3K}\mathbf{I} : \mathbf{C} : \delta\boldsymbol{\varepsilon} - \mathbf{I} : \delta\boldsymbol{\varepsilon} + \frac{1}{K}\delta\hat{p}\right), \quad (3.2.3\text{–}11)$$

where

$$K = \frac{1}{9}\mathbf{I} : \mathbf{C} : \mathbf{I} \quad (3.2.3\text{–}12)$$

is the (instantaneous) bulk modulus. This is a suitable choice for $\delta\lambda$, because the (independent) term proportional to $\delta\hat{p}$ ensures that the modified incremental pressure field, $\Delta\hat{p}$, is properly constrained to the incremental pressure, Δp. If we assume that the volumetric moduli $\mathbf{I} : \mathbf{C}$ and $\boldsymbol{K}$ change slowly with strain and ignore changes in volume, we can write for the second variation $d\delta\lambda$:

$$d\delta\lambda = (1-\rho)J\left(\frac{1}{3K}\mathbf{I} : \mathbf{C} - \mathbf{I}\right) : d\delta\boldsymbol{\varepsilon}. \quad (3.2.3\text{–}13)$$

Hence, we find for the virtual work expression:

$$\delta\overline{W} = \int_V \left[\tilde{\boldsymbol{\sigma}} : \delta\boldsymbol{\varepsilon} + (1-\rho)(\Delta p - \Delta\hat{p})\frac{1}{K}\delta\hat{p}\right] dV, \quad (3.2.3\text{–}14)$$

where

$$\tilde{\boldsymbol{\sigma}} = \overline{\boldsymbol{\sigma}} + (1-\rho)(\Delta p - \Delta\hat{p})\left(\frac{1}{3K}\mathbf{I}:\mathbf{C}-\mathbf{I}\right). \tag{3.2.3–15}$$

For the rate of change of virtual work we find

$$\begin{aligned} d\delta\overline{W} = \int_V \Big\{ & \delta\boldsymbol{\varepsilon}:\mathbf{C}:d\boldsymbol{\varepsilon} - \frac{1}{9K}(1-\rho)\delta\boldsymbol{\varepsilon}:\mathbf{C}^T:\mathbf{I}\,\mathbf{I}:\mathbf{C}:d\boldsymbol{\varepsilon} + \delta\boldsymbol{\varepsilon}:\overline{\boldsymbol{\sigma}}\,\mathbf{I}:d\boldsymbol{\varepsilon} \\ & - \frac{1}{3K}(1-\rho)\left(\delta\hat{p}\,\mathbf{I}:\mathbf{C}:d\boldsymbol{\varepsilon} + \delta\boldsymbol{\varepsilon}:\mathbf{C}^T:\mathbf{I}\,d\hat{p}\right) - \frac{1}{K}(1-\rho)\,\delta\hat{p}\,d\hat{p} \\ & + \tilde{\boldsymbol{\sigma}}:d\delta\boldsymbol{\varepsilon} + \overline{\boldsymbol{\sigma}}:(\delta\boldsymbol{\varepsilon}\cdot d\mathbf{W} - d\mathbf{W}\cdot\delta\boldsymbol{\varepsilon})\Big\}dV. \end{aligned} \tag{3.2.3–16}$$

The initial stress term can be approximated by

$$\tilde{\boldsymbol{\sigma}}:(d\delta\boldsymbol{\varepsilon} + \delta\boldsymbol{\varepsilon}\cdot d\mathbf{W} - d\mathbf{W}\cdot\delta\boldsymbol{\varepsilon}),$$

which can be written as

$$\tilde{\boldsymbol{\sigma}}:\left[\left(\frac{\partial\delta\mathbf{u}}{\partial\mathbf{x}}\right)^T\cdot\frac{\partial d\mathbf{u}}{\partial\mathbf{x}} - 2\delta\boldsymbol{\varepsilon}\cdot d\boldsymbol{\varepsilon}\right],$$

so that the final expression for the rate of virtual work becomes

$$\begin{aligned} d\delta\overline{W} = \int_V \Big\{ & \delta\boldsymbol{\varepsilon}:\mathbf{C}:d\boldsymbol{\varepsilon} - \frac{1}{9K}(1-\rho)\,\delta\boldsymbol{\varepsilon}:\mathbf{C}^T:\mathbf{I}\,\mathbf{I}:\mathbf{C}:d\boldsymbol{\varepsilon} + \delta\boldsymbol{\varepsilon}:\overline{\boldsymbol{\sigma}}\,\mathbf{I}:d\boldsymbol{\varepsilon} \\ & - \frac{1}{3K}(1-\rho)\left(\delta\hat{p}\,\mathbf{I}:\mathbf{C}:d\boldsymbol{\varepsilon} + \delta\boldsymbol{\varepsilon}:\mathbf{C}^T:\mathbf{I}\,d\hat{p}\right) - \frac{1}{K}(1-\rho)\,\delta\hat{p}\,d\hat{p} \\ & + \tilde{\boldsymbol{\sigma}}:\left[\left(\frac{\partial\delta\mathbf{u}}{\partial\mathbf{x}}\right)^T\cdot\frac{\partial d\mathbf{u}}{\partial\mathbf{x}} - 2\delta\boldsymbol{\varepsilon}\cdot d\boldsymbol{\varepsilon}\right]\Big\}dV. \end{aligned} \tag{3.2.3–17}$$

The asymmetric term $\delta\boldsymbol{\varepsilon}:\overline{\boldsymbol{\sigma}}\,\mathbf{I}:d\boldsymbol{\varepsilon}$ is significant only if large volume changes occur. Hence, the term is ignored except for material models with volumetric plasticity, such as the (capped) Drucker-Prager model and the Cam-clay model. For these models the constitutive matrix $\mathbf{C}$ is usually asymmetric anyway so that the addition of this nonsymmetric term does not affect the cost of the analysis. It was assumed in the expression for $d\delta\lambda$ that the (volumetric) moduli change only slowly with strain. This is not the case for material models with volumetric plasticity, in which these moduli can change abruptly. This may lead to slow convergence or even convergence failures. Failures usually occur only in higher-order elements, since in lower-order elements $\Delta p - \Delta\hat{p}$ approaches zero at every point and the error in $d\delta\lambda$ has no impact.

Reference

- "Solid (continuum) elements," Section 28.1.1 of the Abaqus Analysis User's Guide

3.2.4 SOLID ISOPARAMETRIC QUADRILATERALS AND HEXAHEDRA

Products: Abaqus/Standard Abaqus/Explicit

The library of solid elements in Abaqus contains first- and second-order isoparametric elements. The first-order elements are the 4-node quadrilateral for plane and axisymmetric analysis and the 8-node brick for three-dimensional cases. The library of second-order isoparametric elements includes "serendipity" elements: the 8-node quadrilateral and the 20-node brick, and a "full Lagrange" element, the 27-node (variable number of nodes) brick. The term "serendipity" refers to the interpolation, which is based on corner and midside nodes only. In contrast, the full Lagrange interpolation uses product forms of the one-dimensional Lagrange polynomials to provide the two- or three-dimensional interpolation functions.

All these isoparametric elements are available with full or reduced integration. Gauss integration is almost always used with second-order isoparametric elements because it is efficient and the Gauss points corresponding to reduced integration are the Barlow points (Barlow, 1976) at which the strains are most accurately predicted if the elements are well-shaped.

The three-dimensional brick elements can also be used for the analysis of laminated composite solids. Several layers of different material, in different orientations, can be specified in each solid element. The material layers or lamina can be stacked in any of the three isoparametric coordinates, parallel to opposite faces of the master element (Figure 3.2.4–1). These elements use the same interpolation functions as the homogeneous elements, but the integration takes the variation of material properties in the stacking direction into account.

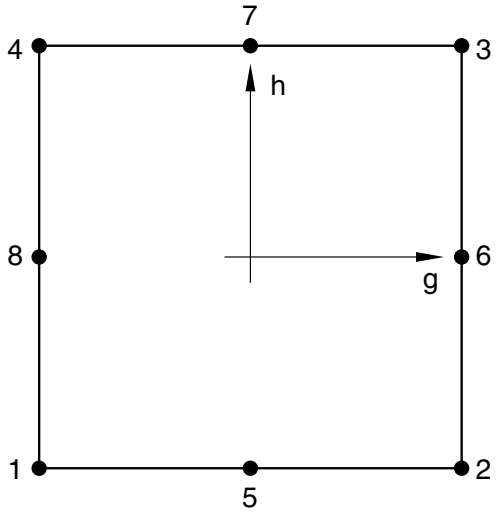
Hybrid pressure-displacement versions of these elements are provided for use with incompressible and nearly incompressible constitutive models (see "Hybrid incompressible solid element formulation," Section 3.2.3, and "Hyperelastic material behavior," Section 4.6.1, for a detailed discussion of the formulations used).

Interpolation

Isoparametric interpolation is defined in terms of the isoparametric element coordinates g, h, r shown in Figure 3.2.4–1. These are material coordinates, since Abaqus is a Lagrangian code. They each span the range -1 to $+1$ in an element. The node numbering convention used in Abaqus for isoparametric elements is also shown in Figure 3.2.4–1. Corner nodes are numbered first, followed by the midside nodes for second-order elements. The interpolation functions are as follows.

First-order quadrilateral:

$$u = \frac{1}{4}(1-g)(1-h)u_1 + \frac{1}{4}(1+g)(1-h)u_2 + \frac{1}{4}(1+g)(1+h)u_3 + \frac{1}{4}(1-g)(1+h)u_4$$



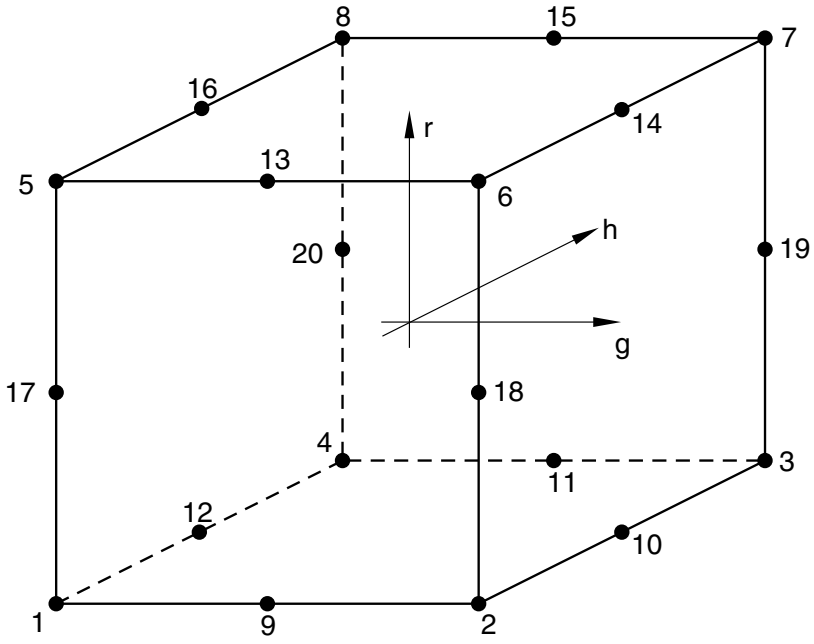


Figure 3.2.4–1 Isoparametric master elements.

Second-order quadrilateral:

$$\begin{aligned} u = &-\frac{1}{4}(1-g)(1-h)(1+g+h)u_1 - \frac{1}{4}(1+g)(1-h)(1-g+h)u_2 \\ &-\frac{1}{4}(1+g)(1+h)(1-g-h)u_3 - \frac{1}{4}(1-g)(1+h)(1+g-h)u_4 \\ &+\frac{1}{2}(1-g)(1+g)(1-h)u_5 + \frac{1}{2}(1-h)(1+h)(1+g)u_6 \\ &+\frac{1}{2}(1-g)(1+g)(1+h)u_7 + \frac{1}{2}(1-h)(1+h)(1-g)u_8 \end{aligned}$$

First-order brick:

$$\begin{aligned} u = &\frac{1}{8}(1-g)(1-h)(1-r)u_1 + \frac{1}{8}(1+g)(1-h)(1-r)u_2 \\ &+\frac{1}{8}(1+g)(1+h)(1-r)u_3 + \frac{1}{8}(1-g)(1+h)(1-r)u_4 \\ &+\frac{1}{8}(1-g)(1-h)(1+r)u_5 + \frac{1}{8}(1+g)(1-h)(1+r)u_6 \\ &+\frac{1}{8}(1+g)(1+h)(1+r)u_7 + \frac{1}{8}(1-g)(1+h)(1+r)u_8 \end{aligned}$$

20-node brick:

$$\begin{aligned} u = &-\frac{1}{8}(1-g)(1-h)(1-r)(2+g+h+r)u_1 - \frac{1}{8}(1+g)(1-h)(1-r)(2-g+h+r)u_2 \\ &-\frac{1}{8}(1+g)(1+h)(1-r)(2-g-h+r)u_3 - \frac{1}{8}(1-g)(1+h)(1-r)(2+g-h+r)u_4 \\ &-\frac{1}{8}(1-g)(1-h)(1+r)(2+g+h-r)u_5 - \frac{1}{8}(1+g)(1-h)(1+r)(2-g+h-r)u_6 \\ &-\frac{1}{8}(1+g)(1+h)(1+r)(2-g-h-r)u_7 - \frac{1}{8}(1-g)(1+h)(1+r)(2+g-h-r)u_8 \\ &+\frac{1}{4}(1-g)(1+g)(1-h)(1-r)u_9 + \frac{1}{4}(1-h)(1+h)(1+g)(1-r)u_{10} \\ &+\frac{1}{4}(1-g)(1+g)(1+h)(1-r)u_{11} + \frac{1}{4}(1-h)(1+h)(1-g)(1-r)u_{12} \\ &+\frac{1}{4}(1-g)(1+g)(1-h)(1+r)u_{13} + \frac{1}{4}(1-h)(1+h)(1+g)(1+r)u_{14} \\ &+\frac{1}{4}(1-g)(1+g)(1+h)(1+r)u_{15} + \frac{1}{4}(1-h)(1+h)(1-g)(1+r)u_{16} \\ &+\frac{1}{4}(1-r)(1+r)(1-g)(1-h)u_{17} + \frac{1}{4}(1-r)(1+r)(1+g)(1-h)u_{18} \\ &+\frac{1}{4}(1-r)(1+r)(1+g)(1+h)u_{19} + \frac{1}{4}(1-r)(1+r)(1-g)(1+h)u_{20} \end{aligned}$$

Integration of homogeneous solids

All the isoparametric solid elements are integrated numerically. Two schemes are offered: "full" integration and "reduced" integration. For the second-order elements Gauss integration is always used because it is efficient and it is especially suited to the polynomial product interpolations used in these elements. For the first-order elements the single-point reduced-integration scheme is based on the "uniform strain formulation": the strains are not obtained at the first-order Gauss point but are obtained as the (analytically calculated) average strain over the element volume. The uniform strain method, first published by Flanagan and Belytschko (1981), ensures that the first-order reduced-integration elements pass the patch test and attain the accuracy when elements are skewed. Alternatively, the "centroidal strain formulation," which uses 1-point Gauss integration to obtain the strains at the element center, is also available for the 8-node brick elements in Abaqus/Explicit for improved computational efficiency. The differences between the uniform strain formulation and the centroidal strain formulation can be shown as follows:

For the 8-node brick elements the interpolation function given above can be rewritten as

$$\mathbf{u} = N^I(g,h,r)\mathbf{u}^I \text{ sum on } I.$$

The isoparametric shape functions N^I can be written as

$$N^I(g,h,r) = \frac{1}{8}\Sigma^I + \frac{1}{4}g\Lambda_1^I + \frac{1}{4}h\Lambda_2^I + \frac{1}{4}r\Lambda_3^I + \frac{1}{2}hr\Gamma_1^I + \frac{1}{2}gr\Gamma_2^I + \frac{1}{2}gh\Gamma_3^I + \frac{1}{2}ghr\Gamma_4^I,$$

where

$$\Sigma^I = [+1,+1,+1,+1,+1,+1,+1,+1],$$

$$\Lambda_1^I = [-1,+1,+1,-1,-1,+1,+1,-1],$$

$$\Lambda_2^I = [-1,-1,+1,+1,-1,-1,+1,+1],$$

$$\Lambda_3^I = [-1,-1,-1,-1,+1,+1,+1,+1],$$

$$\Gamma_1^I = [+1,+1,-1,-1,-1,-1,+1,+1],$$

$$\Gamma_2^I = [+1,-1,-1,+1,-1,+1,+1,-1],$$

$$\Gamma_3^I = [+1, -1, +1, -1, +1, -1, +1, -1],$$

$$\Gamma_4^I = [-1, +1, -1, +1, +1, -1, +1, -1],$$

and the superscript I denotes the node of the element. The last four vectors, Γ_α^I (α has a range of four), are the hourglass base vectors, which are the deformation modes associated with no energy in the 1-point integration element but resulting in a nonconstant strain field in the element.

In the uniform strain formulation the gradient matrix $\mathbf{B}^I$ is defined by integrating over the element as

$$B_i^I = \frac{1}{V_{e\ell}} \int_{V_{e\ell}} N_i^I(g, h, r)\, dV_{e\ell},$$

$$N_i^I(g, h, r) = \frac{\partial N^I}{\partial x_i},$$

where $V_{e\ell}$ is the element volume and i has a range of three.

In the centroidal strain formulation the gradient matrix $\mathbf{B}^I$ is simply given as

$$B_i^I = N_i^I(0, 0, 0),$$

which has the following antisymmetric property:

$$B_i^1 = -B_i^7,$$

$$B_i^3 = -B_i^5,$$

$$B_i^2 = -B_i^8,$$

$$B_i^4 = -B_i^6.$$

It can be seen from the above that the centroidal strain formulation reduces the amount of effort required to compute the gradient matrix. This cost savings also extends to strain and element nodal force calculations because of the antisymmetric property of the gradient matrix. However, the centroidal strain formulation is less accurate when the elements are skewed. For two-dimensional plane elements and hexahedron elements in a parallelepiped configuration the uniform strain approach is identical to the centroidal strain approach.

Full integration means that the Gauss scheme chosen will integrate the stiffness matrix of an element with uniform material behavior exactly if the Jacobian of the mapping from the isoparametric coordinates to the physical coordinates is constant throughout the element; this means that opposing element sides

or faces in three-dimensional elements must be parallel and, in the case of the second-order elements, that the midside nodes must be at the middle of the element sides. If the element does not satisfy these conditions, full integration is not exact because some of the terms in the stiffness are of higher order than those that are integrated exactly by the Gauss scheme chosen. Such inaccuracy in the integration does not appear to be detrimental to the element's performance. As will be discussed below, full integration in Abaqus in first-order elements includes a further approximation and is more accurately called "selectively reduced integration."

Reduced integration usually means that an integration scheme one order less than the full scheme is used to integrate the element's internal forces and stiffness. Superficially this appears to be a poor approximation, but it has proved to offer significant advantages. For second-order elements in which the isoparametric coordinate lines remain orthogonal in the physical space, the reduced-integration points have the Barlow point property (Barlow, 1976): the strains are calculated from the interpolation functions with higher accuracy at these points than anywhere else in the element. For first-order elements the uniform strain method yields the exact average strain over the element volume. Not only is this important with respect to the values available for output, it is also significant when the constitutive model is nonlinear, since the strains passed into the constitutive routines are a better representation of the actual strains.

Reduced integration decreases the number of constraints introduced by an element when there are internal constraints in the continuum theory being modeled, such as incompressibility, or the Kirchhoff transverse shear constraints if solid elements are used to analyze bending problems. In such applications fully integrated elements will "lock"—they will exhibit response that is orders of magnitude too stiff, so the results they provide are quite unusable. The reduced-integration version of the same element will often work well in such cases.

Finally, reduced integration lowers the cost of forming an element; for example, a fully integrated, second-order, 20-node three-dimensional element requires integration at 27 points, while the reduced-integration version of the same element only uses 8 points and, therefore, costs less than 30% of the fully integrated version. This cost savings is especially significant in cases where the element formation costs dominate the overall costs, such as problems with a relatively small wavefront and problems in which the constitutive models require lengthy calculations. The deficiency of reduced integration is that, except in one dimension and in axisymmetric geometries modeled with higher than first-order elements, the element stiffness matrix will be rank deficient. This most commonly exhibits itself in the appearance of singular modes ("hourglass modes") in the response. These are nonphysical response modes that can grow in an unbounded way unless they are controlled. The reduced-integration second-order serendipity interpolation elements in two dimensions—the 8-node quadrilaterals—have one such mode, but it is benign because it cannot propagate in a mesh with more than one element. The second-order three-dimensional elements with reduced integration have modes that can propagate in a single stack of elements. Because these modes rarely cause trouble in the second-order elements, no special techniques are used in Abaqus to control them.

In contrast, when reduced integration is used in the first-order elements (the 4-node quadrilateral and the 8-node brick), hourglassing can often make the elements unusable unless it is controlled. In Abaqus the artificial stiffness method and the artificial damping method given in Flanagan and Belytschko (1981) are used to control the hourglass modes in these elements. The artificial damping method is available only

for the solid and membrane elements in Abaqus/Explicit. To control the hourglass modes, the hourglass shape vectors, γ_α^I, are defined:

$$\gamma_\alpha^I = \Gamma_\alpha^I - \frac{1}{V_{e\ell}} B_i^I x_i^J \Gamma_\alpha^J,$$

which are different from the hourglass base vectors, Γ_α^I. It is essential to use the hourglass shape vectors rather than the hourglass base vectors to calculate the hourglass-resisting forces to ensure that these forces are orthogonal to the linear displacement field and the rigid body field (see Flanagan and Belytschko (1981) for details). However, using the hourglass base vectors to calculate the hourglass-resisting forces may provide computational speed advantages. Therefore, for the 8-node brick elements Abaqus/Explicit provides the option to use the hourglass base vectors in calculating the hourglass-resisting forces. For hexahedron elements in a parallelepiped configuration the hourglass shape vectors are identical to the hourglass base vectors.

The hourglass control methods of Flanagan and Belytschko (1981) are generally successful for linear and mildly nonlinear problems but may break down in strongly nonlinear problems and, therefore, may not yield reasonable results. Success in controlling hourglassing also depends on the loads applied to the structure. For example, a point load is much more likely to trigger hourglassing than a distributed load. Hourglassing can be particularly troublesome in eigenvalue extraction problems: the low stiffness of the hourglass modes may create many unrealistic modes with low eigenfrequencies.

A refinement of the Flanagan and Belytschko (1981) hourglass control method that replaces the artificial stiffness coefficients with those derived from a three-field variational principle is available in Abaqus/Explicit. The approach is based on the enhanced assumed strain and physical hourglass control methods proposed in Engelmann and Whirley (1990), Belytschko and Bindeman (1992), and Puso (2000). It can provide increased resistance to hourglassing for nonlinear problems and coarse mesh displacement solution accuracy for linear elastic problems at a small additional computational cost.

Experience suggests that the reduced-integration, second-order isoparametric elements are the most cost-effective elements in Abaqus for problems in which the solution can be expected to be smooth. Note that in the case of incompressible material behavior, such as hyperelasticity at finite strain, the mixed formulation elements with reduced integration should be used (see "Hybrid incompressible solid element formulation," Section 3.2.3, and "Hyperelastic material behavior," Section 4.6.1). When large strain gradients or strain discontinuities are expected in the solution—such as in plasticity analysis at large strains, limit load analysis, or analysis of severely loaded rubber components—the first-order elements are usually recommended. Reduced integration can be used with such elements, but because the hourglass controls are not always effective in severely nonlinear problems, caution should be exercised.

Fully integrated first-order elements should not be used in cases where "shear locking" can occur, such as when the elements must exhibit bending behavior. The incompatible mode elements ("Continuum elements with incompatible modes," Section 3.2.5) should be used for such applications.

Fully integrated first-order isoparametric elements

For fully integrated first-order isoparametric elements (4-node elements in two dimensions and 8-node elements in three dimensions) the actual volume changes at the Gauss points are replaced by the

average volume change of the element. This is also known as the selectively reduced-integration technique, because the order of integration is reduced in selected terms, or as the $\overline{\beta}$ technique, since the strain-displacement relation (β-matrix) is modified. This technique helps to prevent mesh locking and, thus, provides accurate solutions in incompressible or nearly incompressible cases: see Nagtegaal et al. (1974). In addition, Abaqus uses the average strain in the third (out-of-plane) direction for axisymmetric and generalized plane strain problems. Hence, in the two-dimensional elements only the in-plane terms need to be modified. In the three-dimensional elements the complete volumetric terms are modified. This may cause slightly different behavior between plane strain elements and three-dimensional elements for which a plane strain condition is enforced by boundary conditions.

In a finite-strain formulation the selectively reduced-integration procedure works as follows. Define the modified deformation gradient

$$\overline{\mathbf{F}} = \mathbf{F}\left(\frac{\overline{J}}{J}\right)^{\frac{1}{n}},$$

where $\mathbf{F} = \partial\mathbf{x}/\partial\mathbf{X}$ is the deformation gradient; $\boldsymbol{n}$ is the dimension of the element; $J = \det(\mathbf{F})$ is the Jacobian at the Gauss point; and $\overline{J}$ is the average Jacobian over the element,

$$\overline{J} = \frac{1}{V_{e\ell}}\int_{V_{e\ell}} J\, dV_{e\ell}.$$

For three-dimensional elements $n = 3$ and $\boldsymbol{J}(\overline{J})$ are the volume change; for two-dimensional elements $n = 2$ and $\boldsymbol{J}(\overline{J})$ are the change in area. Note that in the last case $\overline{J}$ is the change in area averaged over the element *volume*, which is different from the actual element area for distorted elements with variable thickness.

The modified rate of deformation tensor, $\overline{\mathbf{D}}$, is obtained from the modified deformation gradient $\overline{\mathbf{F}}$ as

$$\overline{\mathbf{D}} = \text{sym}\left(\dot{\overline{\mathbf{F}}}\cdot\overline{\mathbf{F}}^{-1}\right) = \text{sym}\left(\dot{\mathbf{F}}\cdot\mathbf{F}^{-1}\right) + f\,\mathbf{I}\left(\frac{\dot{\overline{J}}}{\overline{J}} - \frac{\dot{J}}{J}\right),$$

where $f = 1/2$ for two-dimensional elements, $f = 1/3$ for three-dimensional elements, and $\mathbf{I}$ is the identity matrix in two or three dimensions. This expression can also be written directly in terms of the velocities:

$$\overline{\mathbf{D}} = \text{sym}\left(\frac{\partial\dot{\mathbf{u}}}{\partial\mathbf{x}}\right) + f\,\mathbf{I}\left(\text{trace}\left(\overline{\frac{\partial\dot{\mathbf{u}}}{\partial\mathbf{x}}}\right) - \text{trace}\left(\frac{\partial\dot{\mathbf{u}}}{\partial\mathbf{x}}\right)\right),$$

where

$$\overline{\frac{\partial\dot{\mathbf{u}}}{\partial\mathbf{x}}} = \frac{1}{v_{e\ell}}\int_{v_{e\ell}}\frac{\partial\dot{\mathbf{u}}}{\partial\mathbf{x}}\, dv_{e\ell}.$$

This expression is used in the virtual work equation, where it is used to obtain the nodal forces from the element stresses. In Abaqus the central difference operator is used to define an increment of strain from the rate of deformation tensor, so we can write

$$\Delta\overline{\mathbf{D}} = \text{sym}\left(\frac{\partial\Delta\mathbf{u}}{\partial\mathbf{x}}\right) + f\,\mathbf{I}\left(\text{trace}\left(\overline{\frac{\partial\Delta\mathbf{u}}{\partial\mathbf{x}}}\right) - \text{trace}\left(\frac{\partial\Delta\mathbf{u}}{\partial\mathbf{x}}\right)\right).$$

In the above $\mathbf{x} = (1/2)(\mathbf{x}_t + \mathbf{x}_{t+\Delta t})$ is the position of the point at the middle of the increment.

For axisymmetric and generalized plane strain elements, the out-of-plane component of the deformation gradient is obtained by averaging over the original element volume,

$$\overline{F}_{33} = \frac{1}{V_{e\ell}}\int_{V_{e\ell}} F_{33}\,dV_{e\ell},$$

and the out-of-plane strain increment is calculated by averaging over the current element volume,

$$\Delta\overline{D}_{33} = \frac{1}{v_{e\ell}}\int_{v_{e\ell}} \Delta D_{33}\,dv_{e\ell}.$$

Both of these averages are calculated analytically.

Integration of composite solids

The composite solid elements are also integrated numerically to obtain the element matrices. Gauss quadrature is used in the layer plane, and Simpson's rule is used in the stacking direction. These integration positions are referred to as "integration points" and "section points," respectively, for output purposes. The number of section points required for the integration through the thickness of each layer is specified by the user.

Reference

- "Solid (continuum) elements," Section 28.1.1 of the Abaqus Analysis User's Guide

3.2.5 CONTINUUM ELEMENTS WITH INCOMPATIBLE MODES

Products: Abaqus/Standard Abaqus/Explicit

The lower-order quadrilateral continuum elements in Abaqus of type CPS4I, CPE4I, CAX4I, CPEG4I, and C3D8I, as well as the related hybrid elements, are enhanced by incompatible modes to improve the bending behavior. In addition to the displacement degrees of freedom, incompatible deformation modes are added internal to the elements. The primary effect of these degrees of freedom is to eliminate the so-called parasitic shear stresses that are observed in regular displacement elements if they are loaded in bending.

In addition, these degrees of freedom eliminate artificial stiffening due to Poisson's effect in bending. In regular displacement elements the linear variation of the axial stress due to bending is accompanied by a linear variation of the stress perpendicular to the bending direction, which leads to incorrect stresses and an overestimation of the stiffness. The incompatible modes prevent such a stress from occurring.

In the nonhybrid elements (except CPS4I) additional incompatible modes are added to prevent locking of the elements for approximately incompressible material behavior. For fully incompressible material behavior, hybrid elements must be used. In these elements pressure degrees of freedom are added to enforce a linear pressure variation inside the element. In the hybrid elements the additional incompatible modes used to prevent locking are not included.

The incompatible mode elements perform almost as well as second-order elements in many situations if the elements have an approximately rectangular shape. The performance is considerably less if the elements have a parallelogram shape. For trapezoidal element shapes the performance is not much better than the performance of regular displacement elements.

Because of the internal degrees of freedom (4 for CPS4I; 5 for CPE4I, CAX4I, and CPEG4I; and 13 for C3D8I) the elements are somewhat more expensive than regular displacement elements. However, the additional degrees of freedom do not substantially increase the wavefront size since they can be eliminated immediately. In addition, it is not necessary to use selectively reduced integration, which partially offsets the cost of the additional degrees of freedom.

The geometrically linear incompatible mode formulation used in Abaqus is related to the work presented by Simo and Rifai (1990). Simo's formulation is very similar to much earlier work done by Wilson et al. (1973) and Taylor et al. (1976). The nonlinear formulation is based on work by Simo and Armero (1992).

Geometrically linear formulation

As discussed in the paper by Simo, the incompatible mode formulation can be derived in a rigorous way from the general Hu-Washizu variational principle. In this discussion we will not present this derivation but use only the key results of Simo's work.

In the incompatible mode formulation, the displacement gradient $\mathbf{G} = \dfrac{\partial \mathbf{u}}{\partial \mathbf{x}}$ is augmented with an additional, incompatible displacement gradient field $\tilde{\mathbf{G}}$:

$$\overline{\mathbf{G}} = \mathbf{G} + \tilde{\mathbf{G}} = \frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \tilde{\mathbf{G}}.$$

The incompatible displacement gradient $\tilde{\mathbf{G}}$ is chosen internal to an element. The field cannot be selected arbitrarily. It must be independent of the regular displacement gradient.

$$\tilde{\mathbf{G}} \cap \mathbf{G} = 0,$$

which can also be expressed in the form

$$\tilde{\mathbf{G}} - \mathbf{G} \neq 0 \quad \text{for all} \quad \overline{\mathbf{G}} \neq 0.$$

In addition, it must be orthogonal to any constant gradient field, which yields the condition

$$\int_{V_{el}} \tilde{\mathbf{G}}\, dV_{el} = 0. \tag{3.2.5–1}$$

If these conditions are violated, the element does not pass the patch test.

The last condition is used to obtain a suitable general form of the incompatible modes. We describe the incompatible field as a transformation of a parametric gradient field $\tilde{\mathbf{g}}(\boldsymbol{\xi})$:

$$\tilde{\mathbf{G}}(\boldsymbol{\xi}) = \frac{j(0)}{j(\boldsymbol{\xi})}\, \tilde{\mathbf{g}}(\boldsymbol{\xi}) \cdot \mathbf{t}^{-1}, \tag{3.2.5–2}$$

where $\mathbf{t}$ is the parametric transformation at the center of the element

$$\mathbf{t} = \left.\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}\right|_{\boldsymbol{\xi}=0},$$

$j(\boldsymbol{\xi})$ is the Jacobian of the parametric transformation at the location $\boldsymbol{\xi}$, and $j(0)$ is the Jacobian at the center of the element. For planar elements the Jacobian can be written as

$$j(\boldsymbol{\xi}) = h(\boldsymbol{\xi}) \det\left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}\right),$$

where $\boldsymbol{h}$ is the thickness; for axisymmetric elements it is

$$j(\boldsymbol{\xi}) = 2\pi r(\boldsymbol{\xi}) \det\left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}\right),$$

where $\boldsymbol{r}$ is the radius; and for three-dimensional elements it is

$$j(\boldsymbol{\xi}) = \det\left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}\right).$$

Substitution of Equation 3.2.5–2 in Equation 3.2.5–1 allows us to create a simple condition for $\tilde{\mathbf{g}}(\boldsymbol{\xi})$:

$$\int_{V_{el}} \tilde{\mathbf{G}}\, dV_{el} = \int_{V_{el}} \frac{j(0)}{j(\boldsymbol{\xi})}\, \tilde{\mathbf{g}}(\boldsymbol{\xi}) \cdot \mathbf{t}^{-1} dV_{el} = j(0) \left(\int_{V_{par}} \tilde{\mathbf{g}}(\boldsymbol{\xi})\, dV_{par}\right) \cdot \mathbf{t}^{-1} = 0.$$

For two-dimensional elements this yields

$$\int_{-1}^{+1}\int_{-1}^{+1} \tilde{\mathbf{g}}(\boldsymbol{\xi})\, d\xi_1 d\xi_2 = 0;$$

and for three-dimensional elements,

$$\int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \tilde{\mathbf{g}}(\boldsymbol{\xi})\, d\xi_1 d\xi_2 d\xi_3 = 0.$$

This makes it possible to write $\tilde{\mathbf{g}}$ as a simple polynomial in $\boldsymbol{\xi}$. The principal contribution to $\tilde{\mathbf{g}}$ can be written in the form

$$\tilde{\mathbf{g}}_p(\boldsymbol{\xi}) = \boldsymbol{\alpha}^i\, \boldsymbol{\xi}_i,$$

where $\boldsymbol{\alpha}^i$ are vectorial degrees of freedom and $\boldsymbol{\xi}_i$ are vectors and the summation $\boldsymbol{i}$ extends over the parametric coordinates. In two-dimensional elements $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ are vectors of the form

$$\boldsymbol{\xi}_1 = \left\{ \begin{array}{c} \xi_1 \\ 0 \end{array} \right\} \quad \text{and} \quad \boldsymbol{\xi}_2 = \left\{ \begin{array}{c} 0 \\ \xi_2 \end{array} \right\},$$

and in three-dimensional elements

$$\boldsymbol{\xi}_1 = \left\{ \begin{array}{c} \xi_1 \\ 0 \\ 0 \end{array} \right\} \quad , \quad \boldsymbol{\xi}_2 = \left\{ \begin{array}{c} 0 \\ \xi_2 \\ 0 \end{array} \right\} \quad , \text{and} \quad \boldsymbol{\xi}_3 = \left\{ \begin{array}{c} 0 \\ 0 \\ \xi_3 \end{array} \right\}.$$

The principal contribution to the incompatible displacement gradient hence becomes

$$\tilde{\mathbf{G}}_p(\boldsymbol{\xi}) = \boldsymbol{\alpha}^i\, \frac{j(0)}{j(\boldsymbol{\xi})}\, \boldsymbol{\xi}_i \cdot \mathbf{t}^{-1}.$$

With the addition of these terms, parasitic shear and Poisson's effect in bending are eliminated. Note that the vectors $\boldsymbol{\alpha}^i$ in $\tilde{\mathbf{G}}$ appear in a similar form as the nodal displacement vectors $\mathbf{u}^i$ in $\mathbf{G}$,

$$\mathbf{G} = \mathbf{u}^i\, \frac{\partial N_i}{\partial \mathbf{x}},$$

and can be treated similar to displacement degrees of freedom.

For approximately incompressible material behavior, bilinear patterns in the hydrostatic stress can still be observed in all elements except CPS4I. These patterns can be eliminated by the introduction of additional incompatible modes of the form

$$\tilde{\mathbf{G}}_I(\boldsymbol{\xi}) = \beta^j\, \frac{j(0)}{j(\boldsymbol{\xi})}\, \vartheta_j(\boldsymbol{\xi})\, \mathbf{I},$$

where β^j are additional scalar degrees of freedom. For two-dimensional elements a single term β^1 is added with

$$\vartheta_1 = \xi_1 \xi_2.$$

In the three-dimensional elements four additional terms β^j are added with

$$\vartheta_1 = \xi_1 \xi_2, \quad \vartheta_2 = \xi_1 \xi_3, \quad \vartheta_3 = \xi_2 \xi_3, \quad \text{and} \quad \vartheta_4 = \xi_1 \xi_2 \xi_3.$$

Thus, the incompatible displacement gradient $\tilde{\mathbf{G}}$ takes the final form

$$\tilde{\mathbf{G}}(\boldsymbol{\xi}) = \boldsymbol{\alpha}^i \frac{j(0)}{j(\boldsymbol{\xi})} \boldsymbol{\xi}_i \cdot \mathbf{t}^{-1} + \beta^j \frac{j(0)}{j(\boldsymbol{\xi})} \vartheta_j(\boldsymbol{\xi}) \mathbf{I}.$$

The symmetric part of the incompatible displacement gradient contributes to incompatible strains:

$$\begin{Bmatrix} \tilde{\varepsilon}_{11} \\ \tilde{\varepsilon}_{22} \\ \tilde{\varepsilon}_{33} \\ \tilde{\gamma}_{12} \\ \tilde{\gamma}_{13} \\ \tilde{\gamma}_{23} \end{Bmatrix} = \begin{Bmatrix} \tilde{G}_{11} \\ \tilde{G}_{22} \\ \tilde{G}_{33} \\ \tilde{G}_{21} + \tilde{G}_{12} \\ \tilde{G}_{31} + \tilde{G}_{13} \\ \tilde{G}_{32} + \tilde{G}_{23} \end{Bmatrix}.$$

The skew-symmetric part plays no role in the geometrically linear formulation.

Geometrically nonlinear formulation

Since we want to use a formulation that can be used for any material model, we want to express the incompatible modes as a modification of the deformation gradient $\mathbf{F}$. The most obvious approach is to add the incompatible modes to the deformation gradient:

$$\overline{\mathbf{F}} = \mathbf{F} + \tilde{\mathbf{F}} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} + \tilde{\mathbf{F}}. \tag{3.2.5–3}$$

This approach has been used successfully by Simo and Armero. Elements formulated on this basis satisfy the large-strain patch test; i.e., any patch of elements will be able to represent homogeneous deformations exactly. However, once the elements become distorted due to deformation, the patch test will no longer be satisfied in an incremental sense; that is, subsequent homogeneous deformations will not be represented exactly. This turns out to be a fatal flaw in the formulation for problems involving large distortions in compression.

Satisfaction of the instantaneous patch test requires the addition of an incompatible deformation rate tensor to the standard rate of deformation:

$$\dot{\overline{\boldsymbol{\varepsilon}}} = \dot{\boldsymbol{\varepsilon}} + \dot{\tilde{\boldsymbol{\varepsilon}}}.$$

To obtain an approximate relation of this type, we write the total deformation gradient as the product of a series of incremental deformation gradients:

$$\overline{\mathbf{F}}^{(n+1)} = \Delta\overline{\mathbf{F}}^{(n+1)} \cdot \Delta\overline{\mathbf{F}}^{(n)} \cdot \ldots \cdot \Delta\overline{\mathbf{F}}^{(1)}.$$

Principal incompatible modes are then added to the incremental deformation gradient:

$$\Delta\overline{\mathbf{F}}_p = \Delta\mathbf{F} + \Delta\tilde{\mathbf{F}} = \frac{\partial \mathbf{x}_{t+\Delta t}}{\partial \mathbf{x}_t} + \Delta\tilde{\mathbf{F}}.$$

Similar to Equation 3.2.5–2 the principal incompatible modes are described as a transformation of the parametric gradient field $\Delta\tilde{\mathbf{g}}(\boldsymbol{\xi})$:

$$\Delta\tilde{\mathbf{F}}(\boldsymbol{\xi}) = \left.\frac{j(0)}{j(\boldsymbol{\xi})}\right|_t \Delta\tilde{\mathbf{g}}(\boldsymbol{\xi}) \cdot \mathbf{t}_t^{-1}, \tag{3.2.5–4}$$

where $\mathbf{t}_t$ is the parametric transformation at the center of the element in the state at the start of the increment,

$$\mathbf{t}_t = \left.\frac{\partial \mathbf{x}_t}{\partial \boldsymbol{\xi}}\right|_{\boldsymbol{\xi}=0};$$

$j(\boldsymbol{\xi})$ is the Jacobian of the parametric transformation at the location $\boldsymbol{\xi}$; and $j(0)$ is the Jacobian at the centroid at the start of the increment. Note that $j(\boldsymbol{\xi})$ is evaluated based on the deformation caused by the displacement degrees of freedom only and does not include the volume change due to the incompatible modes.

The incremental parametric gradient field $\Delta\tilde{\mathbf{g}}(\boldsymbol{\xi})$ has exactly the same form as in the linear formulation,

$$\Delta\tilde{\mathbf{g}}(\boldsymbol{\xi}) = \Delta\boldsymbol{\alpha}^i\, \boldsymbol{\xi}_i,$$

which yields the principal incremental incompatible deformation gradient

$$\Delta\tilde{\mathbf{F}}(\boldsymbol{\xi}) = \Delta\boldsymbol{\alpha}^i \left.\frac{j(0)}{j(\boldsymbol{\xi})}\right|_t \boldsymbol{\xi}_i \cdot \mathbf{t}_t^{-1}.$$

Bilinear volumetric terms are added to the principal terms in a multiplicative way:

$$\Delta\overline{\mathbf{F}} = \Delta\overline{\mathbf{F}}_p\, \Delta F_I = (\Delta\mathbf{F} + \Delta\tilde{\mathbf{F}})\, \exp\left(\Delta\beta^j \left.\frac{j(0)}{j(\boldsymbol{\xi})}\right|_t \vartheta_j(\boldsymbol{\xi})\right). \tag{3.2.5–5}$$

The variation in the gradient of the position with respect to the current state is obtained from the fundamental relation

$$\delta\overline{\mathbf{L}} = \delta\overline{\mathbf{F}} \cdot \overline{\mathbf{F}}^{-1} = \delta\Delta\overline{\mathbf{F}} \cdot \Delta\overline{\mathbf{F}}^{-1} = \Big[(\delta\Delta\mathbf{F} + \delta\Delta\tilde{\mathbf{F}})\Delta F_I + \Delta\overline{\mathbf{F}}_p \delta\Delta F_I\Big] \cdot \Delta\overline{\mathbf{F}}^{-1}$$
$$= (\delta\Delta\mathbf{F} + \delta\Delta\tilde{\mathbf{F}}) \cdot \Delta\overline{\mathbf{F}}_p^{-1} + \delta\Delta F_I \, \Delta F_I^{-1},$$

with

$$\begin{aligned}
\delta\Delta\mathbf{F} &= \frac{\partial\delta\mathbf{x}}{\partial\mathbf{x}_t}, \\
\delta\Delta\tilde{\mathbf{F}} &= \frac{j(0)}{j(\boldsymbol{\xi})}\bigg|_t \delta\tilde{\mathbf{g}}(\boldsymbol{\xi}) \cdot \mathbf{t}_t^{-1} = \delta\boldsymbol{\alpha}^i \frac{j(0)}{j(\boldsymbol{\xi})}\bigg|_t \boldsymbol{\xi}_i \cdot \mathbf{t}_t^{-1}, \\
\delta\Delta F_I &= \delta\beta^j \frac{j(0)}{j(\boldsymbol{\xi})}\bigg|_t \vartheta_j(\boldsymbol{\xi}) \, \Delta F_I.
\end{aligned}$$

This allows us to write for $\delta\overline{\mathbf{L}}$

$$\delta\overline{\mathbf{L}} = \frac{\partial\delta\mathbf{x}}{\partial\mathbf{x}_t} \cdot \Delta\overline{\mathbf{F}}_p^{-1} + \delta\boldsymbol{\alpha}^i \frac{j(0)}{j(\boldsymbol{\xi})}\bigg|_t \boldsymbol{\xi}_i \cdot \mathbf{t}_t^{-1} \cdot \Delta\overline{\mathbf{F}}_p^{-1} + \delta\beta^j \frac{j(0)}{j(\boldsymbol{\xi})}\bigg|_t \vartheta_j(\boldsymbol{\xi}).$$

The integral of the principal incompatible modes over the element volume at the start of the increment is, hence, equal to

$$\int_{v_{el}} \delta\Delta\tilde{\mathbf{F}} \cdot \Delta\overline{\mathbf{F}}_p^{-1} \, dv_{el} = \int_{v_{el}} \frac{j(0)}{j(\boldsymbol{\xi})}\big|_t \delta\tilde{\mathbf{g}}(\boldsymbol{\xi}) \cdot \mathbf{t}_t^{-1} \cdot \Delta\overline{\mathbf{F}}_p^{-1} \, dv_{el}$$
$$= \int_{V_{par}} j(0)\big|_t \delta\tilde{\mathbf{g}}(\boldsymbol{\xi}) \cdot \mathbf{t}_t^{-1} \cdot \Delta\overline{\mathbf{F}}_p^{-1} \, dV_{par}.$$

Note that the integral will vanish if the incremental deformation is *homogeneous*; that is, $\Delta\overline{\mathbf{F}}_p(\boldsymbol{\xi}) = \Delta\overline{\mathbf{F}}_0$, since in that case the integral can be written as

$$\int_{v_{el}} \delta\Delta\tilde{\mathbf{F}} \cdot \Delta\overline{\mathbf{F}}_p^{-1} \, dv_{el} = j(0)\big|_t \left(\int_{V_{par}} \delta\tilde{\mathbf{g}}_p(\boldsymbol{\xi}) \, dV_{par}\right) \cdot \mathbf{t}_t^{-1} \cdot \Delta\overline{\mathbf{F}}_0^{-1} = 0.$$

Hence, the *incremental* patch test will be satisfied. The rate of the gradient of the position with respect to the current state is obtained similar to the variation

$$\mathrm{d}\overline{\mathbf{L}} = \mathrm{d}\overline{\mathbf{F}} \cdot \overline{\mathbf{F}}^{-1} = \mathrm{d}\Delta\overline{\mathbf{F}} \cdot \Delta\overline{\mathbf{F}}^{-1} = \Big[(\mathrm{d}\Delta\mathbf{F} + \mathrm{d}\Delta\tilde{\mathbf{F}})\Delta F_I + \Delta\overline{\mathbf{F}}_p \mathrm{d}\Delta F_I\Big] \cdot \Delta\overline{\mathbf{F}}^{-1}$$
$$= (\mathrm{d}\Delta\mathbf{F} + \mathrm{d}\Delta\tilde{\mathbf{F}}) \cdot \Delta\overline{\mathbf{F}}_p^{-1} + \mathrm{d}\Delta F_I \, \Delta F_I^{-1},$$

with

$$
\begin{aligned}
\mathrm{d}\Delta\mathbf{F} &= \frac{\partial \mathrm{d}\mathbf{x}}{\partial \mathbf{x}_t},\\
\mathrm{d}\Delta\tilde{\mathbf{F}} &= \frac{j(0)}{j(\boldsymbol{\xi})}\bigg|_t \mathrm{d}\tilde{\mathbf{g}}(\boldsymbol{\xi})\cdot\mathbf{t}_t^{-1} = \mathrm{d}\boldsymbol{\alpha}^i \frac{j(0)}{j(\boldsymbol{\xi})}\bigg|_t \boldsymbol{\xi}_i\cdot\mathbf{t}_t^{-1},\\
\mathrm{d}\Delta F_I &= \mathrm{d}\beta^j \frac{j(0)}{j(\boldsymbol{\xi})}\bigg|_t \vartheta_j(\boldsymbol{\xi})\,\Delta F_I.
\end{aligned}
$$

For a finite-strain increment we use the midincrement approach proposed by Hughes and Winget. This yields

$$
\Delta\overline{\mathbf{L}} \approx \Delta\overline{\mathbf{F}}\cdot\Delta\overline{\mathbf{F}}^{-1}_{t+\Delta t/2}.
$$

The second variation is obtained in the usual way. Since the regular deformation gradient and the principal incompatible modes are purely displacement based, the initial stress stiffness terms are readily obtained as

$$
\int_{v_{el}} \boldsymbol{\sigma} : (\delta\overline{\mathbf{L}}_p^T\cdot\mathrm{d}\overline{\mathbf{L}}_p - 2\delta\overline{\mathbf{D}}_p\cdot\mathrm{d}\overline{\mathbf{D}}_p)\,\mathrm{d}v_{el},
$$

where $\mathrm{d}\overline{\mathbf{L}}_p = \mathrm{d}\overline{\mathbf{F}}_p\cdot\Delta\overline{\mathbf{F}}_p^{-1}$ is the gradient of the velocity and $\delta\overline{\mathbf{L}}_p = \delta\overline{\mathbf{F}}_p\cdot\Delta\overline{\mathbf{F}}_p^{-1}$ is the gradient of the displacement variation including the primary incompatible modes. Further, $\mathrm{d}\overline{\mathbf{D}}_p = \frac{1}{2}(\mathrm{d}\overline{\mathbf{L}}_p + \mathrm{d}\overline{\mathbf{L}}_p^T)$ is the rate of deformation, and $\delta\overline{\mathbf{D}}_p = \frac{1}{2}(\delta\overline{\mathbf{L}}_p + \delta\overline{\mathbf{L}}_p^T)$ is the variation in the deformation.

The additional bilinear modes appear in the first variation only as variations (the values β^i themselves do not appear); hence, the contributions to the second variation can be neglected.

Reference

- "Solid (continuum) elements," Section 28.1.1 of the Abaqus Analysis User's Guide

3.2.6 TRIANGULAR, TETRAHEDRAL, AND WEDGE ELEMENTS

Products: Abaqus/Standard Abaqus/Explicit

The library of solid elements in Abaqus includes first- and second-order triangles, tetrahedra, and wedge elements for planar, axisymmetric, and three-dimensional analysis.

Hybrid versions of these elements are provided for use with incompressible and nearly incompressible constitutive models (see "Hybrid incompressible solid element formulation," Section 3.2.3, for a detailed discussion of the formulation used). However, these hybrid forms should be used only to fill in regions in meshes made of brick elements; otherwise, too many constraint variables may be introduced.

Second-order tetrahedra are not suitable for the analysis of contact problems: a constant pressure on an element face produces zero equivalent loads at the corner nodes. In contact problems this makes the contact condition at the corners indeterminate, with failure of the solution likely because of excessive gap chatter. The same argument holds true for contact on triangular faces of a wedge element.

Interpolation

The interpolation is defined in terms of the element coordinates g, h, and r shown in Figure 3.2.6–1. Since Abaqus is a Lagrangian code for most applications, these are also material coordinates. They each span a range from 0 to 1 in an element but satisfy the constraint that $g + h \leq 1$ for triangles and wedges and $g + h + r \leq 1$ for tetrahedra. The node numbering convention used in Abaqus for these elements is also shown in Figure 3.2.6–1. Corner nodes are numbered first, and then the midside nodes for second-order elements. The interpolation functions are as follows.

First-order triangle (3 nodes):

$$u = (1 - g - h)\, u_1 + g\, u_2 + h\, u_3$$

Second-order triangle (6 nodes):

$$\begin{aligned} u =& 2(\frac{1}{2} - g - h)(1 - g - h)\, u_1 + 2g(g - \frac{1}{2})\, u_2 + 2h(h - \frac{1}{2})\, u_3 \\ &+ 4g(1 - g - h)\, u_4 + 4gh\, u_5 + 4h(1 - g - h)\, u_6 \end{aligned}$$

First-order tetrahedron (4 nodes):

$$u = (1 - g - h - r)\, u_1 + g\, u_2 + h\, u_3 + r\, u_4$$

Second-order tetrahedron (10 nodes):

$$\begin{aligned} u =& (2(1 - g - h - r) - 1)(1 - g - h - r)\, u_1 + (2g - 1)g\, u_2 + (2h - 1)h\, u_3 \\ &+ (2r - 1)r\, u_4 + 4(1 - g - h - r)g\, u_5 + 4gh\, u_6 + 4(1 - g - h - r)h\, u_7 \\ &+ 4(1 - g - h - r)r\, u_8 + 4gr\, u_9 + 4hr\, u_{10} \end{aligned}$$

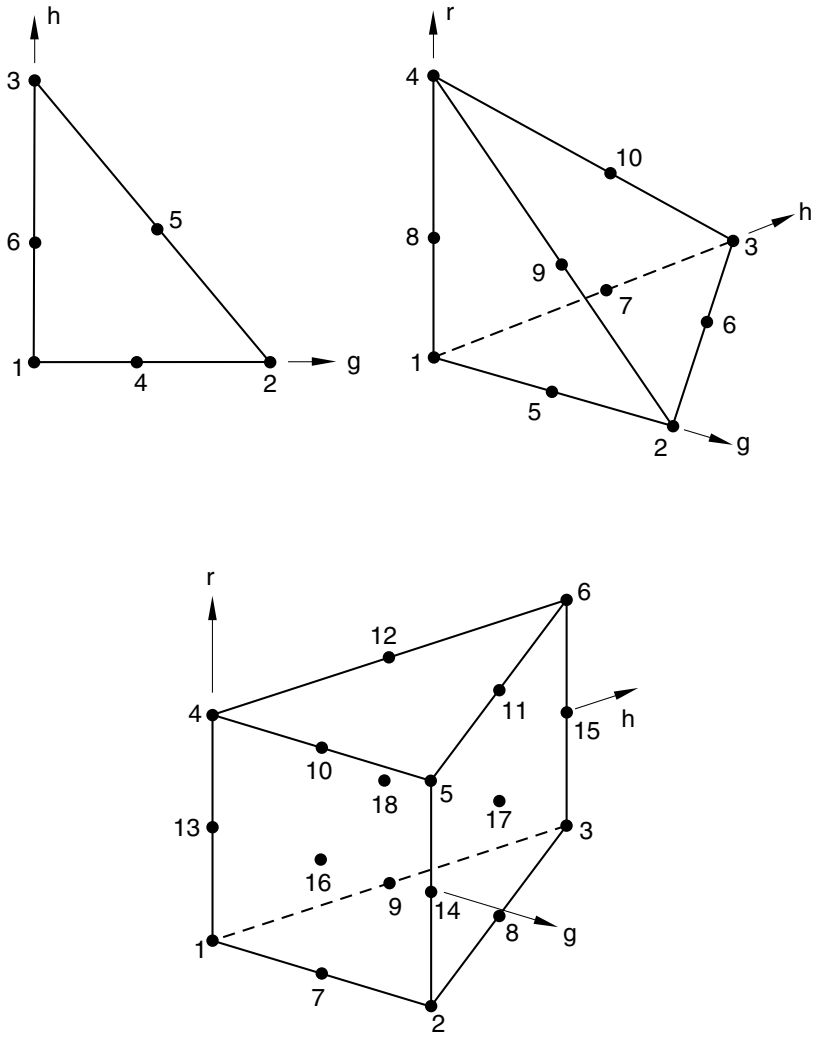


Figure 3.2.6–1 Isoparametric master elements.

First-order wedge (6 nodes):

$$u = \frac{1}{2}(1-g-h)(1-r)u_1 + \frac{1}{2}g(1-r)u_2 + \frac{1}{2}h(1-r)u_3 + \frac{1}{2}(1-g-h)(1+r)u_4 + \frac{1}{2}g(1+r)u_5 + \frac{1}{2}h(1+r)u_6$$

Second-order wedge (15 nodes):

$$
\begin{aligned}
u = &\frac{1}{2}((1-g-h)(2(1-g-h)-1)(1-r)-(1-g-h)(1-r^2))u_1 \\
&+\frac{1}{2}(g(2g-1)(1-r)-g(1-r^2))u_2+\frac{1}{2}(h(2h-1)(1-r)-h(1-r^2))u_3 \\
&+\frac{1}{2}((1-g-h)(2(1-g-h)-1)(1+r)-(1-g-h)(1-r^2))u_4 \\
&+\frac{1}{2}(g(2g-1)(1+r)-g(1-r^2))u_5+\frac{1}{2}(h(2h-1)(1+r)-h(1-r^2))u_6 \\
&+2(1-g-h)g(1-r)u_7+2gh(1-r)u_8+2h(1-g-h)(1-r)u_9 \\
&+2(1-g-h)g(1+r)u_{10}+2gh(1+r)u_{11}+2h(1-g-h)(1+r)u_{12} \\
&+(1-g-h)(1-r^2)u_{13}+g(1-r^2)u_{14}+h(1-r^2)u_{15}
\end{aligned}
$$

Second-order variable 15–18 node wedge (assuming all 18 nodes are defined):

$$
\begin{aligned}
u = &(\frac{1}{2}((1-g-h)(2(1-g-h)-1)(1-r)-(1-g-h)(1-r^2))+\frac{1}{4}(N_{16}+N_{18}))u_1 \\
&+(\frac{1}{2}(g(2g-1)(1-r)-g(1-r^2))+\frac{1}{4}(N_{16}+N_{17}))u_2 \\
&+(\frac{1}{2}(h(2h-1)(1-r)-h(1-r^2))+\frac{1}{4}(N_{17}+N_{18}))u_3 \\
&+(\frac{1}{2}((1-g-h)(2(1-g-h)-1)(1+r)-(1-g-h)(1-r^2))+\frac{1}{4}(N_{16}+N_{18}))u_4 \\
&+(\frac{1}{2}(g(2g-1)(1+r)-g(1-r^2))+\frac{1}{4}(N_{16}+N_{17}))u_5 \\
&+(\frac{1}{2}(h(2h-1)(1+r)-h(1-r^2))+\frac{1}{4}(N_{17}+N_{18}))u_6 \\
&+(2(1-g-h)g(1-r)-\frac{1}{2}N_{16})u_7+(2gh(1-r)-\frac{1}{2}N_{17})u_8 \\
&+(2h(1-g-h)(1-r)-\frac{1}{2}N_{18})u_9+(2(1-g-h)g(1+r)+\frac{1}{2}N_{16})u_{10} \\
&+(2gh(1+r)+\frac{1}{2}N_{17})u_{11}+(2h(1-g-h)(1+r)+\frac{1}{2}N_{18})u_{12} \\
&+((1-g-h)(1-r^2)-\frac{1}{2}(N_{16}+N_{18}))u_{13}+(g(1-r^2)-\frac{1}{2}(N_{16}+N_{17}))u_{14} \\
&+(h(1-r^2)-\frac{1}{2}(N_{17}+N_{18}))u_{15}++N_{16}u_{16}+N_{17}u_{17}+N_{18}u_{18},
\end{aligned}
$$

where

$$N_{16} = 4g(1-g-h)(1-r^2)$$

$$N_{17} = 4gh(1 - r^2)$$

$$N_{18} = 4h(1 - g - h)(1 - r^2).$$

Integration

The first-order triangle and tetrahedron are constant stress elements and use a single integration point for the stiffness calculation when used in stress/displacement applications. A lumped mass matrix is used for both elements, with the total mass divided equally over the nodes. For heat transfer applications a three-point integration scheme is used for the conductivity and heat capacity matrices of the first-order triangle, with the integration points midway between the vertices and the centroid of the element; and a four-point integration scheme is used for the first-order tetrahedron. Distributed loads are integrated with two and three points for first-order triangles and tetrahedrons, respectively.

The three-point scheme is also used for the stiffness of the second-order triangle when it is used in stress/displacement applications. The mass matrix is integrated with a six-point scheme that integrates fourth-order polynomials exactly (Cowper, 1973). Distributed loads are integrated using three points. The heat transfer versions of the element use the six-point scheme for the conductivity and heat capacity matrices.

For stress/displacement applications the second-order tetrahedron uses 4 integration points for its stiffness matrix and 15 integration points for its consistent mass matrix. For heat transfer applications the conductivity and heat capacity matrices are integrated using 15 integration points. The first-order wedge uses 2 integration points for its stiffness matrix but 6 integration points for its lumped mass matrix. The second-order wedge uses 9 integration points for its stiffness matrix but 18 integration points for its consistent mass matrix. The integration schemes used for the second-order tetrahedra and wedge elements can be found in Stroud (1971).

Reference

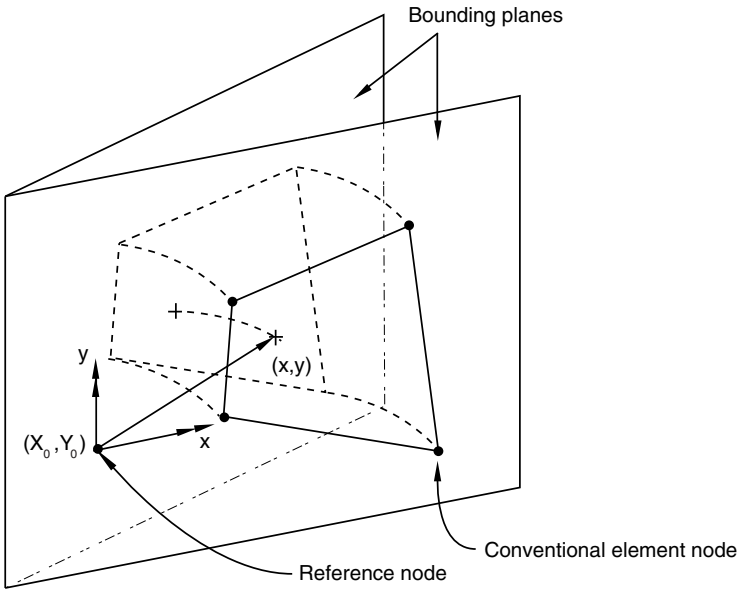
- “Solid (continuum) elements,” Section 28.1.1 of the Abaqus Analysis User’s Guide

3.2.7 GENERALIZED PLANE STRAIN ELEMENTS

Product: Abaqus/Standard

The generalized plane strain theory used in Abaqus assumes that the model lies between two bounding planes, which may move as rigid bodies with respect to each other, thus causing strain of the "thickness direction" fibers of the model. It is assumed that the deformation of the model is independent of position with respect to this thickness direction, so the relative motion of the two planes causes a direct strain of the thickness direction fibers only. This strain and its first and second variations are defined as follows.

Let $P_0(X_0, Y_0)$ be a fixed point in one of the bounding planes, as shown in Figure 3.2.7–1. The length of the fiber between P_0 and its image in the other bounding plane is $t_0 + \Delta u_z$, where t_0 is the length of this fiber in the initial configuration and Δu_z is the change in length of this fiber. Δu_z is the value of degree of freedom 3 at the reference node of the element.



Length of line through the thickness at (x,y) is

$$t_0 + \Delta u_z + \Delta\phi_x\,(y - Y_0) - \Delta\phi_y\,(x - X_0)$$

where quantities are defined in the text.

Figure 3.2.7–1 Generalized plane strain element.

The reference node should be the same for all elements in any given connected region so that the bounding planes are the same for that region. Different regions may have different reference nodes. Since the bounding planes are rigid, the length of a fiber at any other point (x, y) in the element is

$$t = t_0 + \Delta u_z + \Delta\phi_x(y - Y_0) - \Delta\phi_y(x - X_0),$$

where

$$\Delta\phi_x = (\Delta\phi_x)|_0 + (\Delta\phi_x)_1,$$
$$\Delta\phi_y = (\Delta\phi_y)|_0 + (\Delta\phi_y)_1,$$

where $(\Delta\phi_i)|_0; \quad i = x, y$, are the initial values of $\Delta\phi_i$, specified by the user; and $(\Delta\phi_i)|_1$ are the degrees of freedom 4 and 5 at the reference node of the element.

The thickness direction logarithmic strain is

$$\varepsilon_{zz} = \ln\left(\frac{t}{t_0}\right).$$

The first variation of thickness direction strain is, therefore,

$$\delta\varepsilon_{zz} = \frac{\delta t}{t},$$

where

$$\delta t = -\Delta\phi_y\,\delta x + \Delta\phi_x\,\delta y - x\,\delta\Delta\phi_y + y\,\delta\Delta\phi_x + \delta\Delta u_z;$$

and the second variation is

$$d\delta\varepsilon_{zz} = \frac{d\delta t}{t} - \frac{\delta t\,dt}{t^2},$$

where

$$d\delta t = -\delta x\,d\Delta\phi_y + \delta y\,d\Delta\phi_x - dx\,\delta\Delta\phi_y + dy\,\delta\Delta\phi_x.$$

References

- "Choosing the element's dimensionality," Section 27.1.2 of the Abaqus Analysis User's Guide
- "Two-dimensional solid element library," Section 28.1.3 of the Abaqus Analysis User's Guide

3.2.8 AXISYMMETRIC ELEMENTS

Products: Abaqus/Standard Abaqus/Explicit

Abaqus includes two libraries of solid elements, CAX and CGAX, whose geometry is axisymmetric (bodies of revolution) and which can be subjected to axially symmetric loading conditions. In addition, CGAX elements support torsion loading. As a result, CGAX elements will be referred to as generalized axisymmetric elements, and CAX elements as torsionless axisymmetric elements. In both cases, the body of revolution is generated by revolving a plane cross-section about an axis (the symmetry axis) and is readily described in cylindrical polar coordinates r, z, and θ. The radial and axial coordinates of a point on this cross-section are denoted by r and z, respectively. At $\theta = 0$, the radial and axial coordinates coincide with the global Cartesian X- and Y-coordinates.

If the loading consists of radial and axial components that are independent of θ and the material is either isotropic or orthotropic, with θ being a principal material direction, the displacement at any point will only have radial (u_r) and axial (u_z) components and the only stress components that will be nonzero are σ_{rr}, σ_{zz}, $\sigma_{\theta\theta}$, and σ_{rz}. Moreover, the deformation of any r–z plane completely defines the state of strain and stress in the body. Consequently, the geometric model is described by discretizing the reference cross-section at $\theta = 0$.

If one allows for a circumferential component of loading (which is independent of θ) and for general material anisotropy, displacements and stress fields become three-dimensional, but the problem remains axisymmetric in the sense that the solution does not vary as a function of θ and the deformation of the reference r–z cross-section still characterizes the deformation in the entire body. The motion at any point will have, in addition to the aforementioned radial and axial displacements, a twist ϕ (in radians) about the z-axis, which is independent of θ.

This section describes the formulation of the generalized axisymmetric elements. The formulation of the torsionless axisymmetric elements is a subset of this formulation.

Kinematic description

The coordinate system used with both families of elements is the cylindrical system (r, z, θ), where r measures the distance of a point from the axis of the cylindrical system, z measures its position along this axis, and θ measures the angle between the plane containing the point and the axis of the coordinate system and some fixed reference plane that contains the coordinate system axis. The order in which the coordinates and displacements are taken in these elements is based on the convention that z is the second coordinate. This order is not the same as that used in three-dimensional elements in Abaqus, in which z is the third coordinate, nor is it the order (r, θ, z), usually taken in cylindrical systems.

Let $\mathbf{e}_R$, $\mathbf{e}_Z$, and $\mathbf{e}_\Theta$ be unit vectors in the radial, axial, and circumferential directions at a point in the undeformed state, as shown in Figure 3.2.8–1. The reference position $\mathbf{X}$ of the point can be represented in terms of the original radius R and the axial position Z:

$$\mathbf{X} = R\mathbf{e}_R + Z\mathbf{e}_Z \, .$$

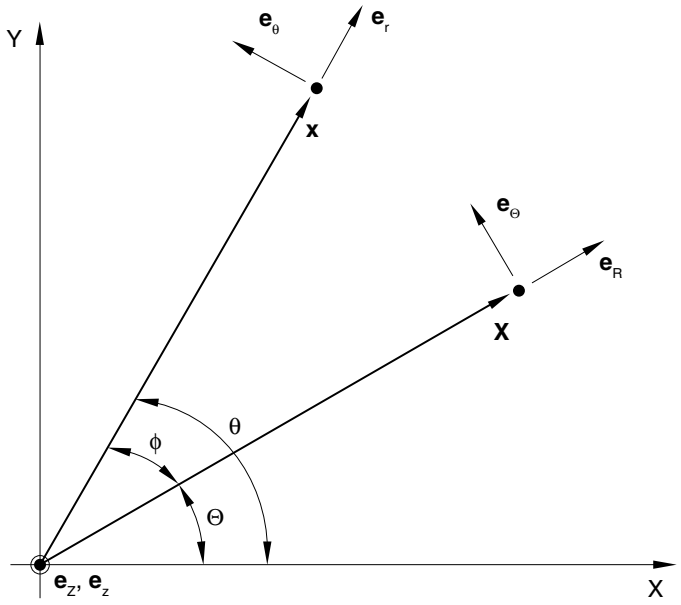


Figure 3.2.8–1 Cylindrical coordinate system and definition of position vectors.

Likewise, let $\mathbf{e}_r$, $\mathbf{e}_z$, and $\mathbf{e}_\theta$ be unit vectors in the radial, axial, and circumferential directions at a point in the deformed state. As shown in Figure 3.2.8–1, the radial and circumferential base vectors depend on the θ coordinate: $\mathbf{e}_r = \mathbf{e}_r(\theta)$ and $\mathbf{e}_\theta = \mathbf{e}_\theta(\theta)$. The current position $\mathbf{x}$ of the point can be represented in terms of the current radius r and the current axial position z:

$$\mathbf{x} = r\mathbf{e}_r + z\mathbf{e}_z \,. \tag{3.2.8–1}$$

The general axisymmetric motion at a point can be described by

$$\begin{aligned} r(R,Z) &= R + u_r(R,Z)\,, \\ z(R,Z) &= Z + u_z(R,Z)\,, \\ \theta(R,Z,\Theta) &= \Theta + \phi(R,Z)\,. \end{aligned} \tag{3.2.8–2}$$

As the above description implies, the degrees of freedom u_r, u_z, and ϕ are independent of Θ . Moreover, the reference cross-section of interest is at $\Theta = 0$, but for the benefit of the mathematical analysis to follow it is important that Θ be nonzero in the above expression for θ.

Parametric interpolation and integration

The following isoparametric interpolation scheme for the motion is used:

$$u_r = N^N(g,h)\bar{u_r}^N\,,$$
$$u_z = N^N(g,h)\bar{u_z}^N\,,$$
$$\phi = N^N(g,h)\bar{\phi}^N\,,$$

where g, h are isoparametric coordinates in the reference r–z cross-section at $\Theta = 0$, and $\bar{u_r}^N$, $\bar{u_z}^N$, $\bar{\phi}^N$ are the nodal degrees of freedom. The interpolation functions $N^N(g,h)$ are those described in "Solid isoparametric quadrilaterals and hexahedra," Section 3.2.4, where the integration scheme of isoparametric solid elements is also discussed.

Deformation gradient

For a material point in space, the deformation gradient $\mathbf{F}$ is defined as the gradient of the current position $\mathbf{x}$ with respect to the original position $\mathbf{X}$:

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}.$$

The current position $\mathbf{x}$ is given by Equation 3.2.8–1, and the gradient operator can be described in terms of partial derivatives with respect to the cylindrical coordinates:

$$\frac{\partial(\cdot)}{\partial \mathbf{X}} = \frac{\partial(\cdot)}{\partial R}\mathbf{e}_R + \frac{\partial(\cdot)}{\partial Z}\mathbf{e}_Z + \frac{1}{R}\frac{\partial(\cdot)}{\partial \Theta}\mathbf{e}_\Theta.$$

Since the radial and circumferential base vectors depend on the original circumferential coordinate θ, the partial derivatives of these base vectors with respect to θ are nonvanishing:

$$\frac{\partial \mathbf{e}_r}{\partial \theta} = \mathbf{e}_\theta, \qquad \frac{\partial \mathbf{e}_\theta}{\partial \theta} = -\mathbf{e}_r.$$

Thus, the chain rule allows us to write

$$\frac{\partial \mathbf{e}_r}{\partial R} = \frac{\partial \mathbf{e}_r}{\partial \theta}\frac{\partial \theta}{\partial R} = \frac{\partial \theta}{\partial R}\mathbf{e}_\theta \quad \text{and} \quad \frac{\partial \mathbf{e}_r}{\partial Z} = \frac{\partial \mathbf{e}_r}{\partial \theta}\frac{\partial \theta}{\partial Z} = \frac{\partial \theta}{\partial Z}\mathbf{e}_\theta\,.$$

With these results, the deformation gradient is obtained as

$$\begin{aligned}\mathbf{F} =& \frac{\partial r}{\partial R}\mathbf{e}_r\mathbf{e}_R + \frac{\partial r}{\partial Z}\mathbf{e}_r\mathbf{e}_Z + \frac{\partial z}{\partial R}\mathbf{e}_z\mathbf{e}_R + \frac{\partial z}{\partial Z}\mathbf{e}_z\mathbf{e}_Z + \\ & r\frac{\partial \theta}{\partial R}\mathbf{e}_\theta\mathbf{e}_R + r\frac{\partial \theta}{\partial Z}\mathbf{e}_\theta\mathbf{e}_Z + \frac{r}{R}\frac{\partial \theta}{\partial \Theta}\mathbf{e}_\theta\mathbf{e}_\Theta\,.\end{aligned} \tag{3.2.8–3}$$

Alternatively, it can be written in matrix form as

$$[F] = \begin{bmatrix} 1+\partial u_r/\partial R & \partial u_r/\partial Z & 0 \\ \partial u_z/\partial R & 1+\partial u_z/\partial Z & 0 \\ r\partial\phi/\partial R & r\partial\phi/\partial Z & r/R \end{bmatrix},$$

where the motion given by Equation 3.2.8–2 has been used explicitly.

Similarly, the inverse deformation gradient $\mathbf{F}^{-1}$ is readily obtained as

$$\mathbf{F}^{-1} = \frac{\partial R}{\partial r}\mathbf{e}_R\mathbf{e}_r + \frac{\partial R}{\partial z}\mathbf{e}_R\mathbf{e}_z + \frac{\partial Z}{\partial r}\mathbf{e}_Z\mathbf{e}_r + \frac{\partial Z}{\partial z}\mathbf{e}_Z\mathbf{e}_z + R\frac{\partial \Theta}{\partial r}\mathbf{e}_\Theta\mathbf{e}_r + R\frac{\partial \Theta}{\partial z}\mathbf{e}_\Theta\mathbf{e}_z + \frac{R}{r}\frac{\partial \Theta}{\partial \theta}\mathbf{e}_\Theta\mathbf{e}_\theta \,.$$

Virtual work

As discussed in "Equilibrium and virtual work," Section 1.5.1, the formulation of equilibrium (virtual work) requires the virtual velocity gradient $\delta\mathbf{L}$, which is the variation in the gradient of the position with respect to the current state. This tensor is given by

$$\delta\mathbf{L} = \delta\mathbf{F} \cdot \mathbf{F}^{-1} \tag{3.2.8–4}$$

where $\delta\mathbf{F}$ is the linearized deformation gradient.

Abaqus formulates the finite element equations in terms of a fixed spatial basis with respect to the axisymmetric twist degree of freedom. Therefore, the desired result for $\delta\mathbf{F}$ in Equation 3.2.8–4 does not simply follow from the linearization of Equation 3.2.8–3. Namely, it is necessary to cancel out the contributions from the variations

$$\delta\mathbf{e}_r = \delta\phi\ \mathbf{e}_\theta \quad \text{and} \quad \delta\mathbf{e}_\theta = -\delta\phi\ \mathbf{e}_r.$$

To this end $\mathbf{F}$ can be modified according to

$$\tilde{\mathbf{F}} = \mathbf{R} \cdot \mathbf{F}\,,$$

where $\mathbf{R} = \mathbf{I}$ instantaneously, but its variation is given by

$$\delta\mathbf{R} = \widehat{\delta\boldsymbol{\phi}} \cdot \mathbf{R}\,,$$

where $\widehat{\delta\boldsymbol{\phi}}$ is skew-symmetric with components

$$\left[\widehat{\delta\phi}\right] = \begin{bmatrix} 0 & \delta\phi & 0 \\ -\delta\phi & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

with respect to the basis $\mathbf{e}_r$, $\mathbf{e}_z$, and $\mathbf{e}_\theta$ at $\Theta = 0$.

With this modification the corotational virtual deformation gradient is given by

$$\delta\tilde{\mathbf{F}} = \delta\mathbf{R} \cdot \mathbf{F} + \mathbf{R} \cdot \delta\mathbf{F} = \widehat{\delta\boldsymbol{\phi}} \cdot \mathbf{F} + \delta\mathbf{F}\,,$$

and the corotational virtual velocity gradient by

$$\delta\tilde{\mathbf{L}} = \delta\tilde{\mathbf{F}} \cdot \tilde{\mathbf{F}}^{-1} = \widehat{\delta\boldsymbol{\phi}} + \delta\mathbf{L}\,,$$

or

$$\begin{aligned}\delta\tilde{\mathbf{L}} = &\frac{\partial\delta r}{\partial r}\mathbf{e}_r\mathbf{e}_r + \frac{\partial\delta r}{\partial z}\mathbf{e}_r\mathbf{e}_z + \frac{\partial\delta z}{\partial r}\mathbf{e}_z\mathbf{e}_r + \frac{\partial\delta z}{\partial z}\mathbf{e}_z\mathbf{e}_z + \\ &r\frac{\partial\delta\theta}{\partial r}\mathbf{e}_\theta\mathbf{e}_r + r\frac{\partial\delta\theta}{\partial z}\mathbf{e}_\theta\mathbf{e}_z + \frac{\delta r}{r}\mathbf{e}_\theta\mathbf{e}_\theta\,.\end{aligned} \tag{3.2.8–5}$$

The modified virtual rate of deformation tensor and spin are simply

$$\delta\tilde{\mathbf{W}} = \frac{1}{2}\left(\delta\tilde{\mathbf{L}} + \delta\tilde{\mathbf{L}}^T\right)\,.$$

Stiffness in the current state

As shown in "Procedures: overview and basic equations," Section 2.1.1, the contribution of the internal work terms to the Jacobian of the Newton method that is used in Abaqus/Standard for solid element formulations is

$$d\delta\Pi = \int_V (d\boldsymbol{\sigma} : \delta\tilde{\mathbf{D}} + \boldsymbol{\sigma} : d\delta\tilde{\mathbf{D}})\, dV.$$

The second variation in $\delta\tilde{\mathbf{D}}$ is obtained as

$$d\delta\tilde{\mathbf{D}} = \text{sym}(d\delta\tilde{\mathbf{F}} \cdot \tilde{\mathbf{F}}^{-1} - \delta\tilde{\mathbf{L}} \cdot d\tilde{\mathbf{L}}),$$

where $d\tilde{\mathbf{L}}$ has the same form as $\delta\tilde{\mathbf{L}}$ in Equation 3.2.8–5. Moreover, in this formulation $d\delta\tilde{\mathbf{F}}$ is nonzero, and it can be shown with the aid of "Rotation variables," Section 1.3.1, that $d\delta\tilde{\mathbf{F}} \cdot \tilde{\mathbf{F}}^{-1}$ has the form

$$d\delta\tilde{\mathbf{F}} \cdot \tilde{\mathbf{F}}^{-1} = \frac{1}{2}(\widehat{\delta\boldsymbol{\phi}} \cdot \widehat{d\boldsymbol{\phi}} + \widehat{d\boldsymbol{\phi}} \cdot \widehat{\delta\boldsymbol{\phi}}) + \widehat{\delta\boldsymbol{\phi}} \cdot d\tilde{\mathbf{L}} + \widehat{d\boldsymbol{\phi}} \cdot \delta\tilde{\mathbf{L}} + d\delta\mathbf{F} \cdot \mathbf{F}^{-1}\,.$$

In component form,

$$d\delta\tilde{\mathbf{F}} \cdot \tilde{\mathbf{F}}^{-1} = \left(\frac{\partial d\theta}{\partial r}\delta r + dr\frac{\partial\delta\theta}{\partial r}\right)\mathbf{e}_\theta\mathbf{e}_r + \left(\frac{\partial d\theta}{\partial z}\delta r + dr\frac{\partial\delta\theta}{\partial z}\right)\mathbf{e}_\theta\mathbf{e}_z\,.$$

Introduction of the corotational stress rate

$$d\check{\boldsymbol{\sigma}} = d\boldsymbol{\sigma} - d\tilde{\mathbf{W}} \cdot \boldsymbol{\sigma} - \boldsymbol{\sigma} \cdot d\tilde{\mathbf{W}}^T$$

yields the more familiar form

$$d\delta\Pi = \int (d\check{\boldsymbol{\sigma}} : \delta\tilde{\mathbf{D}} + \boldsymbol{\sigma} : (\delta\tilde{\mathbf{L}}^T \cdot d\tilde{\mathbf{L}} - 2d\tilde{\mathbf{D}} + d\delta\tilde{\mathbf{F}} \cdot \tilde{\mathbf{F}}^{-1}))dV.$$

Reference

- “Axisymmetric solid element library,” Section 28.1.6 of the Abaqus Analysis User’s Guide

3.2.9 AXISYMMETRIC ELEMENTS ALLOWING NONLINEAR BENDING

Product: Abaqus/Standard

Abaqus/Standard includes a library of solid elements whose geometry is initially axisymmetric and that allow for nonlinear analysis in which bending can occur about the plane $\theta = \pi/2$ in the (r, z, θ) cylindrical coordinate system of the model. The geometric model is defined in the r–z plane only. The displacements are the usual isoparametric interpolations with respect to r and z, augmented by Fourier expansions with respect to θ. Since the elements are written for bending about the plane $\theta = \pi/2$ only, they cannot be used to model torsion of the structure about the original axis of symmetry. Because the elements are intended for nonlinear applications, the orthogonality properties associated with Fourier modes cannot be used to reduce the problem to a series of smaller, uncoupled, cases, since the stiffness before projection onto the Fourier modes is not necessarily constant. For this reason these elements are significantly more expensive to use than the corresponding axisymmetric elements intended for axisymmetric deformations.

Interpolation

The coordinate system used with these elements is the cylindrical system (r, z, θ), where r measures the distance of a point from the axis of the cylindrical system, z measures its position along this axis, and θ measures the angle between the plane containing the point and the axis of the coordinate system and some fixed reference plane that contains the coordinate system axis. The order in which the coordinates and displacements are taken in these elements is based on the convention used in Abaqus for axisymmetric elements, so that z is the second coordinate. This allows these elements to be used in conjunction with other elements in the library that allow only axisymmetric deformation. This order is not the same as that used in three-dimensional elements in Abaqus in which z is the third coordinate, nor is it the order (r, θ, z), usually taken in cylindrical systems.

The original geometry of the elements is assumed to be axisymmetric with respect to the axis of the coordinate system and, thus, independent of θ. Let $\mathbf{e}_r$, $\mathbf{e}_z$, and $\mathbf{e}_\theta$ be unit vectors in the radial, axial, and circumferential directions at a point in the undeformed state. The reference position $\mathbf{X}$ of the point can be represented in terms of the original radius R and the axial position Z:

$$\mathbf{X} = R\mathbf{e}_r + Z\mathbf{e}_z.$$

Similarly the displacement $\mathbf{u}$ of the point can be represented in terms of the components u_r, u_z, and u_θ with respect to these same vectors at the original position of the point:

$$\mathbf{u} = u_r\mathbf{e}_r + u_z\mathbf{e}_z + u_\theta\mathbf{e}_\theta.$$

For small radial and circumferential displacements the circumferential displacement is proportional to the change in circumferential angle ($u_\theta = R\,\Delta\theta$), but for large displacements this relation becomes nonlinear ($u_\theta = (R + u_r)\tan(\Delta\theta)$), as shown in Figure 3.2.9–1. The distinction is of importance only in geometrically nonlinear analysis with radially applied concentrated loads and/or sliding radial

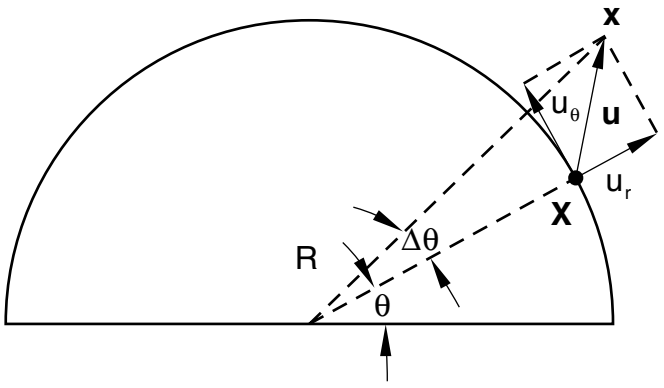


Figure 3.2.9–1 Displacement and rotation in the r–θ plane.

boundary conditions.This definition of the degrees of freedom is equivalent to applying transformations to the global (x, y, z) degrees of freedom associated with a standard continuum element. Hence, the nonlinear equations associated with these elements have the same structure as the equations for standard continuum elements.

A general interpolation scheme for $\mathbf{u}$ using Fourier terms with respect to θ is used:

$$u_\alpha = \sum_{m=1}^{M} H^m(g,h) \left(u_\alpha^{m0} + \sum_{p=1}^{P} (\cos p\theta \, u_{\alpha c}^{mp} + \sin p\theta \, u_{\alpha s}^{mp}) \right), \quad (\alpha = r, z, \theta)$$

where g, h are isoparametric coordinates in the original R–Z plane; H^m are polynomial interpolation functions; and u_α^{m0}, $u_{\alpha c}^{mp}$, and $u_{\alpha s}^{mp}$ are solution amplitude values. M is the number of terms used for interpolation with respect to g, h; and P is the number of terms used in the Fourier interpolation with respect to θ. Purely axisymmetric deformation results when $P = 0$.

We reduce the number of variables in such an element by assuming that bending is allowed only about one plane, $\theta = \pi/2$, so that the plane $\theta = 2n\pi$, n integer, is a plane of symmetry. The only terms that satisfy this condition are

$$\begin{Bmatrix} u_r \\ u_z \\ u_\theta \end{Bmatrix} = \sum_{m=1}^{M} H^m(g,h) \left(\begin{Bmatrix} u_r^{m0} \\ u_z^{m0} \\ 0 \end{Bmatrix} + \sum_{p=1}^{P} \cos p\theta \begin{Bmatrix} u_{rc}^{mp} \\ u_{zc}^{mp} \\ 0 \end{Bmatrix} + \sum_{p=1}^{P} \sin p\theta \begin{Bmatrix} 0 \\ 0 \\ u_{\theta s}^{mp} \end{Bmatrix} \right). \qquad (3.2.9\text{–}1)$$

For convenience we use the values of the u_r and u_z displacement components at specific locations around the model between $\theta = 0$ and $\theta = \pi$ instead of the Fourier amplitudes u_r^{m0} and u_{rc}^{mp}. The main reason for this is to allow the elements to be used with interface elements, such as slide lines, for which physical displacement values are required. This is accurate only if it is assumed that the relative displacements in the θ-direction are small so that the interface conditions are considered with respect to u_r and u_z only; that

is, in planes of constant θ. In addition, we omit the subscript s in the expression for the circumferential displacement: $u_{\theta s}^{mp} \rightarrow u_{\theta}^{mp}$. Equation 3.2.9–1 is, therefore, rewritten

$$\left\{ \begin{array}{c} u_r \\ u_z \\ u_\theta \end{array} \right\} = \sum_{m=1}^{M} H^m(g,h) \left(\sum_{p=1}^{P+1} R^p(\theta) \left\{ \begin{array}{c} u_r^{mp} \\ u_z^{mp} \\ 0 \end{array} \right\} + \sum_{p=1}^{P} \sin p\theta \left\{ \begin{array}{c} 0 \\ 0 \\ u_\theta^{mp} \end{array} \right\} \right), \tag{3.2.9–2}$$

where $R^p(\theta)$ are trigonometric interpolation functions and u_r^{mp}, u_z^{mp} are physical radial and axial displacement components at $\theta = \pi\,(p-1)/P$.

The R^p interpolators at the associated positions θ_p are taken as

$P = 1$:

$$R^1 = \frac{1}{2}(1 + \cos\theta)$$
$$R^2 = \frac{1}{2}(1 - \cos\theta)$$

and

$$\theta^1 = 0$$
$$\theta^2 = \pi$$

$P = 2$:

$$R^1 = \frac{1}{4}(1 + 2\cos\theta + \cos 2\theta)$$
$$R^2 = \frac{1}{2}(1 - \cos 2\theta)$$
$$R^3 = \frac{1}{4}(1 - 2\cos\theta + \cos 2\theta)$$

and

$$\theta^1 = 0$$
$$\theta^2 = \frac{1}{2}\pi$$
$$\theta^3 = \pi$$

$P = 3$:

$$R^1 = \frac{1}{6}(1 + 2\cos\theta + 2\cos 2\theta + \cos 3\theta)$$
$$R^2 = \frac{1}{3}(1 + \cos\theta - \cos 2\theta - \cos 3\theta)$$

$$R^3 = \frac{1}{3}(1 - \cos\theta - \cos 2\theta + \cos 3\theta)$$
$$R^4 = \frac{1}{6}(1 - 2\cos\theta + 2\cos 2\theta - \cos 3\theta)$$

and

$$\theta^1 = 0$$
$$\theta^2 = \frac{1}{3}\pi$$
$$\theta^3 = \frac{2}{3}\pi$$
$$\theta^4 = \pi$$

$P = 4$:

$$R^1 = \frac{1}{8}(1 + 2\cos\theta + 2\cos 2\theta + 2\cos 3\theta + \cos 4\theta)$$
$$R^2 = \frac{1}{4}(1 + \sqrt{2}\cos\theta - \sqrt{2}\cos 3\theta - \cos 4\theta)$$
$$R^3 = \frac{1}{4}(1 - 2\cos 2\theta + \cos 4\theta)$$
$$R^4 = \frac{1}{4}(1 - \sqrt{2}\cos\theta + \sqrt{2}\cos 3\theta - \cos 4\theta)$$
$$R^5 = \frac{1}{8}(1 - 2\cos\theta + 2\cos 2\theta - 2\cos 3\theta + \cos 4\theta)$$

and

$$\theta^1 = 0$$
$$\theta^2 = \frac{1}{4}\pi$$
$$\theta^3 = \frac{1}{2}\pi$$
$$\theta^4 = \frac{3}{4}\pi$$
$$\theta^5 = \pi$$

$P = 4$ is the highest-order interpolation offered with respect to θ in these elements: the elements become significantly more expensive as higher-order interpolation is used; and it is assumed that, because of this, full three-dimensional modeling is less expensive than using these elements with $P > 4$.

Integration

The integration scheme used in these elements is a product of integration with respect to element coordinates in surfaces that were originally in the R–Z plane and integration with respect to θ. For the

former the same scheme is used as in the corresponding purely axisymmetric elements (for example, either full or reduced Gauss integration in the isoparametric quadrilaterals). For integration with respect to θ the trapezoidal rule is used, with the number of integration points set to $2(P+1)$.

Deformation gradient

For a material point in space the deformation gradient $\mathbf{F}$ is defined as the gradient of the current position $\mathbf{x}$ with respect to the original position $\mathbf{X}$:

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}.$$

The current position $\mathbf{x}$ can be described in terms of the original position $\mathbf{X}$ and the displacement $\mathbf{u}$,

$$\mathbf{x} = \mathbf{X} + \mathbf{u} = (R + u_r)\mathbf{e}_r + (Z + u_z)\mathbf{e}_z + u_\theta \mathbf{e}_\theta,$$

and the gradient operator can be described in terms of partial derivatives with respect to the cylindrical coordinates:

$$\frac{\partial}{\partial \mathbf{X}} = \mathbf{e}_r \frac{\partial}{\partial R} + \mathbf{e}_z \frac{\partial}{\partial Z} + \mathbf{e}_\theta \frac{\partial}{R \partial \theta}.$$

Since the radial and circumferential base vectors depend on the original circumferential coordinate θ: $\mathbf{e}_r = \mathbf{e}_r(\theta)$, $\mathbf{e}_\theta = \mathbf{e}_\theta(\theta)$, the partial derivatives of these base vectors with respect to θ are nonvanishing:

$$\frac{\partial \mathbf{e}_r}{\partial \theta} = \mathbf{e}_\theta, \qquad \frac{\partial \mathbf{e}_\theta}{\partial \theta} = -\mathbf{e}_r.$$

With this result the deformation gradient is obtained as

$$\begin{aligned}\mathbf{F} = {} & \mathbf{e}_r \mathbf{e}_r \left(1 + \frac{\partial u_r}{\partial R}\right) + \mathbf{e}_r \mathbf{e}_z \frac{\partial u_r}{\partial Z} + \mathbf{e}_r \mathbf{e}_\theta \left(\frac{\partial u_r}{R \partial \theta} - \frac{u_\theta}{R}\right) \\ & + \mathbf{e}_z \mathbf{e}_r \frac{\partial u_z}{\partial R} + \mathbf{e}_z \mathbf{e}_z \left(1 + \frac{\partial u_z}{\partial Z}\right) + \mathbf{e}_\theta \mathbf{e}_z \frac{\partial u_z}{R \partial \theta} \\ & + \mathbf{e}_\theta \mathbf{e}_r \frac{\partial u_\theta}{\partial R} + \mathbf{e}_\theta \mathbf{e}_z \frac{\partial u_\theta}{\partial Z} + \mathbf{e}_\theta \mathbf{e}_\theta \left(1 + \frac{u_r}{R} + \frac{\partial u_\theta}{R \partial \theta}\right).\end{aligned}$$

Alternatively, this can be written in matrix form with components relative to the local reference basis $\mathbf{e}_r$, $\mathbf{e}_z$, $\mathbf{e}_\theta$:

$$[F] = \begin{bmatrix} 1 + \partial u_r/\partial R & \partial u_r/\partial Z & \partial u_r/R\partial\theta - u_\theta/R \\ \partial u_z/\partial R & 1 + \partial u_z/\partial Z & \partial u_z/R\partial\theta \\ \partial u_\theta/\partial R & \partial u_\theta/\partial Z & 1 + u_r/R + \partial u_\theta/R\partial\theta \end{bmatrix}. \qquad (3.2.9\text{–}3)$$

To be able to analyze approximately incompressible material behavior, the volume change in the fully integrated 4-node quadrilaterals is assumed to be independent of g and h in an R–Z plane. Hence, $[F]$ is modified according to

$$[\overline{F}] = \left(\frac{J_0}{J}\right)^{\frac{1}{3}} [F],$$

where $J = \det[F]$ is the volume change at the integration point and J_0 is the average volume change over the R–Z plane of the element. In addition, the part of the axisymmetric hoop strain that does not depend on θ is made independent of g and h. Experience has shown this considerably improves the solution accuracy for axisymmetric problems. Thus, we use

$$\overline{F}_{\theta\theta} = 1 + \overline{u_r/R} + \partial u_\theta / R\partial\theta,$$

where

$$\overline{u_r/R} = \left[\sum_{m=1}^{M} \frac{H^m(0,0)}{R(0,0)} \sum_{p=1}^{P+1} C^p + \sum_{m=1}^{M} \frac{H^m(g,h)}{R(g,h)} \sum_{p=1}^{P+1} (R^p(\theta) - C^p)\right] u_r^{mp},$$

with C^p the leading (constant) term in $R^p(\theta)$.

Strain and rotation increments

Strain and rotation increments are calculated from the integrated velocity gradient matrix, $\Delta\mathbf{L}$, defined as

$$\Delta\mathbf{L} = \frac{\partial \Delta\mathbf{u}}{\partial \mathbf{x}_{t+\Delta t/2}},$$

where $\mathbf{x}_{t+\Delta t/2} = \mathbf{x}_t + \frac{1}{2}\Delta\mathbf{u} = \mathbf{x}_{t+\Delta t} - \frac{1}{2}\Delta\mathbf{u}$. This expression is not easily evaluated directly, since points that were in an R–Z plane in the undeformed shape will no longer be located in the same plane after deformation. Instead, we calculate the gradient of $\Delta\mathbf{u}$ with respect to the reference state and obtain $\Delta\mathbf{L}$ with the transformation

$$\Delta\mathbf{L} = \frac{\partial \Delta\mathbf{u}}{\partial \mathbf{X}} \cdot \frac{\partial \mathbf{X}}{\partial \mathbf{x}_{t+\Delta t/2}} = \frac{\partial \Delta\mathbf{u}}{\partial \mathbf{X}} \cdot \left(\frac{\partial \mathbf{x}_{t+\Delta t/2}}{\partial \mathbf{X}}\right)^{-1} = \frac{\partial \Delta\mathbf{u}}{\partial \mathbf{X}} \cdot \left(\mathbf{F} - \frac{1}{2}\frac{\partial \Delta\mathbf{u}}{\partial \mathbf{X}}\right)^{-1}.$$

In matrix form this can be written as

$$[\Delta L] = [\partial \Delta u/\partial X] \left[[F] - \frac{1}{2}[\partial \Delta u/\partial X]\right]^{-1}, \tag{3.2.9–4}$$

with

$$\left[\partial\Delta u/\partial X\right] = \begin{bmatrix} \partial\Delta u_r/\partial R & \partial\Delta u_r/\partial Z & \partial\Delta u_r/R\partial\theta - \Delta u_\theta/R \\ \partial\Delta u_z/\partial R & \partial\Delta u_z/\partial Z & \partial\Delta u_z/R\partial\theta \\ \partial\Delta u_\theta/\partial R & \partial\Delta u_\theta/\partial Z & \partial\Delta u_\theta/R\partial\theta + \Delta u_r/R \end{bmatrix}. \tag{3.2.9–5}$$

The strain increments are approximated as the symmetric part of $\Delta\mathbf{L}$:

$$\left[\Delta\varepsilon\right] = \frac{1}{2}\left[\left[\Delta L\right] + \left[\Delta L\right]^T\right].$$

As was the case for the deformation gradient, we modify the volume strain increment in the fully integrated 4-node quadrilaterals to be independent of $\boldsymbol{g}$ and $\boldsymbol{h}$ in an $\boldsymbol{R}$–$\boldsymbol{Z}$ plane, which yields

$$\left[\Delta\overline{\varepsilon}\right] = \frac{1}{2}\left[\left[\Delta\overline{L}\right] + \left[\Delta\overline{L}\right]^T\right], \quad \text{with} \quad \left[\Delta\overline{L}\right] = \left[\Delta L\right] + \frac{1}{3}\left(\mathrm{tr}(\Delta L^0) - \mathrm{tr}(\Delta L)\right)\left[I\right],$$

where $\left[I\right]$ is the unit matrix, $\mathrm{tr}(\Delta L) = \Delta L_{rr} + \Delta L_{zz} + \Delta L_{\theta\theta}$ is the volume strain increment at the integration point, and $\mathrm{tr}(\Delta L^0)$ is the average volume strain increment over the $\boldsymbol{R}$–$\boldsymbol{Z}$ plane of the element. In addition we use the approximation

$$\overline{\Delta u_r/R} = \left[\sum_{m=1}^{M}\frac{H^m(0,0)}{R(0,0)}\sum_{p=1}^{P+1}C^p + \sum_{m=1}^{M}\frac{H^m(g,h)}{R(g,h)}\sum_{p=1}^{P+1}\left(R^p(\theta) - C^p\right)\right]\Delta u_r^{mp}.$$

The spin increments, $\Delta\overline{\boldsymbol{\Omega}}$, are approximated as the antisymmetric part of $\Delta\overline{\mathbf{L}}$, which in matrix form becomes

$$\left[\Delta\overline{\Omega}\right] = \frac{1}{2}\left[\left[\Delta\overline{L}\right] - \left[\Delta\overline{L}\right]^T\right].$$

Virtual work

The formulation of equilibrium (virtual work) requires linearization of the strain-displacement relation in the current state. For fully integrated 4-node elements the volume strain modification provides

$$\delta\overline{\mathbf{D}} = \frac{1}{2}\left(\delta\overline{\mathbf{L}} + \delta\overline{\mathbf{L}}^T\right),$$

where

$$\delta\overline{\mathbf{L}} = \delta\mathbf{L} + \frac{1}{3}\left[\mathrm{tr}(\delta\mathbf{L}^0) - \mathrm{tr}(\delta\mathbf{L})\right]\mathbf{I}$$

and

$$\delta\mathbf{L} = \frac{\partial\delta\mathbf{u}}{\partial\mathbf{x}}.$$

As was the case for the strain increments, the linearized strain-displacement relation involves taking derivatives in the deformed shape ($\mathbf{x} = \mathbf{x}_{t+\Delta t}$), which is troublesome since points that were in an R–Z plane in the undeformed shape will no longer be located in the same plane. Hence, $\delta\mathbf{L}$ is computed in a similar manner to $\Delta\mathbf{L}$:

$$\delta\mathbf{L} = \frac{\partial \delta\mathbf{u}}{\partial \mathbf{X}} \cdot \frac{\partial \mathbf{X}}{\partial \mathbf{x}} = \frac{\partial \delta\mathbf{u}}{\partial \mathbf{X}} \cdot \left(\frac{\partial \mathbf{x}}{\partial \mathbf{X}} \right)^{-1} = \frac{\partial \delta\mathbf{u}}{\partial \mathbf{X}} \cdot \mathbf{F}^{-1}.$$

In matrix form this can be written as

$$\left[\delta L\right] = \left[\partial \delta u / \partial X\right] \left[F\right]^{-1}, \tag{3.2.9–6}$$

where

$$\left[\partial \delta u / \partial X\right] = \begin{bmatrix} \partial \delta u_r / \partial R & \partial \delta u_r / \partial Z & \partial \delta u_r / R \partial \theta - \delta u_\theta / R \\ \partial \delta u_z / \partial R & \partial \delta u_z / \partial Z & \partial \delta u_z / R \partial \theta \\ \partial \delta u_\theta / \partial R & \partial \delta u_\theta / \partial Z & \partial \delta u_\theta / R \partial \theta + \delta u_r / R \end{bmatrix}. \tag{3.2.9–7}$$

For fully integrated, 4-node quadrilaterals we again use the approximation

$$\overline{\delta u_r / R} = \left[\sum_{m=1}^{M} \frac{H^m(0,0)}{R(0,0)} \sum_{p=1}^{P+1} C^p + \sum_{m=1}^{M} \frac{H^m(g,h)}{R(g,h)} \sum_{p=1}^{P+1} \left(R^p(\theta) - C^p \right) \right] \delta u_r^{mp}.$$

The displacements and, hence, the displacement variations, are interpolated in terms of nodal displacement variations with Equation 3.2.9–2. The derivatives of the displacements with respect to R, Z, and θ are readily obtained from these expressions:

$$\begin{Bmatrix} \partial u_r / \partial R \\ \partial u_z / \partial R \\ \partial u_\theta / \partial R \end{Bmatrix} = \sum_{m=1}^{M} \frac{\partial H^m}{\partial R} \left(\sum_{p=1}^{P+1} R^p(\theta) \begin{Bmatrix} u_r^{mp} \\ u_z^{mp} \\ 0 \end{Bmatrix} + \sum_{p=1}^{P} \sin p\theta \begin{Bmatrix} 0 \\ 0 \\ u_\theta^{mp} \end{Bmatrix} \right),$$

$$\begin{Bmatrix} \partial u_r / \partial Z \\ \partial u_z / \partial Z \\ \partial u_\theta / \partial Z \end{Bmatrix} = \sum_{m=1}^{M} \frac{\partial H^m}{\partial Z} \left(\sum_{p=1}^{P+1} R^p(\theta) \begin{Bmatrix} u_r^{mp} \\ u_z^{mp} \\ 0 \end{Bmatrix} + \sum_{p=1}^{P} \sin p\theta \begin{Bmatrix} 0 \\ 0 \\ u_\theta^{mp} \end{Bmatrix} \right),$$

$$\begin{Bmatrix} \partial u_r / \partial \theta \\ \partial u_z / \partial \theta \\ \partial u_\theta / \partial \theta \end{Bmatrix} = \sum_{m=1}^{M} H^m \left(\sum_{p=1}^{P+1} \frac{\partial R^p}{\partial \theta} \begin{Bmatrix} u_r^{mp} \\ u_z^{mp} \\ 0 \end{Bmatrix} + \sum_{p=1}^{P} p \cos p\theta \begin{Bmatrix} 0 \\ 0 \\ u_\theta^{mp} \end{Bmatrix} \right).$$

Stiffness in the current state

Since the elements are formulated in terms of Cartesian components of displacements, the equations presented in "Solid element formulation," Section 3.2.2, apply. For the 4-node quadrilaterals, we can adapt Equation 3.2.2–1 to the averaged volume change formulation, which yields

$$d\delta\Pi = \int_V (d\boldsymbol{\sigma} : \delta\overline{\mathbf{D}} + \boldsymbol{\sigma} : d\delta\overline{\mathbf{D}})\, dV.$$

The second variation in $\overline{\boldsymbol{\varepsilon}}$ is obtained with the standard procedure

$$\mathrm{d}\delta\overline{\boldsymbol{\varepsilon}} = -\mathrm{sym}(\delta\overline{\mathbf{L}} \cdot d\overline{\mathbf{L}}),$$

where $d\overline{\mathbf{L}}$ has the same form as $\delta\overline{\mathbf{L}}$. After introduction of the corotational stress rate, $\check{\boldsymbol{\sigma}}$ this yields

$$d\delta\Pi = \int_V (d\check{\boldsymbol{\sigma}} : \delta\overline{\boldsymbol{\varepsilon}} + \boldsymbol{\sigma} : (\delta\overline{\mathbf{L}}^T \cdot d\overline{\mathbf{L}} - 2\delta\overline{\mathbf{D}} \cdot d\overline{\mathbf{D}})dV.$$

This can be worked out in terms of nodal degrees of freedom with the expressions for $\delta\mathbf{L}$ and $\delta\overline{\boldsymbol{\varepsilon}}$ obtained in the previous paragraph on virtual work.

Hourglass control

In the 4-node reduced integration element the hourglass modes must be controlled. These modes are similar to the ones in regular axisymmetric elements but have some additional features.

1. The hourglass pattern can vary along the circumference, which requires application of an hourglass stiffness at multiple points around the circumference.
2. Hourglassing can also occur in the circumferential direction.

Hence, at each integration point around the circumference, we calculate the hourglass strains

$$\mathbf{q}(\theta) = \mathbf{x}_\gamma(\theta) \cdot \mathbf{F}(\theta) - \mathbf{X}_\gamma.$$

Here $\mathbf{F}$ is the deformation gradient as given by Equation 3.2.9–3 and $\mathbf{x}_\gamma$ and $\mathbf{X}_\gamma$ are the hourglass modes in the deformed and undeformed geometry respectively:

$$\mathbf{x}_\gamma(\theta) = \sum_{m=1}^{M} \gamma^m \mathbf{x}^m(\theta), \qquad \mathbf{X}_\gamma = \sum_{m=1}^{M} \gamma^m \mathbf{X}^m,$$

where γ^m is the same hourglass operator as used for the 4-node axisymmetric continuum elements and $\mathbf{x}^m(\theta)$ and $\mathbf{X}^m$ are the nodal positions at angle θ in the deformed and undeformed states. Observe that since the initial geometry is axisymmetric, $\mathbf{X}^m$ is independent of θ:

$$\mathbf{X}^m = \begin{Bmatrix} R^m \\ Z^m \\ 0 \end{Bmatrix}.$$

In the deformed state we write

$$\mathbf{x}^m(\theta) = \mathbf{X}^m + \mathbf{u}^m(\theta) = \begin{Bmatrix} R^m + u_r^m(\theta) \\ Z^m + u_z^m(\theta) \\ u_\theta^m(\theta) \end{Bmatrix}.$$

With Equation 3.2.9–2 this becomes

$$\mathbf{x}^m = \begin{Bmatrix} R^m \\ Z^m \\ 0 \end{Bmatrix} + \sum_{p=1}^{P+1} R^p(\theta) \begin{Bmatrix} u_r^{mp} \\ u_z^{mp} \\ 0 \end{Bmatrix} + \sum_{p=1}^{P} \sin\, p\theta \begin{Bmatrix} 0 \\ 0 \\ u_\theta^{mp} \end{Bmatrix}.$$

The hourglass "strain" transforms into an hourglass "force" with the hourglass stiffness c:

$$\mathbf{Q} = c\mathbf{q}.$$

This hourglass stiffness can be obtained with the same procedure as used for the regular axisymmetric elements, where the only difference is the scaling factor required to reflect the fact that each point reflects only part of the circumference. The first variation of $\mathbf{q}$ is readily obtained as

$$\delta\mathbf{q} = \delta\mathbf{x}_\gamma \cdot \mathbf{F} + \mathbf{x}_\gamma \cdot \delta\mathbf{F} = \delta\mathbf{u}_\gamma \cdot \mathbf{F} + \mathbf{x}_\gamma \cdot \frac{\partial\delta\mathbf{u}}{\partial\mathbf{X}}.$$

Here $\partial\delta\mathbf{u}/\partial\mathbf{X}$ follows from Equation 3.2.9–7, and for $\delta\mathbf{u}_\gamma$ we obtain

$$\delta\mathbf{u}_\gamma = \sum_{m=1}^{M} \gamma^m \delta\mathbf{u}^m = \sum_{m=1}^{M} \gamma^m \left(\sum_{p=1}^{P+1} R^p(\theta) \begin{Bmatrix} \delta u_r^{mp} \\ \delta u_z^{mp} \\ 0 \end{Bmatrix} + \sum_{p=1}^{P} \sin\, p\theta \begin{Bmatrix} 0 \\ 0 \\ \delta u_\theta^{mp} \end{Bmatrix} \right).$$

Similarly, we obtain for the second variation

$$\mathrm{d}\delta\mathbf{q} = \delta\mathbf{x}_\gamma \cdot \mathrm{d}\mathbf{F} + \mathrm{d}\mathbf{x}_\gamma \cdot \delta\mathbf{F} = \delta\mathbf{u}_\gamma \cdot \frac{\partial \mathrm{d}\mathbf{u}}{\partial\mathbf{X}} + \mathrm{d}\mathbf{u}_\gamma \cdot \frac{\partial\delta\mathbf{u}}{\partial\mathbf{X}},$$

where $\partial \mathrm{d}\mathbf{u}/\partial\mathbf{X}$ and $\mathrm{d}\mathbf{u}_\gamma$ follow with the same expressions as used in the first variation.

Pressure loads and load stiffness

For geometrically linear problems equivalent nodal loads due to applied surface pressures and body forces are readily calculated since the geometry is axisymmetric. For geometrically nonlinear problems the treatment of body forces does not change because of the fixed direction of the forces and because the forces are proportional to the volume, which is assumed to change by a negligible amount. However, for surface pressures nonaxisymmetric deformations must be taken into consideration.

The equivalent nodal loads associated with surface pressure p can be obtained by considering the virtual work contribution

$$\int_A p\mathbf{n} \cdot \delta\mathbf{u} dA = \int_0^{2\pi} \int_{-1}^{+1} p \left(\frac{\partial\mathbf{x}}{\partial\xi} \times \frac{\partial\mathbf{x}}{\partial\theta} \right) \cdot \delta\mathbf{x}\, d\xi d\theta,$$

where ξ is the parametric surface coordinate in the R–Z plane and

$$\mathbf{x} = r\mathbf{e}_r + z\mathbf{e}_z + u_\theta \mathbf{e}_\theta,$$

with $r = R + u_r$ and $z = Z + u_z$. Hence, the current position of a point can be expressed in terms of the surface interpolator $h^m(\xi)$ and the standard circumferential interpolators:

$$\left\{ \begin{array}{c} r \\ z \\ u_\theta \end{array} \right\} = \sum_{m=1}^{M} h^m(\xi) \left(\sum_{p=1}^{P+1} R^p(\theta) \left\{ \begin{array}{c} r^{mp} \\ z^{mp} \\ 0 \end{array} \right\} + \sum_{p=1}^{P} \sin p\theta \left\{ \begin{array}{c} 0 \\ 0 \\ u_\theta^{mp} \end{array} \right\} \right). \tag{3.2.9–8}$$

The terms in Equation 3.2.9–8 can be worked out as follows:

$$\begin{aligned} \frac{\partial \mathbf{x}}{\partial \xi} &= \frac{\partial r}{\partial \xi}\mathbf{e}_r + \frac{\partial z}{\partial \xi}\mathbf{e}_z + \frac{\partial u_\theta}{\partial \xi}\mathbf{e}_\theta, \\ \frac{\partial \mathbf{x}}{\partial \theta} &= \left(\frac{\partial r}{\partial \theta} - u_\theta \right) \mathbf{e_r} + \frac{\partial z}{\partial \theta}\mathbf{e}_z + \left(r + \frac{\partial u_\theta}{\partial \theta} \right) \mathbf{e}_\theta, \end{aligned}$$

and, hence,

$$\begin{aligned} \frac{\partial \mathbf{x}}{\partial \xi} \times \frac{\partial \mathbf{x}}{\partial \theta} = &\left[-\frac{\partial z}{\partial \xi} \left(r + \frac{\partial u_\theta}{\partial \theta} \right) + \frac{\partial u_\theta}{\partial \xi}\frac{\partial z}{\partial \theta} \right] \mathbf{e}_r + \\ &\left[-\frac{\partial u_\theta}{\partial \xi} \left(\frac{\partial r}{\partial \theta} - u_\theta \right) + \frac{\partial r}{\partial \xi} \left(r + \frac{\partial u_\theta}{\partial \theta} \right) \right] \mathbf{e}_z + \\ &\left[-\frac{\partial r}{\partial \xi}\frac{\partial z}{\partial \theta} + \frac{\partial z}{\partial \xi} \left(\frac{\partial r}{\partial \theta} - u_\theta \right) \right] \mathbf{e}_\theta. \end{aligned}$$

Hence, we obtain the virtual work contribution

$$\begin{aligned} \int_A p\,\mathbf{n} \cdot \delta\mathbf{u}\, dA = \int_0^{2\pi} \int_{-1}^{+1} -p \Bigg\{ &\left[\frac{\partial z}{\partial \xi} \left(r + \frac{\partial u_\theta}{\partial \theta} \right) - \frac{\partial u_\theta}{\partial \xi}\frac{\partial z}{\partial \theta} \right] \delta u_r + \\ &\left[\frac{\partial u_\theta}{\partial \xi} \left(\frac{\partial r}{\partial \theta} - u_\theta \right) - \frac{\partial r}{\partial \xi} \left(r + \frac{\partial u_\theta}{\partial \theta} \right) \right] \delta u_z + \\ &\left[\frac{\partial r}{\partial \xi}\frac{\partial z}{\partial \theta} - \frac{\partial z}{\partial \xi} \left(\frac{\partial r}{\partial \theta} - u_\theta \right) \right] \delta u_\theta \Bigg\} d\xi d\theta. \end{aligned}$$

With use of the interpolation functions we, thus, obtain the equivalent nodal forces:

$$F_r^{mp} = \int_0^{2\pi} \int_{-1}^{+1} -p\, h^m(\xi) R^p(\theta) \left[\frac{\partial z}{\partial \xi} \left(r + \frac{\partial u_\theta}{\partial \theta} \right) - \frac{\partial u_\theta}{\partial \xi}\frac{\partial z}{\partial \theta} \right] d\xi d\theta$$

$$F_z^{mp} = \int_0^{2\pi}\int_{-1}^{+1} -p\,h^m(\xi)R^p(\theta)\left[\frac{\partial u_\theta}{\partial\xi}\left(\frac{\partial r}{\partial\theta}-u_\theta\right)-\frac{\partial r}{\partial\xi}\left(r+\frac{\partial u_\theta}{\partial\theta}\right)\right]d\xi d\theta$$
$$F_\theta^{mp} = \int_0^{2\pi}\int_{-1}^{+1} -p\,h^m(\xi)\sin p\theta\left[\frac{\partial r}{\partial\xi}\frac{\partial z}{\partial\theta}-\frac{\partial z}{\partial\xi}\left(\frac{\partial r}{\partial\theta}-u_\theta\right)\right]d\xi d\theta.$$

For geometrically linear analysis this reduces to the standard axisymmetric equivalent nodal loads

$$F_r^{mp} = \int_0^{2\pi}\int_{-1}^{+1} -p\,h^m(\xi)R^p(\theta)\frac{\partial Z}{\partial\xi}\,d\xi R d\theta$$
$$F_z^{mp} = \int_0^{2\pi}\int_{-1}^{+1} p\,h^m(\xi)R^p(\theta)\frac{\partial R}{\partial\xi}\,d\xi R d\theta$$
$$F_\theta^{mp} = 0.$$

The load stiffness matrix follows by linearization:

$$-\begin{Bmatrix} \mathrm{d}F_r^{mp} \\ \mathrm{d}F_z^{mp} \\ \mathrm{d}F_\theta^{mp} \end{Bmatrix} = \begin{bmatrix} K_{rr}^{mpnq} & K_{rz}^{mpnq} & K_{r\theta}^{mpnq} \\ K_{zr}^{mpnq} & K_{zz}^{mpnq} & K_{z\theta}^{mpnq} \\ K_{\theta r}^{mpnq} & K_{\theta z}^{mpnq} & K_{\theta\theta}^{mpnq} \end{bmatrix} \begin{Bmatrix} \mathrm{d}u_r^{nq} \\ \mathrm{d}u_z^{nq} \\ \mathrm{d}u_\theta^{nq} \end{Bmatrix},$$

with

$$K_{rr}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1} p\,h^m(\xi)R^p(\theta)\frac{\partial z}{\partial\xi}\,h^n(\xi)R^q(\theta)\,d\xi d\theta$$

$$K_{zr}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1} p\,h^m(\xi)R^p(\theta)\left[-\frac{\partial r}{\partial\xi}h^n(\xi)R^q(\theta)-\left(r+\frac{\partial u_\theta}{\partial\theta}\right)\frac{dh^n}{d\xi}R^q(\theta) + \frac{\partial u_\theta}{\partial\xi}h^n(\xi)\frac{dR^q}{d\theta}\right]d\xi d\theta$$

$$K_{\theta r}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1} p\,h^m(\xi)\sin p\theta\left[\frac{\partial z}{\partial\theta}\frac{dh^n}{d\xi}R^q(\theta)-\frac{\partial z}{\partial\xi}h^n(\xi)\frac{dR^q}{d\theta}\right]d\xi d\theta$$

$$K_{rz}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1} p\,h^m(\xi)R^p(\theta)\left[\left(r+\frac{\partial u_\theta}{\partial\theta}\right)\frac{dh^n}{d\xi}R^q(\theta)-\frac{\partial u_\theta}{\partial\xi}h^n(\xi)\frac{dR^q}{d\theta}\right]d\xi d\theta$$

$$K_{zz}^{mpnq} = 0$$
$$K_{\theta z}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1} p\,h^m(\xi)\sin p\theta\left[\frac{\partial r}{\partial\xi}h^n(\xi)\frac{dR^q}{d\theta}-\left(\frac{\partial r}{\partial\theta}-u_\theta\right)\frac{dh^n}{d\xi}R^q(\theta)\right]d\xi d\theta$$

$$K_{r\theta}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1} p\,h^m(\xi)R^p(\theta)\left[q\,\frac{\partial z}{\partial\xi}h^n(\xi)\cos q\theta-\frac{\partial z}{\partial\theta}\frac{dh^n}{d\xi}\sin q\theta\right]d\xi d\theta$$

$$K_{z\theta}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1} p\,h^m(\xi)R^p(\theta)\left[-\frac{\partial u_\theta}{\partial \xi}h^n(\xi)\sin q\theta + \left(\frac{\partial r}{\partial \theta} - u_\theta\right)\frac{dh^n}{d\xi}\sin q\theta - q\,\frac{\partial r}{\partial \xi}h^n(\xi)\cos q\theta\right] d\xi d\theta$$

$$K_{\theta\theta}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1} p\,h^m(\xi)\sin p\theta\,\frac{\partial z}{\partial \xi}\,h^n(\xi)\sin q\theta\,d\xi d\theta.$$

In the case of hydrostatic pressure (***p*** dependent on ***z***) some additional terms appear. These terms are readily obtained from the expression

$$\int_A \mathrm{d}p\,\mathbf{n}\cdot\delta\mathbf{u}\,dA = \int_0^{2\pi}\int_{-1}^{+1} -\mathrm{d}u_z\frac{dp}{dz}\left\{\left[\frac{\partial z}{\partial \xi}\left(r + \frac{\partial u_\theta}{\partial \theta}\right) - \frac{\partial u_\theta}{\partial \xi}\frac{\partial z}{\partial \theta}\right]\delta u_r + \left[\frac{\partial u_\theta}{\partial \xi}\left(\frac{\partial r}{\partial \theta} - u_\theta\right) - \frac{\partial r}{\partial \xi}\left(r + \frac{\partial u_\theta}{\partial \theta}\right)\right]\delta u_z + \left[\frac{\partial r}{\partial \xi}\frac{\partial z}{\partial \theta} - \frac{\partial z}{\partial \xi}\left(\frac{\partial r}{\partial \theta} - u_\theta\right)\right]\delta u_\theta\right\} d\xi d\theta.$$

With use of the interpolation functions we, thus, obtain the additional load stiffness contributions:

$$K_{rz}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1}\frac{dp}{dz}\,h^m(\xi)R^p(\theta)\left[\frac{\partial z}{\partial \xi}\left(r + \frac{\partial u_\theta}{\partial \theta}\right) - \frac{\partial u_\theta}{\partial \xi}\frac{\partial z}{\partial \theta}\right]h^n(\xi)R^q(\theta)\,d\xi d\theta$$

$$K_{zz}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1}\frac{dp}{dz}\,h^m(\xi)R^p(\theta)\left[\frac{\partial u_\theta}{\partial \xi}\left(\frac{\partial r}{\partial \theta} - u_\theta\right) - \frac{\partial r}{\partial \xi}\left(r + \frac{\partial u_\theta}{\partial \theta}\right)\right]h^n(\xi)R^q(\theta)\,d\xi d\theta$$

$$K_{\theta z}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1}\frac{dp}{dz}\,h^m(\xi)\sin p\theta\left[\frac{\partial r}{\partial \xi}\frac{\partial z}{\partial \theta} - \frac{\partial z}{\partial \xi}\left(\frac{\partial r}{\partial \theta} - u_\theta\right)\right]h^n(\xi)R^q(\theta)\,d\xi d\theta.$$

Mass matrix

At each material point the displacement components in the three directions (radial, axial, circumferential) are dependent only on the corresponding nodal displacement components. Hence, the mass matrix does not involve any coupling between the radial, axial, and circumferential degrees of freedom, and we can write the mass matrix in the form of three separate expressions:

$$M_{rr}^{mnpq} = \int_V \rho\,N_r^{mp}N_r^{nq}\mathrm{d}V$$

$$M_{zz}^{mnpq} = \int_V \rho\,N_z^{mp}N_z^{nq}\mathrm{d}V$$

$$M_{\theta\theta}^{mnpq} = \int_V \rho\,N_\theta^{mp}N_\theta^{nq}\mathrm{d}V.$$

Here the superscripts m and n refer to a particular node in the r–z plane, and the superscripts p and q refer to a particular position along the circumference. The interpolation functions N_r^{mp}, N_z^{mp}, and N_θ^{mp} are the product of interpolation functions $H^m(g,h)$ in the r–z plane and interpolation functions in the θ-direction:

$$N_r^{mp} = H^m(g,h)R^p(\theta) = N_z^{mp}$$
$$N_\theta^{mp} = H^m(g,h)\sin\ p\theta.$$

The volume integral used to form the mass matrix can be split into an integral over the r–z cross-section and an integral around the circumference. For the r–r component of the mass matrix this yields

$$M_{rr}^{mnpq} = \int_A \rho\, H^m(g,h)H^n(g,h)\, 2\pi R\, \mathrm{d}A\, \frac{1}{2\pi}\int_0^{2\pi} R^p(\theta)R^q(\theta)\, \mathrm{d}\theta.$$

This matrix can be written in a convenient form by defining the primitive mass matrix,

$$M_{prim}^{mn} = \int_A \rho\, H^m(g,h)H^n(g,h)\, 2\pi R\, \mathrm{d}A.$$

This primitive mass matrix is the same mass matrix that is used for the regular axisymmetric elements. We can also define the circumferential distribution matrices

$$f_1^{pq} = \frac{1}{2\pi}\int_0^{2\pi} R^p(\theta)R^q(\theta)\, \mathrm{d}\theta \quad \text{and}$$
$$f_2^{pq} = \frac{1}{2\pi}\int_0^{2\pi} \sin p\theta\ \sin q\theta \mathrm{d}\theta.$$

The radial, axial, and circumferential components of the mass matrix then take the form

$$M_{rr}^{mnpq} = M_{prim}^{mn} f_1^{pq} = M_{zz}^{mnpq}$$
$$M_{\theta\theta}^{mnpq} = M_{prim}^{mn} f_2^{pq}.$$

The circumferential distribution matrices can be evaluated for various values of the number of terms P in the Fourier series. After some calculations the following results are obtained:

$P = 1$:

$$f_1^{pq} = \frac{1}{8}\begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}, \qquad f_2^{pq} = \frac{1}{2}\begin{pmatrix} 1 \end{pmatrix}.$$

$P = 2$:

$$f_1^{pq} = \frac{1}{32}\begin{pmatrix} 7 & 2 & -1 \\ 2 & 12 & 2 \\ -1 & 2 & 7 \end{pmatrix}, \qquad f_2^{pq} = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

$P = 3$:

$$f_1^{pq} = \frac{1}{72}\begin{pmatrix} 11 & 2 & -2 & 1 \\ 2 & 20 & 4 & -2 \\ -2 & 4 & 20 & 2 \\ 1 & -2 & 2 & 11 \end{pmatrix}, \qquad f_2^{pq} = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

$P = 4$:

$$f_1^{pq} = \frac{1}{128}\begin{pmatrix} 15 & 2 & -2 & 2 & -1 \\ 2 & 28 & 4 & -4 & 2 \\ -2 & 4 & 28 & 4 & -2 \\ 2 & -4 & 4 & 28 & 2 \\ -1 & 2 & -2 & 2 & 15 \end{pmatrix}, \qquad f_2^{pq} = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Hybrid and pore pressure elements

For hybrid and pore pressure elements additional degrees of freedom $\boldsymbol{p}$ are added. In the hybrid elements these degrees of freedom are internal to the element and represent the hydrostatic pressure in the material. In the pore pressure elements the degrees of freedom represent the hydrostatic pressure in the fluid as interpolated from the pressure variables at the external, user-defined nodes. Let the interpolation function for the (hydrostatic or pore) pressure in the r–z plane be denoted by $G^m(g,h)$. The interpolation functions are the same as for the regular axisymmetric hybrid and pore pressure elements, respectively. Along the circumference, we observe that in the geometrically linear formulation the volumetric strain ϕ only shows cosine dependence:

$$\phi = \frac{\partial u_r}{\partial R} + \frac{\partial u_z}{\partial Z} + \frac{u_r}{R} + \frac{\partial u_\theta}{R\partial\theta}.$$

Hence, we choose the hydrostatic/pore pressure to have cosine dependence only:

$$p(R,Z,\theta) = \sum_{m=1}^{M} G^m(g,h) \sum_{p=1}^{P=1} R^p(\theta)\, p^{mp}.$$

In the nonlinear case ϕ will exhibit higher-order variations in θ. For approximately incompressible materials, these higher-order terms are likely to lead to "locking" of the finite element mesh for nonaxisymmetric deformations. In the hybrid formulation, however, the higher-order terms in ϕ are not used for calculation of hydrostatic pressure: only the cosine terms as used in the interpolation for $\boldsymbol{p}$ are used. Hence, the hybrid elements prevent locking in the nonaxisymmetric modes as well as in the axisymmetric modes.

Pore pressure gradient

For a material point in space in the pore pressure element, the pore pressure gradient calculation involves taking derivatives of the pore pressure with respect to the current position $\mathbf{x}$. Again, we do not

evaluate the gradient directly but calculate it with respect to the original position $\mathbf{X}$, with the following transformation:

$$\frac{\partial p}{\partial \mathbf{x}} = \frac{\partial p}{\partial \mathbf{X}} \cdot \frac{\partial \mathbf{X}}{\partial \mathbf{x}} = \frac{\partial p}{\partial \mathbf{X}} \cdot \mathbf{F}^{-1},$$

where $\mathbf{F}$ is the deformation gradient, and the cylindrical components of the scalar gradient of the pore pressure with respect to $\mathbf{X}$ are readily obtained from the following expressions:

$$\frac{\partial p}{\partial R} = \sum_{m=1}^{M} \frac{\partial G^m}{\partial R} \sum_{p=1}^{P+1} R^p(\theta) p^{mp},$$
$$\frac{\partial p}{\partial Z} = \sum_{m=1}^{M} \frac{\partial G^m}{\partial Z} \sum_{p=1}^{P+1} R^p(\theta) p^{mp},$$
$$\frac{\partial p}{\partial \theta} = \sum_{m=1}^{M} G^m \sum_{p=1}^{P+1} \frac{\partial R^p(\theta)}{\partial \theta} p^{mp}.$$

Reference

- "Axisymmetric solid elements with nonlinear, asymmetric deformation," Section 28.1.7 of the Abaqus Analysis User's Guide

3.3 Infinite elements

- "Solid infinite elements," Section 3.3.1
- "Acoustic infinite elements," Section 3.3.2

3.3.1 SOLID INFINITE ELEMENTS

Products: Abaqus/Standard Abaqus/Explicit

The stress analyst is often faced with problems defined in unbounded domains or problems in which the region of interest is small compared with the surrounding medium. The unbounded or infinite medium can be approximated by extending the finite element mesh to a far distance, where the influence of the surrounding medium on the region of interest is considered small enough to be neglected. This approach calls for experimentation with mesh sizes and assumed boundary conditions at the truncated edges of the mesh and is not always reliable. It is particularly of concern in dynamic analysis, when the boundary of the mesh may reflect energy back into the region being modeled. A better approach is to use "infinite elements": elements defined over semi-infinite domains with suitably chosen decay functions. Abaqus provides first- and second-order infinite elements that are based on the work of Zienkiewicz et al. (1983) for static response and of Lysmer and Kuhlemeyer (1969) for dynamic response. The elements are used in conjunction with standard finite elements, which model the area around the region of interest, with the infinite elements modeling the far-field region.

Static analysis

The solution in the far field is assumed to be linear, so only linear behavior is provided in the infinite elements. The static behavior of the infinite elements is based on modeling the basic solution variable, u (in stress analysis u is a displacement component) with respect to spatial distance r measured from a "pole" of the solution, so that $u \to 0$ as $r \to \infty$, and $u \to \infty$ as $r \to 0$. The interpolation provides terms of order $1/r$, $1/r^2$, and, when the solution variable is a stress-like variable (such as the pore liquid pressure in the analysis of flow through a porous medium), $1/r^3$ as $r \to 0$. The far-field behavior of many common cases, such as a point load on a half-space, is thereby included. This modeling is achieved by using standard quadratic or cubic interpolation for $u(s)$ in $-1 \le s \le 1$, where s is a mapped coordinate that is chosen such that the mapping $r(s)$ causes $r \to \infty$ as $s \to 1$. We obtain two- and three-dimensional models of domains that reach to infinity by combining this interpolation in the s-direction in a product form with standard linear or quadratic interpolation in orthogonal directions in the mapped space.

In using infinite elements for static analysis, the pole must be located so as to provide a reasonable far-field solution for the particular problem being modeled. The infinite elements in Abaqus are written with nodes on the interface between the finite and infinite elements and, on each edge that stretches to infinity, a node that must be placed in the infinite direction such that the straight line from that node through the corresponding interface node passes through the pole for that ray at a distance on the other side of the interface from the infinite element equal to the distance between these nodes (Figure 3.3.1–1).

The one-dimensional concept is, thus, based on a node (node 1) on the interface between the finite and infinite elements, distance $r_1 = a$ from the pole and at $s = -1$ in the mapped space, and node 2, at $r_2 = 2a$ from the pole (the pole is at $r = 0$) and at $s = 0$ is the mapped space. The $r(s)$ mapping is chosen as

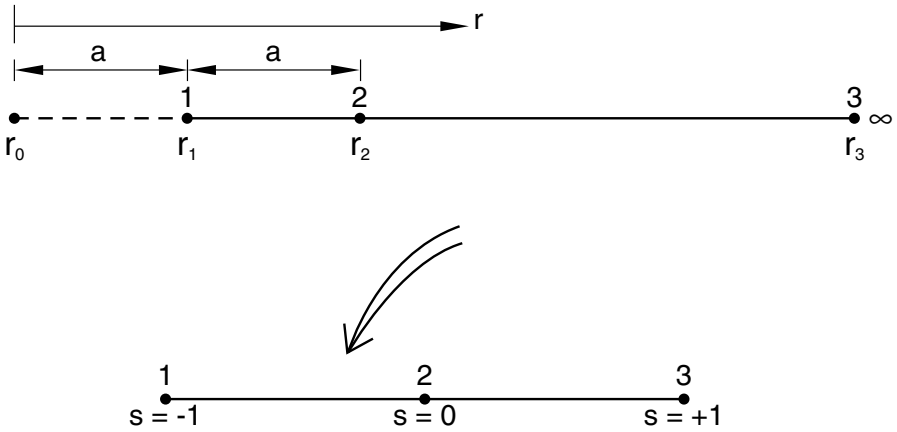


Figure 3.3.1–1 Pole node location for an infinite element.

$$r = -\frac{2s}{1-s} r_1 + \frac{1+s}{1-s} r_2,$$

so that

$$r = \frac{2a}{1-s},$$

which inverts to give

$$s = 1 - \frac{2a}{r}.$$

When an element with $1/r$ and $1/r^2$ behavior is required, we combine this geometric mapping with standard quadratic interpolation of u with respect to s, written in terms of its values at node 1 and at node 2:

$$u = \frac{1}{2} s(s-1) u_1 + (1-s^2) u_2$$

(this gives $u = 0$ at $s = 1$, where $r \to \infty$). Using the inverted geometric mapping to define $u(r)$ then gives

$$u = (-u_1 + 4u_2)\frac{a}{r} + (2u_1 - 4u_2)\left(\frac{a}{r}\right)^2,$$

which provides the desired behavior. Likewise, when $1/r^3$ behavior is also required, we use cubic interpolation of u with respect to s, written in terms of its values at nodes 1 and 2 and at a third node, which we choose to place at $s = \frac{1}{2}$:

$$u = -\frac{1}{3}s(s-1)(s-\frac{1}{2})u_1 - 2(1-s^2)(s-\frac{1}{2})u_2 + \frac{8}{3}s(1-s^2)u_3.$$

The inverted geometric mapping then provides

$$u = (\frac{1}{3}u_1 - 4u_2 + \frac{32}{3}u_3)\frac{a}{r} + (-2u_1 + 20u_2 - 32u_3)\left(\frac{a}{r}\right)^2 + (\frac{8}{3}u_1 - 16u_2 + \frac{64}{3}u_3)\left(\frac{a}{r}\right)^3.$$

The infinite elements in Abaqus consist of two- and three-dimensional elements for uncoupled stress analysis that use quadratic interpolation for displacement components and two- and three-dimensional elements for coupled stress-pore liquid pressure elements, in which the displacements use quadratic interpolation and the pore liquid pressure uses cubic interpolation in the infinite direction. This higher-order interpolation is used for the pore liquid pressure for compatibility: since the displacement varies as $1/r^2$, the strain (and, therefore, the stress) may vary as $1/r^3$.

Dynamic response

The dynamic response of the infinite elements is based on consideration of plane body waves traveling orthogonally to the boundary. Again, we assume the response adjacent to the boundary is of small enough amplitude so that the medium responds in a linear elastic fashion.

The equilibrium equation is

$$-\rho\ddot{\mathbf{u}} + \frac{\partial}{\partial \mathbf{x}} \cdot \boldsymbol{\sigma} = 0,$$

where ρ is the material's density, $\ddot{\mathbf{u}}$ is the material particle acceleration, $\boldsymbol{\sigma}$ is the stress, and $\mathbf{x}$ is position.

We assume the material's response is isotropic, linear elastic, and—thus—can be written as

$$\boldsymbol{\sigma} = \lambda \mathbf{I}\mathbf{I} : \boldsymbol{\varepsilon} + 2G\boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon}$ is the strain and

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$$

and

$$G = \frac{E}{2(1+\nu)}$$

are Lamé's constants (E is Young's modulus and ν is Poisson's ratio). Introducing this material response in the equilibrium equation, and assuming small strain:

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left\{\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \left[\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\right]^T\right\},$$

provides the governing equation for the motion

$$\rho\ddot{u}_i = G\frac{\partial^2 u_i}{\partial x_j \partial x_j} + (\lambda + G)\frac{\partial^2 u_j}{\partial x_i \partial x_j},$$

where index notation has been used for simplicity.

We consider plane waves traveling along the x-axis. Two body wave solutions of this form exist for this equation. One describes plane, longitudinal ("push") waves, which have the form

$$u_x = f(x \pm c_p t), \quad u_y = u_z = 0,$$

where, by substitution in the governing equation above, we find that the wave speed, c_p, is

$$c_p = \sqrt{\frac{\lambda + 2G}{\rho}}.$$

The other solution of this form is the "shear" wave solution

$$u_y = f(x \pm c_s t), \quad u_x = u_z = 0$$

or

$$u_z = f(x \pm c_s t), \quad u_x = u_y = 0,$$

where—again by substitution in the governing equation—we obtain

$$c_s = \sqrt{\frac{G}{\rho}}.$$

In each case the solution $f(x - ct)$ represents waves moving in the direction of increasing x, while $f(x + ct)$ represents waves moving in the direction of decreasing x.

Now consider a boundary at $x = L$ of a medium modeled by finite elements in $x < L$. We introduce distributed damping on this boundary, such that

$$\sigma_{xx} = -d_p \dot{u}_x$$

and

$$\sigma_{xy} = -d_s \dot{u}_y$$
$$\sigma_{xz} = -d_s \dot{u}_z,$$

where we will now choose the damping constants d_p and d_s to avoid reflection of longitudinal and shear wave energy back into the medium in $x < L$. Plane, longitudinal waves approaching the boundary have the form $u_x = f_1(x - c_p t)$, $u_y = u_z = 0$. If they are reflected at all as plane, longitudinal waves, their reflection will travel away from the boundary in some form $u_x = f_2(x + c_p t)$, $u_y = u_z = 0$. Since the problem is linear, superposition provides the total displacement $f_1 + f_2$, with corresponding stresses

$\sigma_{xx} = (\lambda + 2G)(f_1' + f_2')$, all other $\sigma_{ij} = 0$, and velocity $\dot{u}_x = -c_p(f_1' - f_2')$. For this solution to satisfy the damping behavior introduced on the boundary at $x = L$ requires

$$(\lambda + 2G - d_p c_p) f_1' + (\lambda + 2G + d_p c_p) f_2' = 0.$$

We can, therefore, ensure that $f_2 = 0$ (so that $f_2' = 0$) for any f_1 by choosing

$$d_p = \frac{\lambda + 2G}{c_p} = \rho c_p.$$

A similar argument for shear waves provides

$$d_s = \rho c_s.$$

These values of boundary damping are built into the infinite elements in Abaqus. From the above discussion we see that they transmit all normally impinging plane body waves exactly (provided that the material behavior close to the boundary is linear elastic). General problems involve nonplane body waves that do not impinge on the boundary from an orthogonal direction and may also involve Rayleigh surface waves and Love waves. Nevertheless, these "quiet" boundaries work quite well even for such general cases, provided that they are arranged so that the dominant direction of wave propagation is orthogonal to the boundary or, at free surfaces and interfaces where Rayleigh or Love waves are of concern, they are orthogonal to the surface (see, for example, Cohen and Jennings, 1983). As the boundaries are "quiet" rather than silent (perfect transmitters of all waveforms), and because the boundaries rely on the solution adjacent to them being linear elastic, they should be placed some reasonable distance from the region of main interest.

During dynamic response analysis following static preload (as is common in geotechnical applications), the traction provided by the infinite elements to the boundary of the finite element mesh consists of the constant stress obtained from the static response with the quiet boundary damping stress added. Since the elements have no stiffness during dynamic analysis, they allow a net rigid body motion to occur, which is usually not a significant effect.

Reference

- "Infinite elements," Section 28.3.1 of the Abaqus Analysis User's Guide

3.3.2 ACOUSTIC INFINITE ELEMENTS

Products: Abaqus/Standard Abaqus/Explicit

Problems in unbounded domains, or problems in which the region of interest is small compared to the surrounding medium, are common in acoustic analysis. For example, a submarine deep underwater may experience loads due to the fluid and radiate sound into the fluid, as if the ocean were infinitely large. The extent to which an exterior fluid may be considered "unbounded" or "infinite" depends on the number of wavelengths between the body or region of interest and the nearest boundary: the higher this number, the more likely that the influence of these boundaries is small enough to be neglected. For example, the effect of the surrounding fluid on a submarine in a relatively shallow harbor may be like an infinite medium at high frequencies, but the harbor bottom and free surface may exert an effect at lower frequencies. Similarly, a loudspeaker in air may radiate sound from the high-frequency tweeter as if the surrounding air were infinite, but the effects of the walls of the room may affect the radiation pattern of the low-frequency woofer.

Abaqus provides a range of features to model exterior fluid effects. All of the surrounding fluid may be modeled with finite elements; clearly, this is practical only if the extent of the surrounding medium is small. At the next level of sophistication, the user may model a small region of fluid and apply a simple radiation boundary condition to the terminating surface. These radiation boundary conditions are derived using simple models of waves passing through a boundary. Abaqus provides several alternative formulations of these boundary conditions (see "Coupled acoustic-structural medium analysis," Section 2.9.1, and "Acoustic, shock, and coupled acoustic-structural analysis," Section 6.10.1 of the Abaqus Analysis User's Guide). These radiation conditions do not add degrees of freedom to the system and do not affect the symmetry of the matrix.

Finally, acoustic infinite elements are provided, which allow the retained finite element fluid region to be even smaller, with similar accuracy. The acoustic infinite element formulation differs from the radiation boundary condition formulation in several key respects. In the infinite elements the infinite exterior is subdivided into elements, and a method of weighted residuals statement is enforced on the elements in a manner entirely analogous to the usual finite element method. Degrees of freedom, corresponding to interpolation functions in the infinite direction, are added to the overall matrix system. The method of weighted residuals used in Abaqus results in nonsymmetric infinite element matrices, so the relative cost of these elements is higher than that of a simple radiation boundary condition. However, the accuracy of the infinite elements is sufficiently high that the finite element region can be reduced considerably, offsetting the cost in many applications.

Transient formulation

The solution in the unbounded acoustic medium is assumed to be linear and governed by the same equations as the finite acoustic region:

$$\frac{1}{\mathrm{K}_f}\ddot{p} + \frac{\gamma}{\rho_f \mathrm{K}_f}\dot{p} - \frac{\partial}{\partial \mathbf{x}} \cdot \left(\frac{1}{\rho_f} \frac{\partial p}{\partial \mathbf{x}} \right) = 0.$$

γ is used here to denote the volumetric drag parameter. Consider the infinite exterior of a region of acoustic fluid bounded by a convex surface and a conventional finite element mesh defined on this surface. Each facet of this surface mesh, together with the normal vectors at the nodes, defines a subdivision of the infinite exterior that will be referred to as the "infinite element." Application of the method of weighted residuals results in a weak form of this equation over the infinite element volume:

$$\int_{V_f} \delta p \left(\frac{1}{\mathrm{K}_f} \ddot{p} + \frac{\gamma}{\rho_f \mathrm{K}_f} \dot{p} - \frac{\partial}{\partial \mathbf{x}} \cdot \left(\frac{1}{\rho_f} \frac{\partial p}{\partial \mathbf{x}} \right) \right) dV = 0.$$

This equation is formally identical to that used in the finite element region (see "Coupled acoustic-structural medium analysis," Section 2.9.1); however, the choice of basis functions for the weight, δp, and for the solution field, p, will be different. Selection of these functions has been the subject of considerable experimentation and analysis (for example, see Allik, 1991, and Burnett, 1994). In Abaqus considerations of accuracy, numerical stability, time-domain well-posedness, and economy have led to the selection of a form proposed by Astley (Astley, 1994). Here, the Fourier transform is applied to the equilibrium equation, and the functions for the solution field are chosen as tensor products of conventional finite element shape functions in the directions tangential to the terminating surface and polynomials in $1/r$, where $\boldsymbol{r}$ is a coordinate in the infinite direction, and a spatially oscillatory factor. The weights are chosen as complex conjugates of the solution field functions times a $1/r^2$ factor. This combination has shown to result in an element with several desirable properties. First, if the material properties are constant with respect to frequency in steady-state analysis, the mass, damping, and stiffness element matrices are constant as well; equivalently, in transient analysis the element results in a second-order differential equation for the pressure. Moreover, the damping matrix has positive eigenvalues, a necessary condition for well-posedness in transient analysis. Analytical investigations of the formulation demonstrate that the element captures the exact solutions for radiation impedances for modes of a sphere; the order of the spherical mode modeled exactly is equal to the order of the polynomial used in the infinite direction. The element integrals do not contain oscillatory kernels and can be evaluated using standard Gauss quadrature methods. Finally, the element can be formulated using the usual coordinate map due to Bettess (1984) so that arbitrary convex terminating surfaces can be used.

To continue with the derivation, we transform the weighted residual statement into the frequency domain:

$$\int_{V_f} \delta \tilde{p} \left(-\Omega^2 \frac{1}{\mathrm{K}_f} \tilde{p} + i\Omega \frac{\gamma}{\rho_f \mathrm{K}_f} \tilde{p} - \frac{\partial}{\partial \mathbf{x}} \cdot \left(\frac{1}{\rho_f} \frac{\partial \tilde{p}}{\partial \mathbf{x}} \right) \right) dV = 0. \qquad (3.3.2\text{–}1)$$

Now the weight and solution interpolation functions are defined as

$$\delta \tilde{p} = \delta p_i F \phi_i \left(g, h, \xi \right) e^{ik(r - r_0)},$$
$$\tilde{p} = p_j \phi_j \left(g, h, \xi \right) e^{-ik(r - r_0)},$$

where $g(\mathbf{x})$, $h(\mathbf{x})$, and $\xi(\mathbf{x})$ are the parent element coordinates; $F \equiv (r_0/r)^2$; $k \equiv \Omega/c_0$; $c_0^2 \equiv K_f/\rho_f$; and ϕ_i are element shape functions with indices varying over the number of degrees of freedom of the element. The mapping and shape functions are described below.

Inserting the shape functions into Equation 3.3.2–1 and integrating by parts, we obtain the following element equation:

$$\begin{aligned}
&\int_{V_f}\left(\phi_i\frac{\partial\phi_j}{\partial\mathbf{x}}\cdot\left(\frac{1}{\rho_f}\frac{\partial F}{\partial\mathbf{x}}\right)+F\frac{\partial\phi_i}{\partial\mathbf{x}}\cdot\left(\frac{1}{\rho_f}\frac{\partial\phi_j}{\partial\mathbf{x}}\right)\right)dVp_j\\
&\qquad -ik\int_{V_f}\left(\frac{\partial(r-r_0)}{\partial\mathbf{x}}\cdot\left(\frac{1}{\rho_f}\frac{\partial F}{\partial\mathbf{x}}\right)\phi_i\phi_j\right)dVp_j\\
&\qquad +ik\int_{V_f}\frac{F}{\rho_f}\frac{\gamma}{\sqrt{K_f\rho_f}}\phi_i\phi_j dVp_j\\
&-ik\int_{V_f}\frac{F}{\rho_f}\left(\frac{\partial(r-r_0)}{\partial\mathbf{x}}\cdot\frac{\partial\phi_i}{\partial\mathbf{x}}\phi_j-\phi_i\frac{\partial\phi_j}{\partial\mathbf{x}}\cdot\frac{\partial(r-r_0)}{\partial\mathbf{x}}\right)dVp_j\\
&\qquad -k^2\int_{V_f}\frac{F}{K_f}\left(1-\frac{\partial(r-r_0)}{\partial\mathbf{x}}\cdot\frac{\partial(r-r_0)}{\partial\mathbf{x}}\right)\phi_i\phi_j dVp_j=0.
\end{aligned}\tag{3.3.2–2}$$

This equation is clearly nonsymmetric, due to the fact that the weight and trial functions are not identical. Nevertheless, if the material properties are constant as a function of frequency, each element matrix corresponding to the terms above is constant as well. Gradients of density inside the element volume have been ignored in the formulation.

The element shape functions are defined as follows:

$$\phi_i(g,h,r)\equiv fb_\alpha B_\beta,$$

where

$$f\equiv\sqrt{\frac{r_0(g,h)}{r}}$$

in two spatial dimensions and

$$f\equiv\frac{r_0(g,h)}{r}$$

in three. This function serves to specify the minimum rate of decay of the acoustic field inside the infinite element domain. The subindex α ranges over the n nodes of the infinite element on the terminating surface, while the subindex β ranges over the number of functions used in the infinite direction. The function index i is equal to the subindex α for the first n functions, $i\equiv n+\alpha$ for the second n functions, and so on.

The functions $b_\alpha(g,h)$ are conventional two-dimensional shape functions (in three dimensions) or, with $h\equiv 0$, one-dimensional shape functions for axisymmetric or two-dimensional elements. The role of these functions is to specify the variation of the acoustic field in directions tangent to the terminating surface. The variation of the acoustic field in the infinite direction is given by the functions B_β, which are members of a set of ten ninth-order polynomials in $\xi\equiv 1-\frac{2r_0(g,h)}{r}$. The members of this set are constructed to correspond to the Legendre modes of a sphere. That is, if infinite elements are placed on a

sphere and if tangential refinement is adequate, an ith order acoustic infinite element will absorb waves associated with the $(i-1)$th Legendre mode. The first member of this set corresponds to the value of the acoustic pressures on the terminating surface; the other functions are generalized degrees of freedom. Specifically,

$$B_1 \equiv 1,$$

and

$$B_\beta \equiv (1+\xi)\hat{B}_\beta, \text{ for } \beta > 1,$$

where

$$\int_{-1}^{1} \hat{B}_\beta \hat{B}_\gamma (1+\xi)^2 f^2 F d\xi = 0, \text{ for } \beta \neq \gamma.$$

This last condition is enforced to improve conditioning of the element matrices at higher order and results in different functions in two and three spatial dimensions. All of the B_β are equal to zero at the terminating surface, except for B_1.

The coordinate map is described in part by the element shape functions, in the usual isoparametric manner, and in part by a singular function. Together they map the true, semi-infinite domain onto the parent element square or cube. To specify the map for a given element, we first define distances r_α between each infinite element node α on the terminating surface and the element's reference node, located at $\mathbf{x}_0$:

$$r_\alpha \equiv |\mathbf{x}_\alpha - \mathbf{x}_0|.$$

Intermediate points in the infinite direction are defined as offset replicas of the nodes on the terminating surface:

$$\mathbf{y}_\alpha \equiv \mathbf{x}_\alpha + \mathbf{n}_\alpha r_\alpha,$$

where $\mathbf{n}_\alpha$ is the "node ray" at the node. The "node ray" at a node is the unit vector in the direction of the line between the reference point and the node.

We use the interpolated reference distance on the terminating surface,

$$r_0(g,h) \equiv \sum_\alpha r_\alpha b_\alpha(g,h),$$

in the definition of the parent coordinate, ξ, corresponding to the infinite direction:

$$\xi \equiv 1 - \frac{2r_0(g,h)}{r},$$

where r is the distance between an arbitrary point in the infinite element volume and the reference point, x_0. With these definitions, the geometric map can be specified as

$$\mathbf{x}(g,h,\xi) \equiv \sum_{\alpha} b_\alpha(g,h)\left[\mathbf{x}_\alpha + \left(\frac{\xi+1}{\xi-1}\right) r_\alpha \mathbf{n}_\alpha\right].$$

The infinite elements are not isoparametric, since the map uses a lower-order function of the parent coordinates than the interpolation scheme does. However, this singular mapping is convenient and invertible.

The phase factor $(r - r_0)$ appears in the element formulation. This factor models the oscillatory nature of the solution inside the infinite element volume but must also satisfy some additional properties. First, it must be zero at the face of the infinite element that connects to the finite element mesh. Second, it must be C^0 continuous across infinite element lateral boundaries. Third, it should be such that the mass-like term in the infinite element equation,

$$\int_{V_f} \frac{F}{K_f}\left(1 - \frac{\partial(r-r_0)}{\partial \mathbf{x}} \cdot \frac{\partial(r-r_0)}{\partial \mathbf{x}}\right)\phi_i\phi_j dV,$$

be positive semi-definite. This last criterion is not essential for execution of a steady-state analysis, but it is essential for the well-posedness of a transient analysis. Defining

$$q_\alpha \equiv \min_E \left(\mathbf{n}_\alpha \cdot \mathbf{n}_E\right),$$

where the subscript E refers to all elements at node α, the definition

$$r - r_0(g,h) \equiv \sum_{\alpha} r_\alpha q_\alpha b_\alpha(g,h)\left(\frac{\xi}{\xi-1}\right)$$

satisfies these requirements. The inclusion of the factor q_α is the only departure from the definition used in Astley (1994).

In a transient analysis the element matrix equation is transformed back to the time domain, using constant material properties taken from the value at zero frequency. The resulting second-order ordinary differential equation for the pressure degrees of freedom is added to the overall system in the model and integrated in the usual manner.

Steady-state analysis

In steady-state analysis the formulation for the acoustic infinite elements is consistent with that used for acoustic finite elements (see "Coupled acoustic-structural medium analysis," Section 2.9.1, and "Acoustic, shock, and coupled acoustic-structural analysis," Section 6.10.1 of the Abaqus Analysis User's Guide). First, the complex density

$$\tilde{\rho} \equiv \rho_f + \frac{\gamma}{i\Omega}$$

is defined. The resulting weighted residual statement is

$$-\int_{V_f} \delta p \frac{\Omega^2}{\mathrm{K}_f}\,\tilde{p}\,dV + \int_{V_f} \frac{1}{\tilde{\rho}}\frac{\partial \delta p}{\partial \mathbf{x}}\cdot\frac{\partial \tilde{p}}{\partial \mathbf{x}}\,dV + \int_S \delta p\,\frac{1}{\tilde{\rho}}\frac{\partial \tilde{p}}{\partial \mathbf{x}}\cdot\mathbf{n}^-\,dS = 0$$

after integration by parts.

The expression for the basis and weight functions is formally identical to the preceding transient development, but with the complex wave number

$$\tilde{k} \equiv \Omega\sqrt{\frac{\tilde{\rho}}{K_f}}$$

used in the definition of the spatially oscillatory term in the weight and basis functions:

$$e^{\pm\tilde{k}(r-r_0)}.$$

After substitution and some simplification, the acoustic infinite element expression becomes

$$\begin{aligned}
&\int_{V_f}\left(\phi_i\frac{\partial\phi_j}{\partial\mathbf{x}}\cdot\left(\frac{1}{\tilde{\rho}}\frac{\partial F}{\partial\mathbf{x}}\right)+F\,\frac{\partial\phi_i}{\partial\mathbf{x}}\cdot\left(\frac{1}{\tilde{\rho}}\frac{\partial\phi_j}{\partial\mathbf{x}}\right)\right)dV p_j\\
&\qquad -i\int_{V_f}\tilde{k}\left(\frac{\partial(r-r_0)}{\partial\mathbf{x}}\cdot\left(\frac{1}{\tilde{\rho}}\frac{\partial F}{\partial\mathbf{x}}\right)\phi_i\phi_j\right)dV p_j\\
-i\int_{V_f}\tilde{k}\frac{F}{\tilde{\rho}}&\left(\frac{\partial(r-r_0)}{\partial\mathbf{x}}\cdot\frac{\partial\phi_i}{\partial\mathbf{x}}\phi_j-\phi_i\frac{\partial\phi_j}{\partial\mathbf{x}}\cdot\frac{\partial(r-r_0)}{\partial\mathbf{x}}\right)dV p_j\\
&\qquad -\int_{V_f}\frac{\tilde{k}^2}{\tilde{\rho}}\left(1-\frac{\partial(r-r_0)}{\partial\mathbf{x}}\cdot\frac{\partial(r-r_0)}{\partial\mathbf{x}}\right)F\phi_i\phi_j dV p_j = 0.
\end{aligned} \tag{3.3.2–3}$$

Now

$$\frac{\tilde{k}^2}{\tilde{\rho}} = \frac{\Omega^2}{K_f},$$

so the mass matrix M_{f}^{ij} is the same as for the transient case. It is real-valued.

The remaining terms are complex; and, as in the case for acoustic finite elements, they are manipulated into real and imaginary parts. We use

$$\frac{1}{\tilde{\rho}} = \frac{\rho_f}{\rho_f^2+\gamma^2/\Omega^2} + i\frac{\gamma/\Omega}{\rho_f^2+\gamma^2/\Omega^2} \equiv \mathrm{R} + i\mathrm{I}$$

to obtain

$$\frac{\tilde{k}}{\tilde{\rho}} = \left[\frac{\Omega}{\sqrt{K_f}}\right]\left[\frac{1}{\rho_f^2+\gamma^2/\Omega^2}\right]^{(1/4)}\left[\cos\theta/2 + i\sin\theta/2\right] \equiv \left[\frac{\Omega}{\sqrt{K_f}}\right]\left[\mathrm{R}_2 + i\mathrm{I}_2\right],$$

where

$$\tan\theta = \left(\frac{\mathrm{I}}{\mathrm{R}}\right).$$

Using also

$$i\frac{\partial p}{\partial \mathbf{x}} = \frac{1}{\Omega}\frac{\partial \dot{p}}{\partial \mathbf{x}},$$

we obtain real-valued element matrices as follows:

$$\begin{aligned}
C_{\mathrm{f}}^{ij} &= \int_{V_f} \frac{\mathrm{I}}{\Omega}\frac{\partial\left[F\phi_i\right]}{\partial \mathbf{x}}\cdot\frac{\partial\phi_j}{\partial \mathbf{x}}\,dV, \\
&\quad + \int_{V_f} \frac{\mathrm{R}_2}{\sqrt{K}_f}\left[-\frac{\partial F\phi_i}{\partial \mathbf{x}}\cdot\frac{\partial(r-r_0)}{\partial \mathbf{x}}\phi_j + \frac{\partial\phi_j}{\partial \mathbf{x}}\cdot\frac{\partial(r-r_0)}{\partial \mathbf{x}}F\phi_i\right]\,dV, \\
K_{\mathrm{f}}^{ij} &= \int_{V_f} \mathrm{R}\frac{\partial[F\phi_i]}{\partial \mathbf{x}}\cdot\frac{\partial\phi_j}{\partial \mathbf{x}}\,dV, \\
&\quad - \int_{V_f} \frac{\Omega\mathrm{I}_2}{\sqrt{K}_f}\left[-\frac{\partial F\phi_i}{\partial \mathbf{x}}\cdot\frac{\partial(r-r_0)}{\partial \mathbf{x}}\phi_j + \frac{\partial\phi_j}{\partial \mathbf{x}}\cdot\frac{\partial(r-r_0)}{\partial \mathbf{x}}F\phi_i\right]\,dV.
\end{aligned}$$

These are used directly in Equation 2.9.1–24.

Lateral face terms due to impedance conditions

An infinite element volume may have an impedance boundary condition defined on a face extending to infinity. Such a definition can be used to model, for example, an infinite half-space of water above a lossy seabed or under a layer of ice. The derivation of the contribution of the impedance condition to the infinite element matrix is straightforward and follows the derivation for finite elements (see "Coupled acoustic-structural medium analysis," Section 2.9.1). The contributing term in the transient case is

$$\int_{S_{\mathrm{lateral}}} \delta p\left(\frac{\gamma}{\rho_f c_1}\,p + \left(\frac{\gamma}{\rho_f k_1} + \frac{1}{c_1}\right)\dot{p} + \frac{1}{k_1}\,\ddot{p}\right)\,dS.$$

Applying the infinite element basis and weight functions, the lateral face contributes to the mass, damping, and stiffness matrices of the acoustic infinite element:

$$\begin{aligned}
&\int_{S_{\mathrm{lateral}}} F\phi_i\phi_j\left(\frac{\gamma}{\rho_f c_1}\right)\,dS\,p_j + \int_{S_{\mathrm{lateral}}} F\phi_i\phi_j\left(\frac{\gamma}{\rho_f k_1} + \frac{1}{c_1}\right)\,dS\,\dot{p}_j \\
&\qquad + \int_{S_{\mathrm{lateral}}} F\phi_i\phi_j\left(\frac{1}{k_1}\right)\,dS\,\ddot{p}_j.
\end{aligned}$$

In steady-state and eigenvalue analysis the corresponding frequency-domain representation is used.

Coupling between acoustic and solid infinite elements

Lateral faces of acoustic and solid infinite elements (that is, those that extend to infinity) can be coupled to each other. For example, two infinite half-spaces may interact at their interface. As in the case of lateral-face impedance terms, the derivation of the contribution of the coupling condition to the infinite element matrix follows the derivation for finite elements to the greatest degree possible. We restrict the consideration to solid medium infinite elements that share an edge with the acoustic medium infinite elements and where the acoustic infinite element is the slave in a tied contact condition. The contributing terms in the transient case are

$$-\int_{S_{\text{lateral}}} \delta p \left(\mathbf{n}^- \cdot \ddot{\mathbf{u}}^m\right)\, dS$$

on the acoustic infinite element and

$$+\int_{S_{\text{lateral}}} \left(\mathbf{n}^- \cdot \delta\mathbf{u}^m\right) p\, dS$$

on the solid infinite element.

In tied contact between fluid and solid medium elements, a tributary area of the slave node is defined, and the conservation of momentum and continuity of displacement is enforced between the slave node and the master surface. On a lateral face of an infinite element, the "area" is unbounded, so an area measure based on the acoustic infinite element formulation,

$$A_{\text{lateral}} \equiv \int_{S_{\text{lateral}}} F f^2\, dS\,,$$

is defined. This area measure is used to compute a volumetric acceleration load on the acoustic infinite element:

$$-\int_{S_{\text{lateral}}} \delta p \left(\mathbf{n}^- \cdot \ddot{\mathbf{u}}^m\right)\, dS \mapsto A_{\text{lateral}} \left(\mathbf{n}^- \cdot \ddot{\mathbf{u}}^m\right).$$

The structural displacement, $\ddot{\mathbf{u}}^m$, is evaluated at the shared edge to compute this coupling condition. For consistency the boundary traction on the solid medium infinite element due to the acoustic pressure is defined as

$$+\int_{S_{\text{lateral}}} \left(\mathbf{n}^- \cdot \delta\mathbf{u}^m\right) p\, dS \mapsto A_{\text{lateral}} \left(\mathbf{n}^- \cdot \mathbf{u}^m\right).$$

In steady-state and eigenvalue analysis the corresponding frequency-domain representation is used.

References

- "Acoustic, shock, and coupled acoustic-structural analysis," Section 6.10.1 of the Abaqus Analysis User's Guide
- "Infinite elements," Section 28.3.1 of the Abaqus Analysis User's Guide

3.4 Membrane and truss elements

- "Membrane elements," Section 3.4.1
- "Truss elements," Section 3.4.2
- "Axisymmetric membranes," Section 3.4.3

3.4.1 MEMBRANE ELEMENTS

Products: Abaqus/Standard Abaqus/Explicit

Membrane elements are sheets in space that can carry membrane force but do not have any bending or transverse shear stiffness, so the only nonzero stress components in the membrane are those components parallel to the middle surface of the membrane: the membrane is in a state of plane stress.

At any time we use a local orthonormal basis system $\mathbf{e}_i$, where $\mathbf{e}_1$ and $\mathbf{e}_2$ are in the surface of the membrane and $\mathbf{e}_3$ is normal to the membrane. The basis system is defined by the standard convention used in Abaqus for a basis on a surface in space. In this section Greek indices take the range 1, 2, and Latin indices take the range 1, 2, 3. Greek indices are used to refer to components in the first two directions of the local orthonormal basis (in the surface of the membrane).

Equilibrium

The virtual work contribution from the internal forces in a membrane element is

$$\delta W^I = \int_V \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon}\, dV, \tag{3.4.1–1}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress, $\delta\boldsymbol{\varepsilon} = \mathrm{sym}(\delta\mathbf{L})$ is the virtual rate of deformation ($\delta\mathbf{L} = \partial\delta\mathbf{u}/\partial\mathbf{x}$, where $\delta\mathbf{u}$ is the virtual velocity field), and V is the current volume of the membrane.

We assume that only the membrane stress components in the surface of the membrane are nonzero: $\sigma_{3i} = 0$. Then Equation 3.4.1–1 simplifies to

$$\begin{aligned}\delta W^I &= \int_V \sigma_{\alpha\beta}\delta\varepsilon_{\alpha\beta}\, dV \\ &= \int_V \sigma_{\alpha\beta}\delta L_{\alpha\beta}\, dV \quad \text{(since } \boldsymbol{\sigma} \text{ is symmetric)},\end{aligned}$$

where

$$\delta L_{\alpha\beta} = \mathbf{e}_\alpha \frac{\partial\delta\mathbf{u}}{\partial\mathbf{x}} \cdot \mathbf{e}_\beta = \mathbf{e}_\alpha \frac{\partial\delta\mathbf{u}}{\partial x_\beta}$$

and $dV = t\, dA$, where t is the current thickness of the element and A is its current area.

Jacobian

The consistent Jacobian contribution from the element is

$$d\delta W^I = \int_A t\left(\delta\boldsymbol{\varepsilon} : \mathbf{D} : d\boldsymbol{\varepsilon} + \boldsymbol{\sigma} : (\delta\mathbf{L}^T \cdot d\mathbf{L} - 2\delta\boldsymbol{\varepsilon} \cdot d\boldsymbol{\varepsilon})\right) dA.$$

Since we assume that $\sigma_{3i} = 0$, the first term in the integrand is

$$\delta\varepsilon_{\alpha\beta} D_{\alpha\beta\gamma\delta} d\varepsilon_{\gamma\delta}.$$

We also assume that there is no transverse shear strain of the element: $\varepsilon_{3i} = 0$, and, hence, $\delta\varepsilon_{\alpha i} d\varepsilon_{i\beta} = \delta\varepsilon_{\alpha\gamma} d\varepsilon_{\gamma\beta}$. Thus, the second term in the integrand is

$$\sigma_{\alpha\beta}\left(\frac{\partial \mathbf{u}}{\partial x_\alpha}\cdot\frac{\partial \mathbf{u}}{\partial x_\beta} - \frac{1}{2}\left(\mathbf{e}_\gamma\cdot\frac{\partial \mathbf{u}}{\partial x_\alpha} + \mathbf{e}_\alpha\cdot\frac{\partial \mathbf{u}}{\partial x_\gamma}\right)\left(\mathbf{e}_\gamma\cdot\frac{\partial \mathbf{u}}{\partial x_\beta} + \mathbf{e}_\beta\cdot\frac{\partial \mathbf{u}}{\partial x_\gamma}\right)\right).$$

We can write this out as

$$\begin{aligned}
&\sigma_{11}\frac{\partial\delta\mathbf{u}}{\partial x_1}\cdot\frac{\partial d\mathbf{u}}{\partial x_1} + \sigma_{22}\frac{\partial\delta\mathbf{u}}{\partial x_2}\cdot\frac{\partial d\mathbf{u}}{\partial x_2} + \sigma_{12}\left(\frac{\partial\delta\mathbf{u}}{\partial x_1}\cdot\frac{\partial d\mathbf{u}}{\partial x_2} + \frac{\partial\delta\mathbf{u}}{\partial x_2}\cdot\frac{\partial d\mathbf{u}}{\partial x_1}\right)\\
&- 2\sigma_{11}\mathbf{e}_1\cdot\frac{\partial\delta\mathbf{u}}{\partial x_1}\mathbf{e}_1\cdot\frac{\partial d\mathbf{u}}{\partial x_1} - 2\sigma_{22}\mathbf{e}_2\cdot\frac{\partial\delta\mathbf{u}}{\partial x_2}\mathbf{e}_2\cdot\frac{\partial d\mathbf{u}}{\partial x_2}\\
&- \frac{1}{2}(\sigma_{11}+\sigma_{22})\left(\mathbf{e}_2\cdot\frac{\partial\delta\mathbf{u}}{\partial x_1}\mathbf{e}_2\cdot\frac{\partial d\mathbf{u}}{\partial x_1} + \mathbf{e}_1\cdot\frac{\partial\delta\mathbf{u}}{\partial x_2}\mathbf{e}_1\cdot\frac{\partial d\mathbf{u}}{\partial x_2} + \mathbf{e}_1\cdot\frac{\partial\delta\mathbf{u}}{\partial x_2}\mathbf{e}_2\cdot\frac{\partial d\mathbf{u}}{\partial x_1} + \mathbf{e}_2\cdot\frac{\partial\delta\mathbf{u}}{\partial x_1}\mathbf{e}_1\cdot\frac{\partial d\mathbf{u}}{\partial x_2}\right)\\
&- \sigma_{12}\left(\mathbf{e}_1\cdot\frac{\partial\delta\mathbf{u}}{\partial x_1}\mathbf{e}_1\cdot\frac{\partial d\mathbf{u}}{\partial x_1} + \mathbf{e}_2\cdot\frac{\partial\delta\mathbf{u}}{\partial x_1}\mathbf{e}_2\cdot\frac{\partial d\mathbf{u}}{\partial x_2} + \mathbf{e}_1\cdot\frac{\partial\delta\mathbf{u}}{\partial x_2}\mathbf{e}_1\cdot\frac{\partial d\mathbf{u}}{\partial x_1} + \mathbf{e}_2\cdot\frac{\partial\delta\mathbf{u}}{\partial x_2}\mathbf{e}_2\cdot\frac{\partial d\mathbf{u}}{\partial x_1}\right.\\
&\qquad\left. + \mathbf{e}_1\cdot\frac{\partial\delta\mathbf{u}}{\partial x_2}\mathbf{e}_2\cdot\frac{\partial d\mathbf{u}}{\partial x_2} + \mathbf{e}_2\cdot\frac{\partial\delta\mathbf{u}}{\partial x_1}\mathbf{e}_1\cdot\frac{\partial d\mathbf{u}}{\partial x_1} + \mathbf{e}_1\cdot\frac{\partial\delta\mathbf{u}}{\partial x_1}\mathbf{e}_2\cdot\frac{\partial d\mathbf{u}}{\partial x_1} + \mathbf{e}_2\cdot\frac{\partial\delta\mathbf{u}}{\partial x_2}\mathbf{e}_1\cdot\frac{\partial d\mathbf{u}}{\partial x_2}\right)
\end{aligned}$$

Thickness change

In geometrically nonlinear analyses the cross-section thickness changes as a function of the membrane strain with a user-defined "effective section Poisson's ratio," ν.

In plane stress $\sigma_{33} = 0$; linear elasticity gives

$$\epsilon_{33} = -\frac{\nu}{1-\nu}(\epsilon_{11}+\epsilon_{22}).$$

Treating these as logarithmic strains,

$$\ln\left(\frac{t}{t_0}\right) = -\frac{\nu}{1-\nu}\left(\ln\left(\frac{l_1}{l_1^0}\right) + \ln\left(\frac{l_2}{l_2^0}\right)\right) = -\frac{\nu}{1-\nu}\ln\left(\frac{A}{A^0}\right),$$

where $\boldsymbol{A}$ is the area on the membrane's reference surface. This nonlinear analogy with linear elasticity leads to the thickness change relationship:

$$\frac{t}{t_0} = \left(\frac{A}{A_0}\right)^{-\frac{\nu}{1-\nu}}.$$

For $\nu = 0.5$ the material is incompressible; for $\nu = 0$ the section thickness does not change.

Total deformation

The deformation gradient is $\mathbf{F} = \partial\mathbf{x}/\partial\mathbf{X}$. Since we take $\mathbf{e}_3$ normal to the current membrane surface and assume no transverse shear of the membrane,

$$F_{3\alpha} = \mathbf{e}_3 \cdot \frac{\partial \mathbf{x}}{\partial X_\alpha} = 0 \quad \text{and} \quad F_{\alpha 3} = \mathbf{e}_\alpha \cdot \frac{\partial \mathbf{x}}{\partial X_3} = 0.$$

By the thickness change assumption above, the direct out-of-plane component of the deformation gradient is

$$F_{33} = \frac{1}{(F_{11}F_{22} - F_{12}F_{21})^{\frac{\nu}{1-\nu}}} .$$

To calculate the deformation gradient at the end of the increment, first we calculate the two tangent vectors at the end of the increment defined by the derivative of the position with respect to the reference coordinates:

$$\frac{\partial \mathbf{x}^{n+1}}{\partial X_\beta} = \frac{\partial \mathbf{x}^{n+1}}{\partial \xi_\alpha}\frac{\partial \xi_\alpha}{\partial X_\beta} ,$$

where $\partial\mathbf{x}^{n+1}/\partial\xi_\alpha$ is obtained by interpolation with the shape function derivatives from the nodal coordinates and the change of coordinate transformation $\partial\xi_\alpha/\partial X_\beta$ is based on the reference geometry. The deformation gradient components are defined

$$F_{\alpha\beta} = \mathbf{e}_\alpha^{n+1} \cdot \frac{\partial \mathbf{x}^{n+1}}{\partial X_\beta} .$$

To choose the element basis directions $\mathbf{e}_\alpha^{n+1}$, we do the following. Find any pair of in-plane orthonormal vectors $\hat{\mathbf{e}}_\alpha^{n+1}$ (by the standard Abaqus projection). Then find the angle $\Delta\psi$ such that the element basis vectors $\mathbf{e}_\alpha^{n+1}$, defined

$$\begin{aligned} \mathbf{e}_1^{n+1} &= \cos(\Delta\psi)\hat{\mathbf{e}}_1^{n+1} + \sin(\Delta\psi)\hat{\mathbf{e}}_2^{n+1} \\ \mathbf{e}_2^{n+1} &= -\sin(\Delta\psi)\hat{\mathbf{e}}_1^{n+1} + \cos(\Delta\psi)\hat{\mathbf{e}}_2^{n+1} , \end{aligned}$$

satisfy the symmetry condition

$$F_{\alpha\beta} = \mathbf{e}_\alpha^{n+1} \cdot \frac{\partial \mathbf{x}^{n+1}}{\partial X_\beta} = F_{\beta\alpha} .$$

Using the definitions of $\mathbf{e}_\alpha^{n+1}$ in terms of $\hat{\mathbf{e}}_\alpha^{n+1}$ in the above equation, the rotation angle $\Delta\psi$ is found to be

$$\Delta\psi = \tan^{-1}\left[\frac{\hat{F}_{21} - \hat{F}_{12}}{\hat{F}_{11} + \hat{F}_{22}}\right] ,$$

where

$$\hat{F}_{\alpha\beta} \stackrel{\text{def}}{=} \hat{\mathbf{e}}_{\alpha}^{n+1} \cdot \frac{\partial \mathbf{x}^{n+1}}{\partial X_{\beta}} .$$

The deformation gradient then follows immediately.

For elastomers we work directly in terms of $B_{\alpha\beta} = F_{\alpha\gamma}F_{\beta\gamma}$ and $B_{33} = (F_{33})^2$. For inelastic material models we need measures of incremental strain and average material rotation, which we compute from $\Delta\mathbf{F}$ defined by $\mathbf{F} = \Delta\mathbf{F} \cdot \mathbf{F}^n$, where $\mathbf{F}^n$ is the deformation gradient at the start of the current increment (at increment "n"):

$$(F^n)_{\alpha\beta} = (\mathbf{e}^n)_{\alpha} \cdot \frac{\partial \mathbf{x}^n}{\partial X_{\beta}}.$$

We can define the components of $\Delta\mathbf{F}^{-1}$ by

$$(\Delta F^{-1})_{\alpha\beta} = (\mathbf{e}^n)_{\alpha} \cdot \frac{\partial \mathbf{x}^n}{\partial x_{\beta}}$$

and, hence, define $\Delta F_{\alpha\beta}$ by inversion.

The incremental strain and rotation are then defined from the polar decomposition $\Delta\mathbf{F} = \Delta\mathbf{V} \cdot \Delta\mathbf{R}$, where $\Delta\mathbf{R}$ is a rotation matrix and $\Delta\mathbf{V}$ is a pure stretch:

$$\Delta\mathbf{V} = \sum_{I=1}^{3} \Delta\lambda_I \mathbf{a}^I \, \mathbf{a}^I$$

(see "Deformation," Section 1.4.1). We find the $\Delta\lambda_I$ and the corresponding eigenvectors $\mathbf{a}_I$ by solving the eigenproblem for

$$\Delta\mathbf{F} \cdot \Delta\mathbf{F}^T = \Delta\mathbf{V} \cdot \Delta\mathbf{V} = \sum_{I=1}^{3} (\Delta\lambda_I)^2 \mathbf{a}^I \, \mathbf{a}^I.$$

Since we assume no transverse shear in the membrane, the normal direction (along $\mathbf{e}_3$) is always a principal direction, so the eigenproblem is the 2×2 problem

$$\Delta F_{\alpha\gamma} F_{\beta\gamma} = \sum_{I=1}^{2} (\Delta\lambda)^2 a_{\alpha}^I a_{\beta}^I.$$

The logarithmic strain increment is then

$$\Delta\varepsilon_{\alpha\beta} = \sum_{I=1}^{2} \ln(\Delta\lambda_I) a_{\alpha}^I a_{\beta}^I,$$

and the average material rotation increment is defined from the polar decomposition of the increment:

$$\Delta R_{\alpha\beta} = \sum_{I=1}^{2} \frac{1}{\Delta\lambda_I} a_\alpha^I a_\gamma^I \Delta F_{\gamma\beta}.$$

Due to the choice of the element basis directions, we can assume that

$$\Delta R_{\alpha\beta} \approx \delta_{\alpha\beta}\,.$$

Reference

- “Membrane elements,” Section 29.1.1 of the Abaqus Analysis User’s Guide

3.4.2 TRUSS ELEMENTS

Products: Abaqus/Standard Abaqus/Explicit

Truss elements are one-dimensional bars or rods that are assumed to deform by axial stretching only. They are pin jointed at their nodes, and so only translational displacements and the initial position vector at each node are used in the discretization. When the strains are large, the formulation is simplified by assuming that the trusses are made of incompressible material.

There are two truss elements in Abaqus: a 2-node linear interpolation truss and a 3-node quadratic interpolation truss. The quadratic interpolation version is in the library mainly for compatibility with the quadratic interpolation elements of other types, such as shell element S8R5. The same interpolation functions are used for both the Cartesian displacement components and for the Cartesian components of the initial position vector, so these elements are the simplest form of isoparametric elements.

The elements are one-dimensional: a single material (isoparametric) coordinate, g, is defined along the element, with $-1 \le g \le 1$ in the element. In a 2-node element node 1 is at $g = -1$ and node 2 is at $g = +1$. In the 3-node version node 1 is at $g = -1$, node 2 is at $g = 0$, and node 3 is at $g = +1$.

Interpolation

The interpolation for the 2-node element is

$$u(g) = \frac{1}{2}(1-g)u_1 + \frac{1}{2}(1+g)u_2,$$

and for the 3-node element,

$$u(g) = \frac{1}{2}g(g-1)u_1 + (1-g^2)u_2 + \frac{1}{2}g(g+1)u_3,$$

where u_1, u_2, and u_3 are the values of a variable at the nodes and $u(g)$ is the interpolated value of this variable.

Strain measure

These are one-dimensional elements, and the only strain considered is that along the axis of the element. The stretch ratio along the axis is

$$\lambda = \frac{dl}{dL},$$

where l measures length along the truss axis in the current configuration:

$$dl = \sqrt{\frac{d\mathbf{x}}{dg} \cdot \frac{d\mathbf{x}}{dg}}\, dg,$$

and dL measures length along the axis in the original configuration.

For geometrically nonlinear analysis we use a logarithmic strain measure:

$$\varepsilon = \ln\left(\frac{dl}{dL}\right).$$

First variation of strain

The first variation of strain is

$$\delta\varepsilon = \frac{dg}{dl}\,\mathbf{t}\cdot\frac{d\,\delta\mathbf{x}}{dg},$$

where

$$\mathbf{t} = \frac{dg}{dl}\frac{d\mathbf{x}}{dg}$$

is a unit tangent along the truss axis.

Second variation of strain

The second variation of strain is

$$d\delta\varepsilon = \left(\frac{dg}{dl}\right)^2 \frac{d\,\delta\mathbf{x}}{dg}\cdot[\mathbf{I}-2\mathbf{t}\,\mathbf{t}]\cdot\frac{d\,d\mathbf{x}}{dg}.$$

Integration

Stiffness

The linear truss is a constant strain element and so is integrated exactly. The quadratic truss is integrated numerically using two Gauss points.

Mass and consistent loads

A linear truss has two Gauss points. A quadratic truss has three Gauss points.

Virtual work contribution

The virtual work contribution from the stress in a truss element is

$$\delta W = \int_l a\sigma\delta\varepsilon\,dl,$$

where a is the current cross-sectional area of the truss, σ is the "true" (Cauchy) stress along the truss, ε is the logarithmic strain, and l is the length of the element.

Since we assume the truss is incompressible, $a\,dl = A\,dL$, where A is the original area and L the original length of the truss. So,

$$\delta W = \int_L A\sigma\delta\varepsilon\,dL.$$

This is the form in which the internal virtual work contribution is used for truss elements.

Mixed (hybrid) forms

"Hybrid" truss elements are also available in Abaqus/Standard. In those elements the axial force at the integration points is taken as an additional variable, with the compatibility condition introduced to define these variables. The formulation is identical to that used for the hybrid beam elements ("Hybrid beam elements," Section 3.5.4), without the bending terms.

Reference

- "Truss elements," Section 29.2.1 of the Abaqus Analysis User's Guide

3.4.3 AXISYMMETRIC MEMBRANES

Product: Abaqus/Standard

Abaqus includes two libraries of axisymmetric membrane elements, MAX and MGAX, whose geometry is axisymmetric (bodies of revolution) and that can be subjected to axially symmetric loading conditions. In addition, MGAX elements support torsion loading and general material anisotropy. Therefore, MGAX elements will be referred to as generalized axisymmetric membrane elements, and MAX elements will be referred to as regular axisymmetric membrane elements. In both cases the body of revolution is generated by revolving a line that represents the membrane surface (a membrane has negligible thickness) about an axis (the symmetry axis) and is described readily in cylindrical coordinates r, z, and θ. The radial and axial coordinates of a point on this cross-section are denoted by r and z, respectively. At $\theta = 0$, the radial and axial coordinates coincide with the global Cartesian X- and Y-coordinates.

If the loading consists of radial and axial components that are independent of θ and the material is either isotropic or orthotropic, with θ being a principal material direction, the displacement at any point will have only radial (u_r) and axial (u_z) components. The only nonzero stress components are σ_{ss} and $\sigma_{\theta\theta}$, where s denotes a length measuring coordinate along the line representing the membrane surface on any r–z plane. The deformation of any r–z plane (or, more precisely, any r–z line) completely defines the state of stress and strain in the body. Consequently, the geometric model is described by discretizing the reference cross-section at $\theta = 0$.

If one allows for a circumferential component of loading (which is independent of θ) and general material anisotropy, displacements and stress fields become three-dimensional. However, the problem remains axisymmetric in the sense that the solution does not vary as a function of θ, and the deformation of the reference r–z cross-section characterizes the deformation in the entire body. The motion at any point will have—in addition to the aforementioned radial and axial displacements—a twist ϕ (in radians) about the z-axis, which is independent of θ. There will also be a nonzero in-plane shear stress, $\sigma_{s\theta}$, as a result of the deformation.

This section describes the formulation of the generalized axisymmetric membrane elements. The formulation of the regular axisymmetric membrane elements is a subset of this formulation.

Kinematic description

The coordinate system used with both families of elements is the cylindrical system (r, z, θ), where r measures the distance of a point from the axis of the cylindrical system, z measures its position along this axis, and θ measures the angle between the plane containing the point and the axis of the coordinate system and some fixed reference plane that contains the coordinate system axis. The order in which the coordinates and displacements are taken in these elements is based on the convention that z is the second coordinate. This order is not the same as that used in three-dimensional elements in Abaqus, in which z is the third coordinate; nor is it the order (r, θ, z) that is usually taken in cylindrical systems.

Let $\mathbf{e}_R$, $\mathbf{e}_Z$, and $\mathbf{e}_\Theta$ be unit vectors in the radial, axial, and circumferential directions at a point in the undeformed state, as shown in Figure 3.4.3–1. The reference position $\mathbf{X}$ of the point can be represented in terms of the original radius, R, and the axial position, Z:

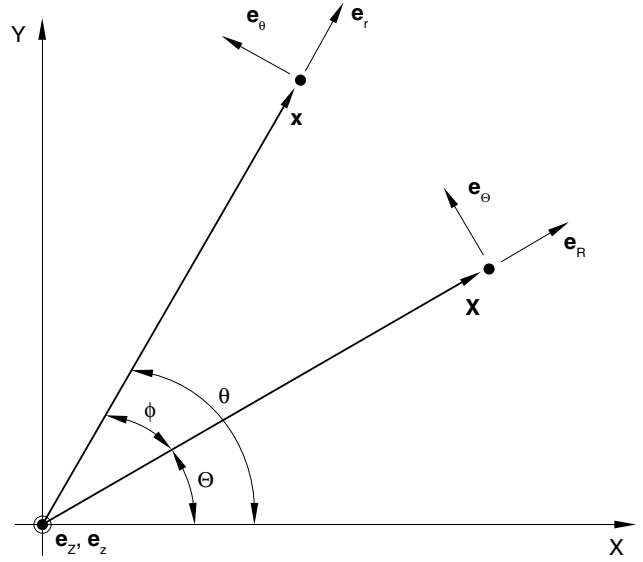


Figure 3.4.3–1 Cylindrical coordinate system and definition of position vectors.

$$\mathbf{X} = R\mathbf{e}_R + Z\mathbf{e}_Z,$$

Likewise, let $\mathbf{e}_r$, $\mathbf{e}_z$, and $\mathbf{e}_\theta$ be unit vectors in the radial, axial, and circumferential directions at a point in the deformed state. As shown in Figure 3.4.3–1, the radial and circumferential base vectors depend on the θ coordinate: $\mathbf{e}_r = \mathbf{e}_r(\theta)$ and $\mathbf{e}_\theta = \mathbf{e}_\theta(\theta)$.

The current position, $\mathbf{x}$, of the point can be represented in terms of the current radius, r, and the current axial position, z, as

$$\mathbf{x} = r\mathbf{e}_r + z\mathbf{e}_z.$$

The general axisymmetric motion at a point on the membrane surface can be described by

$$\begin{aligned} r(R,Z) &= R + u_r(R,Z)\,, \\ z(R,Z) &= Z + u_z(R,Z)\,, \\ \theta(R,Z,\Theta) &= \Theta + \phi(R,Z)\,. \end{aligned} \tag{3.4.3–1}$$

As this description implies, the degrees of freedom u_r, u_z, and ϕ are independent of Θ. Moreover, the reference cross-section of interest is at $\Theta = 0$; however, for the benefit of the mathematical analysis to follow, it is important that Θ be considered an independent variable in the above expression for θ.

Parametric interpolation and integration

The following isoparametric interpolation scheme is used for the motion:

$$u_r = N^N(g)\bar{u_r}^N,$$
$$u_z = N^N(g)\bar{u_z}^N,$$
$$\phi = N^N(g)\bar{\phi}^N,$$

where g is the isoparametric coordinate in the reference r–z cross-section at $\Theta = 0$; and $\bar{u_r}^N$, $\bar{u_z}^N$, $\bar{\phi}^N$ are the nodal degrees of freedom. The interpolation functions are identical to those used for truss elements (see "Truss elements," Section 3.4.2). All elements use reduced integration.

Deformation gradient

For a material point the deformation gradient $\mathbf{F}$ is defined as the gradient of the current position, $\mathbf{x}$, with respect to the original position, $\mathbf{X}$, and is given by

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}. \tag{3.4.3–2}$$

The components of the deformation gradient require that two sets of orthonormal basis vectors be defined. In the undeformed configuration the basis vectors are defined by

$$\mathbf{E}_i = \frac{\partial \mathbf{X}}{\partial S_i},$$

where the S_i denote length measuring coordinates in the reference configuration along the element length and the hoop direction, respectively. Thus,

$$\mathbf{E}_1 = \frac{\partial \mathbf{X}}{\partial S},\ \mathbf{E}_2 = \frac{\partial \mathbf{X}}{R\partial\Theta}.$$

In the current configuration Abaqus formulates the equations in terms of a fixed spatial basis with respect to the axisymmetric twist degree of freedom. The basis vectors convect with the material. However, because of the axisymmetry of the model in the deformed configuration, these vectors can be defined at $\Theta = 0$ as

$$\mathbf{e}_i = \frac{\partial \mathbf{x}}{\partial s_i},$$

where the s_i denote length measuring coordinates in the current configuration along the element length and the hoop direction, respectively. Thus, the basis vectors in the reference and current configurations can be written as

$$\mathbf{E}_1 = \frac{\partial \mathbf{X}}{\partial S},\ \mathbf{E}_2 = \frac{\partial \mathbf{X}}{R\partial\Theta},\ \mathbf{e}_1 = \frac{\partial \mathbf{x}}{\partial s},\ \mathbf{e}_2 = \frac{\partial \mathbf{x}}{r\partial\theta}, \tag{3.4.3–3}$$

where S and s are length measuring coordinates along the element length in the reference and current configurations, respectively. The components of the deformation gradient in the two sets of basis vectors may be computed as

$$F_{\alpha\beta} = \mathbf{e}_\alpha \cdot \mathbf{F} \cdot \mathbf{E}_\beta .$$

Using the definitions of the basis vectors in Equation 3.4.3–3, the components of the deformation gradient tensor are

$$\begin{aligned} F_{11} &= \frac{\partial r}{\partial s}\frac{\partial r}{\partial S} + \frac{\partial z}{\partial s}\frac{\partial z}{\partial S}\,, \\ F_{21} &= r\frac{\partial \phi}{\partial S}\,, \\ F_{12} &= 0\,, \\ F_{22} &= \frac{r}{R}\,. \end{aligned}$$

Virtual work

As discussed in "Equilibrium and virtual work," Section 1.5.1, the formulation of equilibrium (virtual work) requires the virtual velocity gradient, which takes the form

$$\delta\mathbf{L} = \delta\mathbf{F} \cdot \mathbf{F}^{-1}\,,$$

where $\delta\mathbf{F}$ represents the first variation of the deformation gradient tensor. Alternatively, the virtual velocity gradient can be written as

$$\delta\mathbf{L} = \frac{\partial \delta\mathbf{x}}{\partial \mathbf{x}}.$$

Recall that Abaqus formulates the finite element equations in terms of a fixed spatial basis with respect to the axisymmetric twist degree of freedom. Therefore, the desired result for $\delta\mathbf{F}$ does not simply follow from the linearization of Equation 3.4.3–2. Namely, the contributions from the variations

$$\delta\mathbf{e}_r = \delta\phi\ \mathbf{e}_\theta \quad \text{and} \quad \delta\mathbf{e}_\theta = -\delta\phi\ \mathbf{e}_r$$

arising from the spin of the coordinate system must be canceled out. To this end, $\mathbf{F}$ can be modified according to

$$\tilde{\mathbf{F}} = \mathbf{R} \cdot \mathbf{F}\,,$$

where $\mathbf{R} = \mathbf{I}$ instantaneously, but its variation is given by

$$\delta\mathbf{R} = \widehat{\delta\boldsymbol{\phi}} \cdot \mathbf{R}\,,$$

where $\widehat{\delta\boldsymbol{\phi}}$ is skew-symmetric with components

$$[\widehat{\delta\boldsymbol{\phi}}] = \begin{bmatrix} 0 & \delta\phi & 0 \\ -\delta\phi & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

with respect to the basis $\mathbf{e}_r$, $\mathbf{e}_z$, and $\mathbf{e}_\theta$ at $\theta = 0$.

With this modification, the corotational virtual deformation gradient is given by

$$\delta\tilde{\mathbf{F}} = \delta\mathbf{R} \cdot \mathbf{F} + \mathbf{R} \cdot \delta\mathbf{F} = \widehat{\delta\boldsymbol{\phi}} \cdot \mathbf{F} + \delta\mathbf{F}\,,$$

and the corotational virtual velocity gradient by

$$\delta\tilde{\mathbf{L}} = \delta\tilde{\mathbf{F}} \cdot \tilde{\mathbf{F}}^{-1} = \widehat{\delta\boldsymbol{\phi}} + \delta\mathbf{L}\,.$$

The individual components of $\delta\tilde{\mathbf{L}}$ are given by

$$\begin{aligned}
\delta\tilde{L}_{11} &= \frac{\partial r}{\partial s}\frac{\partial \delta r}{\partial s} + \frac{\partial z}{\partial s}\frac{\partial \delta z}{\partial s}\,,\\
\delta\tilde{L}_{21} &= r\frac{\partial \delta\phi}{\partial s}\,,\\
\delta\tilde{L}_{31} &= \frac{\partial r}{\partial s}\frac{\partial \delta z}{\partial s} - \frac{\partial z}{\partial s}\frac{\partial \delta r}{\partial s}\,,\\
\delta\tilde{L}_{12} &= 0\,,\\
\delta\tilde{L}_{22} &= \frac{\delta r}{r}\,,\\
\delta\tilde{L}_{32} &= 0\,.
\end{aligned}$$

The components $\delta\tilde{L}_{i3}$ are not determined by the kinematics.

Stiffness in the current state

The second variation has the usual contribution:

$$d\delta\tilde{D}_{ij} = -\mathrm{sym}(\delta\tilde{L}_{ik}\, d\tilde{L}_{kj}).$$

Moreover, there are additional contributions from $d\delta\tilde{L}_{ij}$, which are given by

$$
\begin{aligned}
d\delta\tilde{L}_{11} &= 0\,,\\
d\delta\tilde{L}_{21} &= \delta r\frac{\partial d\phi}{\partial s} + dr\frac{\partial \delta\phi}{\partial s}\,,\\
d\delta\tilde{L}_{12} &= 0\,,\\
d\delta\tilde{L}_{22} &= 0\,.
\end{aligned}
$$

The remaining terms do not contribute since $\sigma_{i3} = 0$.

Reference

- "Axisymmetric membrane element library," Section 29.1.4 of the Abaqus Analysis User's Guide

3.5 Beam elements

- “Beam element overview,” Section 3.5.1
- “Beam element formulation,” Section 3.5.2
- “Euler-Bernoulli beam elements,” Section 3.5.3
- “Hybrid beam elements,” Section 3.5.4
- “Mass and inertia for Timoshenko beams,” Section 3.5.5
- “Meshed beam cross-sections,” Section 3.5.6

3.5.1 BEAM ELEMENT OVERVIEW

The element library in Abaqus contains several types of beam elements. A "beam" in this context is an element in which assumptions are made so that the problem is reduced to one dimension mathematically: the primary solution variables are functions of position along the beam axis only. For such assumptions to be reasonable, it is intuitively clear that a beam must be a continuum in which we can define an axis such that the shortest distance from the axis to any point in the continuum is small compared to typical lengths along the axis. This idea is made more precise in the detailed derivations in "Beam element formulation," Section 3.5.2. There are several levels of complexity in the assumptions upon which the reduction to a one-dimensional problem can be made, and different beam elements in Abaqus use different assumptions.

The simplest approach to beam theory is the classical Euler-Bernoulli assumption, that plane cross-sections initially normal to the beam's axis remain plane, normal to the beam axis, and undistorted. The beam elements in Abaqus that use cubic interpolation (element types B23, B33, etc.) all use this assumption, implemented in the context of arbitrarily large rotations but small strains. The Euler-Bernoulli beam elements are described in "Euler-Bernoulli beam elements," Section 3.5.3. This approximation can also be used to formulate beams for large axial strains as well as large rotations. The beam elements in Abaqus that use linear and quadratic interpolation (B21, B22, B31, B32, PIPE21, PIPE22, PIPE31, PIPE32, etc.) are based on such a formulation, with the addition that these elements also allow "transverse shear strain"; that is, the cross-section may not necessarily remain normal to the beam axis. This extension leads to Timoshenko beam theory (Timoshenko, 1956) and is generally considered useful for thicker beams, whose shear flexibility may be important. (These elements in Abaqus are formulated so that they are efficient for thin beams—where Euler-Bernoulli theory is accurate—as well as for thick beams: because of this they are the most effective beam elements in Abaqus.) The large-strain formulation in these elements allows axial strains of arbitrary magnitude; but quadratic terms in the nominal torsional strain are neglected compared to unity, and the axial strain is assumed to be small in the calculation of the torsional shear strain. Thus, while the axial strain may be arbitrarily large, only "moderately large" torsional strain is modeled correctly, and then only when the axial strain is not large. We assume that, throughout the motion, the radius of curvature of the beam is large compared to distances in the cross-section: the beam cannot fold into a tight hinge. A further assumption is that the strain in the beam's cross-section is the same in any direction in the cross-section and throughout the section. Some additional assumptions are made in the derivation of these elements: these are introduced in the detailed derivation in "Beam element formulation," Section 3.5.2.

For certain important designs the beam is constructed from thin segments made up into an open section. The response of such open sections is strongly effected by warping, when material particles move out of the plane of the section along lines parallel to the beam axis so as to minimize the shearing between lines along the wall of the section and along the beam axis. The beam element formulation ("Beam element formulation," Section 3.5.2) includes provision for such effects. Beam elements that allow for warping of open sections (B31OS, B32OS etc.) are also derived. The particular approach used for modeling open-section warping in Abaqus is based on the assumption that the warping amplitude is never large anywhere along the beam axis because the warping will be constrained at some points along the beam—perhaps because one or both ends of the beam are built into a stiff structure or because some form of transverse stiffeners are added.

The regular beam elements can be used for slender and moderately thick beams. For extremely slender beams, for which the length to thickness ratio is $O(10^3)$ or more and geometrically nonlinear analysis is required (such as pipelines), convergence may become very poor. For such cases use of the hybrid elements, in which the axial (and transverse) forces are treated as independent degrees of freedom, can be beneficial. The hybrid beam formulation is described in "Hybrid beam elements," Section 3.5.4. Distributed pressure loads applied to beams (for example, due to wind or current) will rotate with the beam, leading to follower force effects. The derivation of the load stiffness that accounts for this effect is presented in "Load stiffness for beam elements," Section 6.5.2.

In some piping applications thin-walled, circular, relatively straight pipes are subjected to relatively large magnitudes of internal pressure. This has the effect of creating high levels of hoop stress around the wall of the pipe section so that, if the section yields plastically, the axial yield stress will be different in tension and compression because of the interaction with this hoop stress. The PIPE elements allow for this effect by providing uniform radial expansion of the cross-section caused by internal pressure.

In other piping cases thin-walled straight pipes might be subjected to large amounts of bending so that the section collapses ("Brazier collapse"); or a section of pipe may already be curved in its initial configuration—it might be an "elbow." In such cases the ovalization and, possibly, warping, of the cross-section may be important: these effects can reduce the bending stiffness of the member by a factor of five or more in common piping designs. For material linear analysis these effects can be incorporated by making suitable adjustments to the section's bending stiffness (by multiplying the bending stiffness calculated from beam theory by suitable flexibility factors); but when nonlinear material response is a part of the problem it is necessary to model this ovalization and warping explicitly. Elbow elements are provided for that purpose; they are described in "Elbow elements," Section 3.9.1. Elbow elements look like beam elements to the user, but they incorporate displacement variables that allow ovalization and warping and so are much more complex in their formulation. In particular, ovalization of the section implies a strong gradient of strain with respect to position through the wall of the pipe: this requires numerical integration through the pipe wall, on top of that used around the pipe section, to capture the material response. This makes the elbow elements computationally more expensive than beams.

Since consideration of planar deformation only provides considerable simplification in formulating beam elements, for each beam element type in Abaqus a corresponding beam element is provided that only moves in the (X, Y) plane. However, the open-section beams are provided only in three dimensions for reasons that are obvious.

Reference

- "Beam modeling: overview," Section 29.3.1 of the Abaqus Analysis User's Guide

3.5.2 BEAM ELEMENT FORMULATION

Products: Abaqus/Standard Abaqus/Explicit

At a given stage in the deformation history of the beam, the position of a material point in the cross-section is given by the expression

$$\hat{\mathbf{x}}(S, S^\alpha) = \mathbf{x}(S) + f(S)S^\alpha \mathbf{n}_\alpha(S) + w(S)\psi(S^\alpha)\mathbf{t}(S).$$

In this expression $\mathbf{x}(S)$ is the position of a point on the centerline, $\mathbf{n}_\alpha(S)$ are unit orthogonal direction vectors in the plane of the beam section, $\mathbf{t}(S)$ is the unit vector orthogonal to $\mathbf{n}_1$ and $\mathbf{n}_2$, $\psi(S^\alpha)$ is the warping function of the section, $w(S)$ is the warping amplitude, and $f(S)$ is a cross-sectional scaling factor depending on the stretch of the beam.

These quantities are functions of the beam axis coordinate S and the cross-sectional coordinates S^α, which are assumed to be distances measured in the original (reference) configuration of the beam. The warping function is chosen such that the value at the origin of the section vanishes: $\psi(0) = 0$.

It is assumed that at the integration points along the beam, the beam section directions are approximately orthogonal to the beam axis tangent $\mathbf{s}$ given by

$$\mathbf{s} = \lambda^{-1}\frac{d\mathbf{x}}{dS},$$

where λ is the axial stretch given by

$$\lambda = \left|\frac{d\mathbf{x}}{dS}\right|.$$

The normality condition is enforced numerically by penalizing the transverse shear strains

$$\gamma_\alpha = \mathbf{s}\cdot\mathbf{n}_\alpha.$$

This condition is assumed to be satisfied exactly in the original configuration.

In what follows, ϵ_α^β is the alternator

$$\epsilon_1^2 = -\epsilon_2^1 = 1\ , \epsilon_1^1 = \epsilon_2^2 = 0.$$

The curvature of the beam is defined by

$$b_\alpha = \epsilon_\alpha^\beta\ \mathbf{t}\cdot\frac{d\mathbf{n}_\beta}{dS},$$

and the twist of the beam follows from

$$b = \mathbf{n}_2\cdot\frac{d\mathbf{n_1}}{dS} = -\mathbf{n}_1\cdot\frac{d\mathbf{n_2}}{dS}.$$

The "bicurvature" of the beam is defined by

$$\chi = \frac{dw}{dS}.$$

The bicurvature defines the axial strain variation in the section due to the twist of the beam. The expression for the curvature and the twist can be combined to yield

$$\frac{d\mathbf{n}_\alpha}{dS} = \epsilon_\alpha^\beta(-b_\beta \mathbf{t} + b\mathbf{n}_\beta).$$

Before we derive strain measures from these expressions, we will consider in detail how the above quantities and their first and second variations are obtained for a typical beam finite element.

Beam element in undeformed configuration

In the undeformed configuration, we use capital letters for all quantities. We assume that the undeformed state has no warping, so the position of a material point is given by

$$\hat{\mathbf{X}}(S, S^\alpha) = \mathbf{X}(S) + S^\alpha \mathbf{N}_\alpha(S)$$

and the curvatures and twist are defined by

$$B_\alpha = \epsilon_\alpha^\beta \mathbf{T} \cdot \frac{d\mathbf{N}_\beta}{dS},$$
$$B = \mathbf{N}_2 \cdot \frac{d\mathbf{N}_1}{dS} = -\mathbf{N}_1 \cdot \frac{d\mathbf{N}_2}{dS},$$

where $\mathbf{T}$ is the unit vector orthogonal to $\mathbf{N}_1$ and $\mathbf{N}_2$; i.e.,

$$\mathbf{T} = \mathbf{N}_1 \times \mathbf{N}_2.$$

We assume that the section normal $\mathbf{T}$ coincides with the beam tangent $\mathbf{S}$.

In the element the position of a point on the axis is interpolated from nodal positions $\mathbf{X}^N$ with standard interpolation functions as

$$\mathbf{X} = g_N(\xi)\mathbf{X}^N,$$

where ξ is a parametric coordinate, typically varying between -1 and *1* along the element. The beam axis tangent is readily computed as

$$\mathbf{S} = \frac{d\mathbf{X}}{d\xi} \Big/ \frac{dS}{d\xi} = \frac{dg_N}{d\xi}\mathbf{X}^N \Big/ \left|\frac{dg_M}{d\xi}\mathbf{X}^M\right|.$$

The section normals are interpolated from user-defined nodal normals $\mathbf{N}^N$. However, we cannot use a simple interpolation since this would not create integration point normals orthogonal to the beam tangent $\mathbf{S}$. Hence, we use a two-step approach. First, we create approximate normals by the interpolation

$$\overline{\mathbf{N}}_\alpha = g_N(\xi)\mathbf{N}^N_\alpha.$$

Then, we orthonormalize these vectors with respect to $\mathbf{S}$ by

$$\tilde{\mathbf{N}}_\alpha = (\overline{\mathbf{N}}_\alpha - \mathbf{S}\,\mathbf{S}\cdot\overline{\mathbf{N}}_\alpha)\big/\left|(\overline{\mathbf{N}}_\alpha - \mathbf{S}\,\mathbf{S}\cdot\overline{\mathbf{N}}_\alpha)\right| \qquad (\text{no summation})$$

and subsequently, with respect to each other by

$$\mathbf{N}_\alpha = (\tilde{\mathbf{N}}_\alpha + \epsilon^\beta_\alpha\tilde{\mathbf{N}}_\beta\times\mathbf{S})/|(\tilde{\mathbf{N}}_\alpha + \epsilon^\beta_\alpha\tilde{\mathbf{N}}_\beta\times\mathbf{S})| \qquad (\text{no sum on } \alpha),$$

where it has been assumed that $\overline{\mathbf{N}}_\alpha$ and $\mathbf{S}$ form a right-handed system. This provides $\mathbf{T} = \mathbf{S}$.

The curvature and the twist in the initial configuration are calculated directly from $\overline{\mathbf{N}}_\alpha$ as

$$\begin{aligned} B_\alpha &= \epsilon^\alpha_\beta\mathbf{T}\cdot\frac{d\overline{\mathbf{N}}_\beta}{dS} = \epsilon^\beta_\alpha\mathbf{T}\cdot\frac{dg_N}{d\xi}\mathbf{N}^N_\alpha\Big/\frac{dS}{d\xi},\\ B &= \frac{1}{2}\epsilon^\beta_\alpha\overline{\mathbf{N}}_\beta\cdot\frac{d\overline{\mathbf{N}}_\alpha}{dS} = \frac{1}{2}\epsilon^\beta_\alpha\overline{\mathbf{N}}_\beta\cdot\frac{dg_N}{d\xi}\mathbf{N}^N\Big/\frac{dS}{d\xi}. \end{aligned}$$

The "average" twist is taken, since, in general,

$$\overline{\mathbf{N}}_2\cdot\frac{d\overline{\mathbf{N}}_1}{dS} \neq -\overline{\mathbf{N}}_1\cdot\frac{d\overline{\mathbf{N}}_2}{dS}.$$

The gradient of the normal vectors is then obtained as

$$\frac{d\mathbf{N}_\alpha}{dS} = \epsilon^\beta_\alpha(-B_\beta\mathbf{T} + B\mathbf{N}_\beta);$$

and, therefore, the curvature and twist are also equal to

$$\begin{aligned} B_\alpha &= \epsilon^\beta_\alpha\mathbf{T}\cdot\frac{d\mathbf{N}_\beta}{dS},\\ B &= \mathbf{N}_2\cdot\frac{d\mathbf{N}_1}{dS} = -\mathbf{N}_1\cdot\frac{d\mathbf{N}_2}{dS}. \end{aligned}$$

The procedure followed above to derive $\mathbf{N}_\alpha$ and $\dfrac{d\mathbf{N}_\alpha}{dS}$ is not unique but provides values that satisfy the proper orthonormality conditions. This procedure is followed only for the undeformed configuration. For subsequent configurations $\mathbf{n}_\alpha$ and $\dfrac{d\mathbf{n}_\alpha}{dS}$ are obtained individually by forward integration of the kinematic equations.

Change of position, warping, and normal direction

We assume that the position of the beam axis and the orientation of the normals can undergo (independent) changes. The change in position of the axis is described by the velocity vector $\mathbf{v} = \mathbf{v}(\xi)$, which can be obtained from nodal velocities $\mathbf{v}^N$ with the standard interpolation functions as

$$\mathbf{v}(\xi) = g_N(\xi)\mathbf{v}^N.$$

The change in orientation of the normals is described by the spin vector $\boldsymbol{\omega} = \boldsymbol{\omega}(\xi)$, which is obtained from nodal spin vectors $\boldsymbol{\omega}^N$ with the same interpolation functions $g_N(\xi)$, as

$$\dot{\mathbf{n}}_\alpha = \boldsymbol{\omega}(\xi) \times \mathbf{n}_\alpha, \;\; \boldsymbol{\omega}(\xi) = g_N(\xi)\boldsymbol{\omega}^N.$$

Rigid body motion is included since the original position $\mathbf{X}(\xi)$ is obtained by the same interpolation as $\mathbf{v}(\xi)$. The rate of change of warping is also defined in terms of the rate of change of nodal warping w^N with the standard interpolation functions as

$$\dot{w}(\xi) = g_N(\xi)\dot{w}^N.$$

The velocity and spin describe the *rate* of change of the position and orientation. *Finite* changes in position are obtained by integration of the velocities over a finite time increment as

$$\Delta\mathbf{x}(\xi) = \int_t^{t+\Delta t} \mathbf{v}(\xi)dt = \; g_N(\xi) \int_t^{t+\Delta t} \mathbf{v}^N dt = g_N(\xi)\Delta\mathbf{X}^N.$$

Similarly, for the warping,

$$\Delta w(\xi) = \int_t^{t+\Delta t} \dot{w}(\xi)dt = \; g_N(\xi) \int_t^{t+\Delta t} \dot{w}^N dt = g_N(\xi)\Delta w^N.$$

The spin is related to the rate of change of the rotation quaternion $\boldsymbol{q}$ by

$$2\boldsymbol{q}^\dagger\dot{\boldsymbol{q}} = \boldsymbol{\omega}, \quad \boldsymbol{q} = \left(\cos\frac{\theta}{2}\,, \sin\frac{\theta}{2}\mathbf{n}\right), \quad \boldsymbol{q}^\dagger = \left(\cos\frac{\theta}{2}\,, -\sin\frac{\theta}{2}\mathbf{n}\right),$$

where $\boldsymbol{\theta} = \theta\mathbf{n}$ is the total or Euler rotation.

The relation between spin and quaternion can be integrated exactly if it is assumed that the spin is constant over the time increment Δt. We then define

$$\Delta\boldsymbol{\phi} = \omega\Delta t, \;\; \Delta\phi = |\Delta\boldsymbol{\phi}|, \;\; \mathbf{n} = \Delta\boldsymbol{\phi}/\Delta\phi,$$

and the incremental rotation quaternion follows from

$$\Delta\boldsymbol{q} = \left(\cos\frac{\Delta\phi}{2}\,, \sin\frac{\Delta\phi}{2}\mathbf{n}\right).$$

This allows us to update the normal directions at the nodes and at stress points with

$$\mathbf{n}_\alpha = \Delta\boldsymbol{q}\mathbf{N}_\alpha\Delta\boldsymbol{q}^\dagger.$$

In this equation we use the notation that $\mathbf{N}_\alpha$ is $\mathbf{n}_\alpha$ at the beginning of the current increment; i.e.,

$$\mathbf{N}_\alpha \stackrel{\text{def}}{=} \mathbf{n}_\alpha(t).$$

For the entire motion the new normal directions can be formally expressed as

$$\mathbf{n}_\alpha = \boldsymbol{q}\mathbf{N}_\alpha\boldsymbol{q}^\dagger,$$

where $\boldsymbol{q}$ is defined by the product rule

$$\boldsymbol{q} = \prod_{i=1}^{n} \Delta\boldsymbol{q}^{(i)}.$$

Here $\boldsymbol{i}$ is an increment and $\Delta\boldsymbol{q}^{(i)}$ is the rotation quaternion for that increment. The section normal $\mathbf{t}$ is updated the same way.

Change of curvature and twist

The curvature and the twist involve the derivative of the normal vector with respect to $\boldsymbol{S}$. From the update rule for $\mathbf{n}_\alpha$,

$$\frac{d\mathbf{n}_\alpha}{dS} = \frac{d\Delta\boldsymbol{q}}{dS}\mathbf{N}_\alpha\Delta\boldsymbol{q}^\dagger + \Delta\boldsymbol{q}\,\mathbf{N}_\alpha\frac{d\Delta\boldsymbol{q}^\dagger}{dS} + \Delta\boldsymbol{q}\frac{d\mathbf{N}_\alpha}{dS}\Delta\boldsymbol{q}^\dagger.$$

The second term can be written as

$$\Delta\boldsymbol{q}\mathbf{N}_\alpha\frac{d\Delta\boldsymbol{q}^\dagger}{dS} = \left[\frac{d\Delta\boldsymbol{q}}{dS}\mathbf{N}_\alpha^\dagger\Delta\boldsymbol{q}^\dagger\right]^\dagger = -\left[\frac{d\Delta\boldsymbol{q}}{dS}\mathbf{N}_\alpha\Delta\boldsymbol{q}^\dagger\right]^\dagger.$$

Hence, the scalar parts of the first two terms cancel each other, and the vector parts are the same. Therefore,

$$\frac{d\mathbf{n}_\alpha}{dS} = 2\mathbf{V}\left(\frac{d\Delta\boldsymbol{q}}{dS}\mathbf{N}_\alpha\Delta\boldsymbol{q}^\dagger\right) + \Delta\boldsymbol{q}\frac{d\mathbf{N}_\alpha}{dS}\Delta\boldsymbol{q}^\dagger.$$

Because $\Delta\boldsymbol{q}$ is a rotation quaternion, its inverse is equal to its conjugate ($\Delta\boldsymbol{q}^{-1} = \Delta\boldsymbol{q}^\dagger$), and, hence, we can write

$$\frac{d\mathbf{n}_\alpha}{dS} = 2\mathbf{V}\left(\frac{d\Delta\boldsymbol{q}}{dS}\Delta\boldsymbol{q}^\dagger\Delta\boldsymbol{q}\,\mathbf{N}_\alpha\Delta\boldsymbol{q}^\dagger\right) + \Delta\boldsymbol{q}\frac{d\mathbf{N}_\alpha}{dS}\Delta\boldsymbol{q}^\dagger$$
$$= \mathbf{V}\left(2\frac{d\Delta\boldsymbol{q}}{dS}\Delta\boldsymbol{q}^\dagger\mathbf{n}_\alpha\right) + \Delta\boldsymbol{q}\frac{d\mathbf{N}_\alpha}{dS}\Delta\boldsymbol{q}^\dagger,$$

where $\mathbf{p} = \mathbf{V}(\boldsymbol{p})$ denotes the vector part of a quaternion. For the first term we use the relation

$$\frac{d\Delta\boldsymbol{q}}{dS} = \left(-\frac{1}{2}\sin\frac{\Delta\phi}{2}\frac{d\Delta\phi}{dS}\,,\ \frac{1}{2}\cos\frac{\Delta\phi}{2}\mathbf{n}\frac{d\Delta\phi}{dS} + \sin\frac{\Delta\phi}{2}\frac{d\mathbf{n}}{dS}\right).$$

This leads to

$$2\frac{d\Delta\boldsymbol{q}}{dS}\Delta\boldsymbol{q}^\dagger = \mathbf{n}\frac{d\Delta\phi}{dS} + \sin\Delta\phi\frac{d\mathbf{n}}{dS} + (1-\cos\Delta\phi)\mathbf{n}\times\frac{d\mathbf{n}}{dS} = \Delta\boldsymbol{\varrho},$$

which is a vector. From the definitions of $\Delta\phi$ and $\mathbf{n}$ it follows that

$$\frac{d\Delta\phi}{dS} = \Delta\phi^{-1}\Delta\boldsymbol{\phi}\cdot\frac{d\Delta\boldsymbol{\phi}}{dS} = \mathbf{n}\cdot\frac{d\Delta\boldsymbol{\phi}}{dS},$$
$$\frac{d\mathbf{n}}{dS} = \Delta\phi^{-1}\frac{d\Delta\boldsymbol{\phi}}{dS} - \Delta\phi^{-2}\Delta\boldsymbol{\phi}\,\mathbf{n}\cdot\frac{d\Delta\boldsymbol{\phi}}{dS} = \Delta\phi^{-1}\left[\frac{d\Delta\boldsymbol{\phi}}{dS} - \mathbf{n}\,\mathbf{n}\cdot\frac{d\Delta\boldsymbol{\phi}}{dS}\right].$$

These results provide

$$\Delta\boldsymbol{\varrho} = \mathbf{n}\,\mathbf{n}\cdot\frac{d\Delta\boldsymbol{\phi}}{dS} + \frac{\sin\Delta\phi}{\Delta\phi}\frac{d\Delta\boldsymbol{\phi}}{dS} - \frac{\sin\Delta\phi}{\Delta\phi}\mathbf{n}\,\mathbf{n}\cdot\frac{d\Delta\boldsymbol{\phi}}{dS} + \frac{(1-\cos\Delta\phi)}{\Delta\phi}\mathbf{n}\times\frac{d\Delta\boldsymbol{\phi}}{dS}$$
$$= \frac{\sin\Delta\phi}{\Delta\phi}\frac{d\Delta\boldsymbol{\phi}}{dS} + \frac{(1-\cos\Delta\phi)}{\Delta\phi}\mathbf{n}\times\frac{d\Delta\boldsymbol{\phi}}{dS} + (1-\frac{\sin\Delta\phi}{\Delta\phi})\mathbf{n}\,\mathbf{n}\cdot\frac{d\Delta\boldsymbol{\phi}}{dS}.$$

We see that $\Delta\boldsymbol{\varrho} \Rightarrow \dfrac{d\Delta\boldsymbol{\phi}}{dS}$ if $\Delta\phi \to 0$.

For the second term we express $\dfrac{d\mathbf{N}_\alpha}{dS}$ in terms of the curvature and twist at the beginning of the increment, which yields

$$\Delta\boldsymbol{q}\frac{d\mathbf{N}_\alpha}{dS}\Delta\boldsymbol{q}^\dagger = \Delta\boldsymbol{q}\epsilon_\alpha^\beta(-B_\beta\mathbf{T} + B\mathbf{N}_\beta)\Delta\boldsymbol{q}^\dagger = \epsilon_\alpha^\beta(-B_\beta\mathbf{t} + B\mathbf{n}_\beta).$$

Combining these terms,

$$\frac{d\mathbf{n}_\alpha}{dS} = \Delta\boldsymbol{\varrho}\times\mathbf{n}_\alpha + \epsilon_\alpha^\beta(-B_\beta\mathbf{t} + B\mathbf{n}_\beta).$$

The current curvature and twist are, therefore,

$$b_\alpha = \epsilon_\alpha^\beta\mathbf{t}\cdot\frac{d\mathbf{n}_\beta}{dS} = \epsilon_\alpha^\beta\Delta\boldsymbol{\varrho}\cdot(\mathbf{n}_\beta\times\mathbf{t}) - \epsilon_\alpha^\beta\epsilon_\beta^\gamma B_\gamma = \Delta\boldsymbol{\varrho}\cdot\mathbf{n}_\alpha + B_\alpha$$

and

$$b \;= \mathbf{n}_2 \cdot \frac{d\mathbf{n}_1}{dS} = \Delta\boldsymbol{\varrho} \cdot (\mathbf{n}_1 \times \mathbf{n}_2) + \epsilon_1^2 B = \Delta\boldsymbol{\varrho} \cdot \mathbf{t} + B.$$

Hence, the current curvature and twist are updated by a summation over all increments as given by

$$\begin{aligned} b_\alpha^{(n)} &= \sum_{i=1}^{n} \Delta\boldsymbol{\varrho}^{(i)} \cdot \mathbf{n}_\alpha^{(i)} + B_\alpha, \\ b^{(n)} &= \sum_{i=1}^{n} \Delta\boldsymbol{\varrho}^{(i)} \cdot \mathbf{t}_\alpha^{(i)} + B. \end{aligned}$$

First variations

The first variations of the geometric quantities are readily obtained. Recall that

$$\begin{aligned} \lambda &= \left| \frac{d\mathbf{x}}{dS} \right|, \\ \gamma_\alpha &= \mathbf{s} \cdot \mathbf{n}_\alpha \qquad (\text{with } \mathbf{s} = \lambda^{-1} \frac{d\mathbf{x}}{dS}), \\ b_\alpha &= \epsilon_\alpha^\beta \mathbf{t} \cdot \frac{d\mathbf{n}_\beta}{dS}, \\ b &= \mathbf{n}_2 \cdot \frac{d\mathbf{n}_1}{dS}, \\ \chi &= \frac{dw}{dS}. \end{aligned}$$

It follows that

$$\begin{aligned} \delta\lambda &= \mathbf{s} \cdot \frac{d\delta\mathbf{x}}{dS}, \\ \delta\gamma_\alpha &= \delta\mathbf{s} \cdot \mathbf{n}_\alpha + \mathbf{s} \cdot \delta\mathbf{n}_\alpha = -\lambda^{-1}\gamma_\alpha \delta\lambda + \lambda^{-1} \frac{d\delta\mathbf{x}}{dS} \cdot \mathbf{n}_\alpha + \mathbf{s} \cdot (\delta\boldsymbol{\phi} \times \mathbf{n}_\alpha) \\ &\approx \lambda^{-1}\mathbf{n}_\alpha \cdot \frac{d\delta\mathbf{x}}{dS} - \epsilon_\alpha^\beta \mathbf{n}_\beta \cdot \delta\boldsymbol{\phi}, \\ \delta\chi &= \frac{d\delta w}{dS}, \end{aligned}$$

where in the expression for $\delta\gamma_\alpha$ we have assumed that $\gamma_\alpha \simeq 0$ and $\mathbf{s} \simeq \mathbf{t}$. For the variations in the curvature and twist, we note that $\delta\mathbf{t} = \delta\boldsymbol{\phi} \times \mathbf{t}$. Hence, it follows that

$$\begin{aligned}
\delta b_\alpha &= \epsilon_\alpha^\beta \left(\delta\mathbf{t} \cdot \frac{d\mathbf{n}_\beta}{dS} + \mathbf{t} \cdot \frac{d\delta\mathbf{n}_\beta}{dS} \right) = \epsilon_\alpha^\beta \left[(\delta\boldsymbol{\phi} \times \mathbf{t}) \cdot \frac{d\mathbf{n}_\beta}{dS} + \mathbf{t} \cdot (\delta\boldsymbol{\phi} \times \frac{d\mathbf{n}_\beta}{dS} + \frac{d\delta\boldsymbol{\phi}}{dS} \times \mathbf{n}_\beta) \right] \\
&= \epsilon_\alpha^\beta \frac{d\delta\boldsymbol{\phi}}{dS} \cdot (\mathbf{n}_\beta \times \mathbf{t}) = \mathbf{n}_\alpha \cdot \frac{d\delta\boldsymbol{\phi}}{dS}, \\
\delta b &= \delta\mathbf{n}_2 \cdot \frac{d\mathbf{n}_1}{dS} + \mathbf{n}_2 \cdot \frac{d\delta\mathbf{n}_1}{dS} = (\delta\boldsymbol{\phi} \times \mathbf{n}_2) \cdot \frac{d\mathbf{n}_1}{dS} + \mathbf{n}_2 \cdot (\delta\boldsymbol{\phi} \times \frac{d\mathbf{n}_1}{dS} + \frac{d\delta\boldsymbol{\phi}}{dS} \times \mathbf{n}_1) \\
&= \frac{d\delta\boldsymbol{\phi}}{dS} \cdot (\mathbf{n}_1 \times \mathbf{n}_2) = \mathbf{t} \cdot \frac{d\delta\boldsymbol{\phi}}{dS}.
\end{aligned}$$

These terms will be used in the virtual work equation that will be discussed later. In using the above expressions the rotational quantities are obtained by interpolation from nodal variational quantities that are assumed to be valid for the velocity fields as

$$\begin{aligned}
\delta\mathbf{x}(\xi) &= g_N(\xi)\delta\mathbf{x}^N, \\
\delta w(\xi) &= g_N(\xi)\delta w^N, \\
\delta\boldsymbol{\phi}(\xi) &= g_N(\xi)\delta\boldsymbol{\phi}^N.
\end{aligned}$$

Solution with Newton's algorithm

Newton's algorithm involves linearization of the incremental equations. The equations must be linearized around the current (latest) state. At the integration points these equations simply take the form

$$\begin{aligned}
\mathrm{d}\lambda &= \mathbf{s} \cdot \frac{d\mathrm{d}\mathbf{x}}{dS}, \\
\mathrm{d}\gamma_\alpha &\approx \lambda^{-1}\mathbf{n}_\alpha \cdot \frac{d\mathrm{d}\mathbf{x}}{dS} - \epsilon_\alpha^\beta \mathbf{n}_\beta \cdot \mathrm{d}\boldsymbol{\phi}, \\
\mathrm{d}b_\alpha &= \frac{d\mathrm{d}\boldsymbol{\phi}}{dS} \cdot \mathbf{n}_\alpha, \\
\mathrm{d}b &= \frac{d\mathrm{d}\boldsymbol{\phi}}{dS} \cdot \mathbf{t}, \\
\mathrm{d}\chi &= \frac{d\mathrm{d}w}{dS}.
\end{aligned}$$

The corrections $\mathrm{d}\mathbf{x}$ and $\mathrm{d}w$ are readily obtained from the nodal corrections with the interpolation functions $g^N(\xi)$ as

$$\mathrm{d}\mathbf{x} = g^N(\xi)\mathrm{d}\mathbf{x}^N, \; \mathrm{d}w = g^N(\xi)\mathrm{d}w^N.$$

The corrections $\mathrm{d}\boldsymbol{\phi}$ are defined with respect to the end of the increment: we call such corrections "compound" corrections. However, it has been assumed that the total incremental rotation $\Delta\boldsymbol{\phi}$ would be interpolated with $g^N(\xi)$ as

$$\Delta\boldsymbol{\phi}(\xi) = g^N(\xi)\Delta\boldsymbol{\phi}^N.$$

Linearization of this equation yields

$$\mathrm{d}\Delta\boldsymbol{\phi}(\xi) = g^N(\xi)\mathrm{d}\Delta\boldsymbol{\phi}^N,$$

where $\mathrm{d}\Delta\boldsymbol{\phi}$ are corrections in $\Delta\boldsymbol{\phi}$ in an additive sense such that

$$\Delta\boldsymbol{\phi}^{new} = \Delta\boldsymbol{\phi}^{old} + \mathrm{d}\Delta\boldsymbol{\phi}.$$

To relate the additive correction $\mathrm{d}\Delta\boldsymbol{\phi}$ to the compound correction $\mathrm{d}\boldsymbol{\phi}$, we use the formula obtained for $\Delta\boldsymbol{\phi}$ to find

$$\mathrm{d}\boldsymbol{\phi} = \Delta\boldsymbol{q}^{\dagger}\mathrm{d}\boldsymbol{q} = \frac{\sin\Delta\phi}{\Delta\phi}\mathrm{d}\Delta\boldsymbol{\phi} + \frac{(1-\cos\Delta\phi)}{\Delta\phi}\mathbf{n}\times\mathrm{d}\Delta\boldsymbol{\phi} + (1-\frac{\sin\Delta\phi}{\Delta\phi})\mathbf{n}\,\mathbf{n}\cdot\mathrm{d}\Delta\boldsymbol{\phi}.$$

Similarly, at the nodes we find

$$\begin{aligned}\mathrm{d}\boldsymbol{\phi}^N = &\frac{\sin\Delta\phi^N}{\Delta\phi^N}\mathrm{d}\Delta\boldsymbol{\phi}^N + \frac{(1-\cos\Delta\phi^N)}{\Delta\phi^N}\mathbf{n}^N\times\mathrm{d}\Delta\boldsymbol{\phi}^N \\ &+ (1-\frac{\sin\Delta\phi^N}{\Delta\phi^N})\mathbf{n}^N\mathbf{n}^N\cdot\mathrm{d}\Delta\boldsymbol{\phi}^N \text{ (no summation)}.\end{aligned}$$

Now we assume that the difference in incremental rotation along a beam element is small; i.e.,

$$|\Delta\boldsymbol{\phi}(\xi) - \Delta\boldsymbol{\phi}^N| \ll 1,$$

which implies that either

$$|\mathbf{n}(\xi) - \mathbf{n}^N| \ll 1 \quad \text{and} \quad |\Delta\phi(\xi) - \Delta\phi^N| \ll 1$$

or

$$|\Delta\phi(\xi)| \ll 1 \quad \text{and} \quad |\Delta\phi^N| \ll 1.$$

In the first case we can use the approximations $\mathbf{n}(\xi) \approx \mathbf{n}^N \approx \overline{\mathbf{n}}$ and $\Delta\boldsymbol{\phi}(\xi) \approx \Delta\boldsymbol{\phi}^N \approx \Delta\overline{\boldsymbol{\phi}}$ which, with the use of the interpolation functions for $\mathrm{d}\Delta\boldsymbol{\phi}$, gives

$$\begin{aligned}\mathrm{d}\boldsymbol{\phi} &\approx g_N(\xi)\left[\frac{\sin\Delta\overline{\phi}}{\Delta\overline{\phi}}\mathrm{d}\Delta\boldsymbol{\phi}^N + \frac{(1-\cos\Delta\overline{\phi})}{\Delta\overline{\phi}}\overline{\mathbf{n}}\times\mathrm{d}\Delta\boldsymbol{\phi}^N + (1-\frac{\sin\Delta\overline{\phi}}{\Delta\overline{\phi}})\overline{\mathbf{n}}\,\overline{\mathbf{n}}\cdot\mathrm{d}\Delta\boldsymbol{\phi}^N\right], \\ \mathrm{d}\boldsymbol{\phi}^N &\approx \frac{\sin\Delta\overline{\phi}}{\Delta\overline{\phi}}\mathrm{d}\Delta\boldsymbol{\phi}^N + \frac{(1-\cos\Delta\overline{\phi})}{\Delta\overline{\phi}}\overline{\mathbf{n}}\times\mathrm{d}\Delta\boldsymbol{\phi}^N + (1-\frac{\sin\Delta\overline{\phi}}{\Delta\overline{\phi}})\overline{\mathbf{n}}\,\overline{\mathbf{n}}\cdot\mathrm{d}\Delta\boldsymbol{\phi}^N.\end{aligned}$$

In the second case it follows directly that

$$\mathrm{d}\boldsymbol{\phi} \approx g_N(\xi)\mathrm{d}\Delta\boldsymbol{\phi}^N,\ \mathrm{d}\boldsymbol{\phi}^N \approx \mathrm{d}\Delta\boldsymbol{\phi}^N.$$

In either case

$$\mathrm{d}\boldsymbol{\phi}(\xi) \approx g_N(\xi)\mathrm{d}\boldsymbol{\phi}^N.$$

This approximate relationship is used only in the creation of the Jacobian, so the approximation will at worst result in reduced speed of convergence.

Once a nodal update vector $\mathrm{d}\boldsymbol{\phi}^N$ is obtained, an exact update procedure is followed. This is achieved by a transformation into quaternions, use of exact quaternion update formula, and transformation of the results back into an incremental Euler rotation vector:

$$\mathrm{d}\boldsymbol{q}^N = \left(\mathrm{d}q^N\,,\,\mathrm{d}\mathbf{q}^N\right) = \left(\cos\frac{\mathrm{d}\phi^N}{2}\,,\,\frac{\mathrm{d}\boldsymbol{\phi}^N}{\mathrm{d}\phi^N}\sin\frac{\mathrm{d}\phi^N}{2}\right), \qquad (\text{no summation})$$

$$\Delta\boldsymbol{q}^N = \left(\Delta q^N\,,\,\Delta\mathbf{q}^N\right) = \left(\cos\frac{\Delta\phi^N}{2}\,,\,\frac{\Delta\boldsymbol{\phi}^N}{\Delta\phi^N}\sin\frac{\Delta\phi^N}{2}\right), \qquad (\text{no summation})$$

$$\begin{aligned}\Delta\boldsymbol{q}^N_{new} &= \mathrm{d}\boldsymbol{q}^N\Delta\boldsymbol{q}^N \\ &= \left(\mathrm{d}q^N\Delta q^N - \mathrm{d}\mathbf{q}^N\cdot\Delta\mathbf{q}^N\,,\,\mathrm{d}q^N\Delta\mathbf{q}^N + \mathrm{d}\mathbf{q}^N\Delta q^N + \mathrm{d}\mathbf{q}^N\times\Delta\mathbf{q}^N\right), \text{ (no summation)}\end{aligned}$$

$$\Delta\boldsymbol{\phi}^N_{new} = 2\frac{\mathbf{q}^N_{new}}{|\mathbf{q}^N_{new}|}\tan^{-1}(|\mathbf{q}^N_{new}|, q^N_{new}) \qquad (\text{no summation}).$$

The incremental rotation vector at the integration point is obtained by interpolation. Subsequently, we can calculate the updated integration point normals and the incremental curvature and twist.

Second variations

For the calculation of the Jacobian, we also need the second variation in the generalized quantities. These follow from the first variations:

$$
\begin{aligned}
\mathrm{d}\delta\lambda &= \mathrm{d}\mathbf{s}\cdot\frac{d\delta\mathbf{x}}{dS} = \lambda^{-1}\frac{d\mathrm{d}\mathbf{x}}{dS}\cdot(\mathbf{I}-\mathbf{ss})\cdot\frac{d\delta\mathbf{x}}{dS},\\
\mathrm{d}\delta\gamma_\alpha &= \lambda^{-2}\gamma_\alpha\mathrm{d}\lambda\delta\lambda - \lambda^{-1}\gamma_\alpha\mathrm{d}\delta\lambda - \lambda^{-1}\delta\lambda\mathrm{d}\gamma_\alpha - \lambda^{-2}\mathrm{d}\lambda\frac{d\delta\mathbf{x}}{dS}\cdot\mathbf{n}_\alpha\\
&\quad + \lambda^{-1}\frac{d\delta\mathbf{x}}{dS}\cdot(\mathrm{d}\boldsymbol{\phi}\times\mathbf{n}_\alpha) - \lambda^{-1}\mathrm{d}\lambda\,\mathbf{s}\cdot(\delta\boldsymbol{\phi}\times\mathbf{n}_\alpha) + \lambda^{-1}\frac{d\mathrm{d}\mathbf{x}}{dS}\cdot(\delta\boldsymbol{\phi}\times\mathbf{n}_\alpha)\\
&\quad + \frac{1}{2}\mathbf{s}\cdot\left[\delta\boldsymbol{\phi}\times(\mathrm{d}\boldsymbol{\phi}\times\mathbf{n}_\alpha) + \mathrm{d}\boldsymbol{\phi}\times(\delta\boldsymbol{\phi}\times\mathbf{n}_\alpha)\right]\\
&\approx -\lambda^{-1}(\delta\lambda\,\mathrm{d}\gamma_\alpha + \mathrm{d}\lambda\,\delta\gamma_\alpha) + \frac{1}{2}\delta\boldsymbol{\phi}\cdot(\mathbf{n}_\alpha\mathbf{t}+\mathbf{t}\,\mathbf{n}_\alpha)\cdot\mathrm{d}\boldsymbol{\phi}\\
&\quad - \lambda^{-1}\epsilon_\alpha^\beta(\delta\lambda\,\mathrm{d}\boldsymbol{\phi}\cdot\mathbf{n}_\beta + \mathrm{d}\lambda\,\delta\boldsymbol{\phi}\cdot\mathbf{n}_\beta) + \lambda^{-1}\epsilon_\alpha^\beta(\frac{d\delta\mathbf{x}}{dS}\cdot\mathbf{n}_\beta\mathbf{t}\cdot\mathrm{d}\boldsymbol{\phi} + \frac{d\mathrm{d}\mathbf{x}}{dS}\cdot\mathbf{n}_\beta\mathbf{t}\cdot\delta\boldsymbol{\phi})\\
&\approx -\lambda^{-2}\frac{d\delta\mathbf{x}}{dS}\cdot(\mathbf{n}_\alpha\mathbf{t}+\mathbf{t}\,\mathbf{n}_\alpha)\cdot\frac{d\mathrm{d}\mathbf{x}}{dS} + \lambda^{-1}\epsilon_\alpha^\beta\left(\frac{d\delta\mathbf{x}}{dS}\cdot\mathbf{n}_\beta\mathbf{t}\cdot\mathrm{d}\boldsymbol{\phi} + \delta\boldsymbol{\phi}\cdot\mathbf{t}\,\mathbf{n}_\beta\cdot\frac{d\mathrm{d}\mathbf{x}}{dS}\right)\\
&\quad + \frac{1}{2}\delta\boldsymbol{\phi}\cdot(\mathbf{n}_\alpha\mathbf{t}+\mathbf{t}\,\mathbf{n}_\alpha)\cdot\dot{\boldsymbol{\phi}},
\end{aligned}
$$

where we have again used $\gamma_\alpha \approx 0$.

For the second variation of the curvature we find

$$
\begin{aligned}
\mathrm{d}\delta b_\alpha =& \epsilon_\alpha^\beta\left[\mathrm{d}\delta\mathbf{t}\cdot\frac{d\mathbf{n}_\beta}{dS} + \delta\mathbf{t}\cdot\frac{d\mathrm{d}\mathbf{n}_\beta}{dS} + \mathrm{d}\mathbf{t}\cdot\frac{d\delta\mathbf{n}_\beta}{dS} + \mathbf{t}\cdot\frac{d\mathrm{d}\delta\mathbf{n}_\beta}{dS}\right]\\
=& \epsilon_\alpha^\beta\left[\frac{1}{2}\delta\boldsymbol{\phi}\cdot\left(\mathbf{t}\frac{d\mathbf{n}_\beta}{dS} + \frac{d\mathbf{n}_\beta}{dS}\mathbf{t}\right)\cdot\mathrm{d}\boldsymbol{\phi} - \left(\mathbf{t}\cdot\frac{d\mathbf{n}_\beta}{dS}\right)(\delta\boldsymbol{\phi}\cdot\mathrm{d}\boldsymbol{\phi})\right.\\
&+ (\delta\boldsymbol{\phi}\times\mathbf{t})\cdot\left(\frac{d\mathrm{d}\boldsymbol{\phi}}{dS}\times\mathbf{n}_\beta + \mathrm{d}\boldsymbol{\phi}\times\frac{d\mathbf{n}_\beta}{dS}\right) + (\mathrm{d}\boldsymbol{\phi}\times\mathbf{t})\cdot\left(\frac{d\delta\boldsymbol{\phi}}{dS}\times\mathbf{n}_\beta + \delta\boldsymbol{\phi}\times\frac{d\mathbf{n}_\beta}{dS}\right)\\
&+ \mathbf{t}\cdot\left(\frac{1}{2}\frac{d\mathrm{d}\boldsymbol{\phi}}{dS}\delta\boldsymbol{\phi}\cdot\mathbf{n}_\beta + \frac{1}{2}\mathrm{d}\boldsymbol{\phi}\frac{d\delta\boldsymbol{\phi}}{dS}\cdot\mathbf{n}_\beta + \frac{1}{2}\mathrm{d}\boldsymbol{\phi}\,\delta\boldsymbol{\phi}\cdot\frac{d\mathbf{n}_\beta}{dS}\right)\\
&+ \mathbf{t}\cdot\left(\frac{1}{2}\frac{d\delta\boldsymbol{\phi}}{dS}\mathrm{d}\boldsymbol{\phi}\cdot\mathbf{n}_\beta + \frac{1}{2}\delta\boldsymbol{\phi}\frac{d\mathrm{d}\boldsymbol{\phi}}{dS}\cdot\mathbf{n}_\beta + \frac{1}{2}\delta\boldsymbol{\phi}\,\mathrm{d}\boldsymbol{\phi}\cdot\frac{d\mathbf{n}_\beta}{dS}\right)\\
&\left.- \mathbf{t}\cdot(\frac{d\delta\boldsymbol{\phi}}{dS}\cdot\mathrm{d}\boldsymbol{\phi}\,\mathbf{n}_\beta + \delta\boldsymbol{\phi}\cdot\frac{d\mathrm{d}\boldsymbol{\phi}}{dS}\mathbf{n}_\beta + \delta\boldsymbol{\phi}\cdot\mathrm{d}\boldsymbol{\phi}\frac{d\mathbf{n}_\beta}{dS})\right].
\end{aligned}
$$

Rewriting the second and third terms and combining the others yields

$$
\begin{aligned}
\mathrm{d}\delta b_\alpha =& \epsilon_\alpha^\beta \Bigg[\delta\boldsymbol{\phi} \cdot \left(\mathbf{t} \frac{d\mathbf{n}_\beta}{dS} + \frac{d\mathbf{n}_\beta}{dS} \mathbf{t} \right) \cdot \mathrm{d}\boldsymbol{\phi} - 2 \left(\mathbf{t} \cdot \frac{d\mathbf{n}_\beta}{dS} \right) (\delta\boldsymbol{\phi} \cdot \mathrm{d}\boldsymbol{\phi}) \\
& + \frac{1}{2} \frac{d\delta\boldsymbol{\phi}}{dS} \cdot (\mathbf{n}_\beta \mathbf{t} + \mathbf{t}\, \mathbf{n}_\beta) \cdot \mathrm{d}\boldsymbol{\phi} + \frac{1}{2} \delta\boldsymbol{\phi} \cdot (\mathbf{n}_\beta \mathbf{t} + \mathbf{t}\, \mathbf{n}_\beta) \cdot \frac{d\mathrm{d}\boldsymbol{\phi}}{dS} \\
& - \delta\boldsymbol{\phi} \cdot \mathbf{n}_\beta \mathbf{t} \cdot \frac{d\mathrm{d}\boldsymbol{\phi}}{dS} + \left(\mathbf{t} \cdot \frac{d\mathbf{n}_\beta}{dS} \right) (\delta\boldsymbol{\phi} \cdot \mathrm{d}\boldsymbol{\phi}) - \delta\boldsymbol{\phi} \cdot \frac{d\mathbf{n}_\beta}{dS} \mathbf{t} \cdot \mathrm{d}\boldsymbol{\phi} \\
& - \mathrm{d}\boldsymbol{\phi} \cdot \mathbf{n}_\beta \mathbf{t} \cdot \frac{d\delta\boldsymbol{\phi}}{dS} + \left(\mathbf{t} \cdot \frac{d\mathbf{n}_\beta}{dS} \right) (\mathrm{d}\boldsymbol{\phi} \cdot \delta\boldsymbol{\phi}) - \mathrm{d}\boldsymbol{\phi} \cdot \frac{d\mathbf{n}_\beta}{dS} \mathbf{t} \cdot \delta\boldsymbol{\phi} \Bigg] \\
=& \frac{1}{2} \epsilon_\alpha^\beta \left[\frac{d\delta\boldsymbol{\phi}}{dS} \cdot (\mathbf{n}_\beta \mathbf{t} - \mathbf{t}\, \mathbf{n}_\beta) \cdot \mathrm{d}\boldsymbol{\phi} - \delta\boldsymbol{\phi} \cdot (\mathbf{n}_\beta \mathbf{t} - \mathbf{t}\, \mathbf{n}_\beta) \cdot \frac{d\mathrm{d}\boldsymbol{\phi}}{dS} \right].
\end{aligned}
$$

The second variation of twist is

$$
\begin{aligned}
\mathrm{d}\delta b =& \mathrm{d}\delta\mathbf{n}_2 \cdot \frac{d\mathbf{n}_1}{dS} + \delta\mathbf{n}_2 \cdot \frac{d\mathrm{d}\mathbf{n}_1}{dS} + \mathrm{d}\mathbf{n}_2 \cdot \frac{d\delta\mathbf{n}_1}{dS} + \mathbf{n}_2 \cdot \frac{d\mathrm{d}\delta\mathbf{n}_1}{dS} \\
=& \frac{1}{2} \delta\boldsymbol{\phi} \cdot \left(\mathbf{n}_2 \frac{d\mathbf{n}_1}{dS} + \frac{d\mathbf{n}_1}{dS} \mathbf{n}_2 \right) \cdot \mathrm{d}\boldsymbol{\phi} - \left(\mathbf{n}_2 \cdot \frac{d\mathbf{n}_1}{dS} \right) (\delta\boldsymbol{\phi} \cdot \mathrm{d}\boldsymbol{\phi}) \\
& + (\delta\boldsymbol{\phi} \times \mathbf{n}_2) \cdot \left(\frac{d\mathrm{d}\boldsymbol{\phi}}{dS} \times \mathbf{n}_1 + \mathrm{d}\boldsymbol{\phi} \times \frac{d\mathbf{n}_1}{dS} \right) + (\mathrm{d}\boldsymbol{\phi} \times \mathbf{n}_2) \cdot \left(\frac{d\delta\boldsymbol{\phi}}{dS} \times \mathbf{n}_1 + \delta\boldsymbol{\phi} \times \frac{d\mathbf{n}_1}{dS} \right) \\
& + \mathbf{n}_2 \cdot \left(\frac{1}{2} \frac{d\mathrm{d}\boldsymbol{\phi}}{dS} \delta\boldsymbol{\phi} \cdot \mathbf{n}_1 + \frac{1}{2} \mathrm{d}\boldsymbol{\phi} \frac{d\delta\boldsymbol{\phi}}{dS} \cdot \mathbf{n}_1 + \frac{1}{2} \mathrm{d}\boldsymbol{\phi}\, \delta\boldsymbol{\phi} \cdot \frac{d\mathbf{n}_1}{dS} \right) \\
& + \mathbf{n}_2 \cdot \left(\frac{1}{2} \frac{d\mathrm{d}\boldsymbol{\phi}}{dS} \mathrm{d}\boldsymbol{\phi} \cdot \mathbf{n}_1 + \frac{1}{2} \delta\boldsymbol{\phi} \frac{d\mathrm{d}\boldsymbol{\phi}}{dS} \cdot \mathbf{n}_1 + \frac{1}{2} \delta\boldsymbol{\phi}\, \mathrm{d}\boldsymbol{\phi} \cdot \frac{d\mathbf{n}_1}{dS} \right) \\
& - \mathbf{n}_2 \cdot \left(\frac{d\delta\boldsymbol{\phi}}{dS} \cdot \mathrm{d}\boldsymbol{\phi}\, \mathbf{n}_1 + \delta\boldsymbol{\phi} \cdot \frac{d\mathrm{d}\boldsymbol{\phi}}{dS} \mathbf{n}_1 + \delta\boldsymbol{\phi} \cdot \mathrm{d}\boldsymbol{\phi} \frac{d\mathbf{n}_1}{dS} \right).
\end{aligned}
$$

Then, following the same procedure as for $\mathrm{d}\delta b_\alpha$,

$$
\begin{aligned}
\mathrm{d}\delta b =& \delta\boldsymbol{\phi} \cdot \left(\mathbf{n}_2 \frac{d\mathbf{n}_1}{dS} + \frac{d\mathbf{n}_1}{dS} \mathbf{n}_2 \right) \cdot \delta\boldsymbol{\phi} - 2 \left(\mathbf{n}_2 \cdot \frac{d\mathbf{n}_1}{dS} \right) (\delta\boldsymbol{\phi} \cdot \mathrm{d}\boldsymbol{\phi}) \\
& + \frac{1}{2} \frac{d\delta\boldsymbol{\phi}}{dS} \cdot (\mathbf{n}_1 \mathbf{n}_2 + \mathbf{n}_2 \mathbf{n}_1) \cdot \mathrm{d}\boldsymbol{\phi} + \frac{1}{2} \delta\boldsymbol{\phi} \cdot (\mathbf{n}_1 \mathbf{n}_2 + \mathbf{n}_2 \mathbf{n}_1) \cdot \frac{d\mathrm{d}\boldsymbol{\phi}}{dS} \\
& - \delta\boldsymbol{\phi} \cdot \mathbf{n}_1 \mathbf{n}_2 \cdot \frac{d\mathrm{d}\boldsymbol{\phi}}{dS} + \left(\mathbf{n}_2 \cdot \frac{d\mathbf{n}_1}{dS} \right) (\delta\boldsymbol{\phi} \cdot \mathrm{d}\boldsymbol{\phi}) - \delta\boldsymbol{\phi} \cdot \frac{d\mathbf{n}_1}{dS} \mathbf{n}_2 \cdot \mathrm{d}\boldsymbol{\phi} \\
& - \mathrm{d}\boldsymbol{\phi} \cdot \mathbf{n}_1 \mathbf{n}_2 \cdot \frac{d\mathrm{d}\boldsymbol{\phi}}{dS} + \left(\mathbf{n}_2 \cdot \frac{d\mathbf{n}_1}{dS} \right) (\mathrm{d}\boldsymbol{\phi} \cdot \delta\boldsymbol{\phi}) - \mathrm{d}\boldsymbol{\phi} \cdot \frac{d\mathbf{n}_1}{dS} \mathbf{n}_2 \cdot \delta\boldsymbol{\phi} \\
=& \frac{1}{2} \left[\frac{d\delta\boldsymbol{\phi}}{dS} \cdot (\mathbf{n}_1 \mathbf{n}_2 - \mathbf{n}_2 \mathbf{n}_1) \cdot \mathrm{d}\boldsymbol{\phi} - \delta\boldsymbol{\phi} \cdot (\mathbf{n}_1 \mathbf{n}_2 - \mathbf{n}_2 \mathbf{n}_1) \cdot \frac{d\mathrm{d}\boldsymbol{\phi}}{dS} \right].
\end{aligned}
$$

Finally, we note

$$\mathrm{d}\delta\chi = 0.$$

The resulting second variations are summarized as

$$\begin{aligned}
\mathrm{d}\delta\lambda &= \lambda^{-1}\frac{d\mathrm{d}\mathbf{x}}{dS}\cdot(\mathbf{I}-\mathbf{ss})\cdot\frac{d\delta\mathbf{x}}{dS},\\
\mathrm{d}\delta\gamma_\alpha &= -\lambda^{-2}\frac{d\delta\mathbf{x}}{dS}\cdot(\mathbf{n}_\alpha\mathbf{s}+\mathbf{s}\,\mathbf{n}_\alpha)\cdot\frac{d\mathrm{d}\mathbf{x}}{dS}+\lambda^{-1}\epsilon_\alpha^\beta\left(\frac{d\delta\mathbf{x}}{dS}\cdot\mathbf{n}_\beta\mathbf{t}\cdot\mathrm{d}\boldsymbol{\phi}+\delta\boldsymbol{\phi}\cdot\mathbf{n}_\beta\mathbf{t}\cdot\frac{d\mathrm{d}\mathbf{x}}{dS}\right)\\
&\quad+\frac{1}{2}\delta\boldsymbol{\phi}\cdot(\mathbf{n}_\alpha\mathbf{t}+\mathbf{t}\,\mathbf{n}_\alpha)\cdot\dot{\boldsymbol{\phi}},\\
\mathrm{d}\delta b_\alpha &= \frac{1}{2}\epsilon_\alpha^\beta\left[\frac{d\delta\boldsymbol{\phi}}{dS}\cdot(\mathbf{n}_\beta\mathbf{t}-\mathbf{t}\,\mathbf{n}_\beta)\cdot\mathrm{d}\boldsymbol{\phi}-\delta\boldsymbol{\phi}\cdot(\mathbf{n}_\beta\mathbf{t}-\mathbf{t}\,\mathbf{n}_\beta)\cdot\frac{d\mathrm{d}\boldsymbol{\phi}}{dS}\right],\\
\mathrm{d}\delta b &= \frac{1}{2}\left[\frac{d\delta\boldsymbol{\phi}}{dS}\cdot(\mathbf{n}_1\mathbf{n}_2-\mathbf{n}_2\mathbf{n}_1)\cdot\mathrm{d}\boldsymbol{\phi}-\delta\boldsymbol{\phi}\cdot(\mathbf{n}_1\mathbf{n}_2-\mathbf{n}_2\mathbf{n}_1)\cdot\frac{d\mathrm{d}\boldsymbol{\phi}}{dS}\right],\\
\mathrm{d}\delta\chi &= 0.
\end{aligned}$$

Strains

The gradient of the current position of a point in the section with respect to the coordinate S is

$$\frac{\partial\hat{\mathbf{x}}}{\partial S}=\frac{d\mathbf{x}}{dS}+\frac{df}{dS}S^\alpha\mathbf{n}_\alpha+fS^\alpha\frac{d\mathbf{n}_\alpha}{dS}+\frac{dw}{dS}\psi\mathbf{t}+w\psi\frac{d\mathbf{t}}{dS}.$$

We will keep only terms up to order $\frac{h}{\ell}$. We assume that $\frac{df}{dS}\ll\frac{1}{\ell}$, and—hence—the second term can be neglected. We also assume that $w\psi=O(\frac{h^2}{\ell})$, and—since $\frac{d\mathbf{t}}{dS}=O(\frac{1}{\ell})$—we can neglect the last term as well. However, the warping function may vary rapidly near the ends where warping is constrained. Hence, $\frac{dw}{dS}\psi=O(\frac{h}{\ell})$ and should be preserved. With those approximations,

$$\frac{\partial\hat{\mathbf{x}}}{\partial S}\approx\frac{d\mathbf{x}}{dS}+fS^\alpha\frac{d\mathbf{n}_\alpha}{dS}+\frac{dw}{dS}\psi\mathbf{t}=\lambda\mathbf{s}+fS^\alpha\epsilon_\alpha^\beta(-b_\beta\mathbf{t}+b\mathbf{n}_\beta)+\chi\psi\mathbf{t}.$$

The gradients with respect to S^α are

$$\frac{\partial\hat{\mathbf{x}}}{\partial S^\alpha}=f\mathbf{n}_\alpha+w\frac{\partial\psi}{\partial S^\alpha}\mathbf{t}.$$

Correspondingly, in the original configuration

$$\frac{\partial \hat{\mathbf{X}}}{\partial S} = \frac{d\mathbf{X}}{dS} + S^\alpha \frac{d\mathbf{N}_\alpha}{dS} = \mathbf{T} + S^\alpha \epsilon_\alpha^\beta (-B_\beta \mathbf{T} + B\mathbf{N}_\beta),$$
$$\frac{\partial \hat{\mathbf{X}}}{\partial S^\alpha} = \mathbf{N}_\alpha.$$

The above relations are readily inverted to yield

$$\frac{\partial S}{\partial \hat{\mathbf{X}}} = (1 - S^\gamma \epsilon_\gamma^\delta B_\delta)^{-1} \mathbf{T},$$
$$\frac{\partial S^\alpha}{\partial \hat{\mathbf{X}}} = \mathbf{N}_\alpha - (1 - S^\gamma \epsilon_\gamma^\delta B_\delta)^{-1} S^\beta \epsilon_\beta^\alpha B\mathbf{T}.$$

The deformation gradient then becomes

$$\begin{aligned}\mathbf{F} =& \frac{\partial \hat{\mathbf{x}}}{\partial S}\frac{\partial S}{\partial \hat{\mathbf{X}}} + \frac{\partial \hat{\mathbf{x}}}{\partial S^\alpha}\frac{\partial S^\alpha}{\partial \hat{\mathbf{X}}} \\ =& (1 - S^\gamma \epsilon_\gamma^\delta B_\delta)^{-1} \Big[\lambda \mathbf{s}\ \mathbf{T} + (-f S^\alpha \epsilon_\alpha^\beta b_\beta + \chi\psi)\mathbf{t}\ \mathbf{T} + f S^\alpha \epsilon_\alpha^\beta b\ \mathbf{n}_\beta \mathbf{T} \\ &+ (1 - S^\alpha \epsilon_\alpha^\beta B_\beta) f\ \mathbf{n}_\alpha \mathbf{N}_\alpha + (1 - S^\alpha \epsilon_\alpha^\beta B_\beta) w \frac{d\psi}{dS^\eta} \mathbf{t}\ \mathbf{N}_\eta \\ &- S^\alpha \epsilon_\alpha^\beta B\ f \mathbf{n}_\beta \mathbf{T} - S^\alpha \epsilon_\alpha^\beta B w \frac{\partial \psi}{\partial S^\beta} \mathbf{t}\ \mathbf{T} \Big].\end{aligned}$$

We define the initial length ratio R as

$$R = 1 - S^\alpha \epsilon_\alpha^\beta B_\beta.$$

In the expression enclosed within square brackets above, terms of order h^2/ℓ^2 can again be neglected. However, it is assumed that the section may have low resistance to torsion and that, hence, the warping and twist may be large. This is particularly true for thin walled open sections. Hence, we obtain:

$$\begin{aligned}\mathbf{F} = R^{-1} \Bigg\{ \lambda \mathbf{s}\mathbf{T} + \left[-S^\alpha \epsilon_\alpha^\beta \left(f b_\beta + B w \frac{\partial \psi}{\partial S^\beta} \right) + \chi\psi \right] \mathbf{t}\mathbf{T} + \\ f S^\alpha \epsilon_\alpha^\beta (b - B) \mathbf{n}_\beta \mathbf{T} + w \frac{\partial \psi}{\partial S^\beta} \mathbf{t}\mathbf{N}_\beta + R f \mathbf{n}_\beta \mathbf{N}_\beta \Bigg\}.\end{aligned}$$

We calculate the components of $\mathbf{F}$ in a corotational system with the approximation $\mathbf{t} \cdot \mathbf{s} = 1$. This provides

$$
\begin{aligned}
F_{11} &= \mathbf{t}\cdot\mathbf{F}\cdot\mathbf{T} = R^{-1}\left[\lambda - S^{\beta}\epsilon^{\alpha}_{\beta}\left(f b_{\alpha} + Bw\frac{\partial\psi}{\partial S^{\alpha}}\right) + \chi\psi\right],\\
F_{(\alpha+1)1} &= \mathbf{n}_{\alpha}\cdot\mathbf{F}\cdot\mathbf{T} = R^{-1}(\lambda\gamma_{\alpha} + \epsilon^{\alpha}_{\beta} f S^{\beta}(b-B)),\\
F_{1(\beta+1)} &= \mathbf{t}\cdot\mathbf{F}\cdot\mathbf{N}_{\beta} = R^{-1} w\frac{d\psi}{dS^{\beta}},\\
F_{(\alpha+1)(\beta+1)} &= \mathbf{n}_{\alpha}\cdot\mathbf{F}\cdot\mathbf{N}_{\beta} = f\,\delta_{\alpha\beta}.
\end{aligned}
$$

We again neglect all terms of order h^n/ℓ^n for $n \geq 2$, except the term involving Bw. The equations then simplify to

$$
\begin{aligned}
F_{11} &= \lambda - S^{\beta}\epsilon^{\alpha}_{\beta}\left(f b_{\alpha} + Bw\frac{\partial\psi}{\partial S^{\alpha}} - \lambda B_{\alpha}\right) + \chi\psi,\\
F_{(\alpha+1)1} &= \lambda\gamma_{\alpha} + \epsilon^{\alpha}_{\beta} f S^{\beta}(b-B),\\
F_{1(\beta+1)} &= w\frac{d\psi}{dS^{\beta}},\\
F_{(\alpha+1)(\beta+1)} &= f\,\delta_{\alpha\beta}.
\end{aligned}
$$

Consistent with traditional shell and beam theories, we slightly adapt the term involving the initial curvature B_{α}—instead of multiplying it with λ, we multiply it with f. Such a change does not significantly increase the error in the curvature calculation, since we do not properly account for the initial curvature in the volume integration anyway. Hence, we find for F_{11}:

$$
F_{11} = \lambda - S^{\beta}\epsilon^{\alpha}_{\beta}\left(f(b_{\alpha} - B_{\alpha}) + Bw\frac{\partial\psi}{\partial S^{\alpha}}\right) + \chi\psi.
$$

We now make a multiplicative decomposition of $\mathbf{F}$ into a "stretch" part $\mathbf{F}^s$ and a "distortion" part $\mathbf{F}^d$ such that $\mathbf{F} = \mathbf{F}^d\cdot\mathbf{F}^s$.

For $\mathbf{F}^s$ we choose

$$
F^s_{11} = \lambda, \quad F^s_{(\alpha+1)1} = F^s_{1(\beta+1)} = 0, \quad \text{and } F^s_{(\alpha+1)(\beta+1)} = f\,\delta_{\alpha\beta};
$$

and, hence, $\mathbf{F}^d$ is

$$
\begin{aligned}
F^d_{11} &= \lambda^{-1}F_{11} = 1 - S^{\beta}\epsilon^{\alpha}_{\beta}\lambda^{-1}\left(f(b_{\alpha} - B_{\alpha}) + Bw\frac{\partial\psi}{\partial S^{\alpha}}\right) + \lambda^{-1}\chi\psi,\\
F^d_{(\alpha+1)1} &= \lambda^{-1}F_{(\alpha+1)1} = \gamma_{\alpha} + \epsilon^{\alpha}_{\beta} f\lambda^{-1} S^{\beta}(b-B),\\
F^d_{1(\beta+1)} &= f^{-1}F_{1(\beta+1)} = f^{-1} w\frac{\partial\psi}{\partial S^{\beta}},\\
F^d_{(\alpha+1)(\beta+1)} &= f^{-1}F_{(\alpha+1)(\beta+1)} = \delta_{\alpha\beta}.
\end{aligned}
$$

Since $\mathbf{F}^s$ is a diagonal tensor, the logarithmic "stretch" strains are immediately available as

$$e^s_{11} = \ln\lambda, \quad \gamma^s_{(\alpha+1)1} = 0, \quad \gamma^s_{1(\alpha+1)} = 0, \quad e^s_{(\alpha+1)(\beta+1)} = \delta_{\alpha\beta}\ln(f).$$

Since the "distortional" strains are small, we obtain them from $\mathbf{F}^d$ with the Green-Lagrange formula:

$$\mathbf{E}^d = \frac{1}{2}\left(\mathbf{F}^{dT}\cdot\mathbf{F}^d - \mathbf{I}\right).$$

For the components this yields

$$\begin{aligned}
E^d_{11} =& -S^\beta\epsilon^\beta_\alpha\lambda^{-1}\left(f(b_\alpha - B_\alpha) + Bw\frac{\partial\psi}{\partial S^\alpha}\right) + \lambda^{-1}\chi\psi \\
&- S^\beta\epsilon^\beta_\alpha\left(b_\alpha - B_\alpha + Bw\frac{\partial\psi}{\partial S^\alpha}\right)\lambda^{-1}\chi\psi + \gamma_\alpha\epsilon^\beta_\alpha f\lambda^{-1}S^\beta(b-B) \\
&+ \frac{1}{2}S^\gamma\epsilon^\alpha_\gamma S^\beta\epsilon^\delta_\beta\left(b_\alpha - B_\alpha + Bw\frac{\partial\psi}{\partial S^\alpha}\right)\left(b_\delta - B_\delta + Bw\frac{\partial\psi}{\partial S^\delta}\right) \\
&+ \frac{1}{2}\lambda^{-2}\chi^2\psi^2 + \frac{1}{2}\gamma_\alpha\gamma_\alpha + \frac{1}{2}f^2\lambda^{-2}S^\alpha S^\alpha(b-B)^2,
\end{aligned}$$

$$\begin{aligned}
E_{(\alpha+1)1} =& \frac{1}{2}\Bigg[f^{-1}w\frac{\partial\psi}{\partial S^\alpha} + \gamma_\alpha + \epsilon^\beta_\alpha f\lambda^{-1}S^\beta(b-B) \\
&- \lambda^{-1}w\frac{\partial\psi}{\partial S^\alpha}\epsilon^\gamma_\beta\left(b_\gamma - B_\gamma + Bw\frac{\partial\psi}{\partial S^\gamma}\right) + f^{-1}w\frac{\partial\psi}{\partial S^\alpha}\lambda^{-1}\chi\psi\Bigg], \\
E_{(\alpha+1)(\beta+1)} =& \frac{1}{2}f^{-2}w^2\frac{\partial\psi}{\partial S^\alpha}\frac{\partial\psi}{\partial S^\beta}.
\end{aligned}$$

Note that these strains are small. Since the various terms have different dependence on the cross-sectional coordinates, this leads to the conditions

$$\begin{aligned}
S^\beta\epsilon^\beta_\alpha f\lambda^{-1}(b_\alpha - B_\alpha) &\ll 1, \\
\lambda^{-1}\chi\psi &\ll 1, \\
\gamma_\alpha &\ll 1, \\
f^{-1}w\frac{\partial\psi}{\partial S^\alpha} + \epsilon^\beta_\alpha f\lambda^{-1}S^\beta(b-B) &\ll 1.
\end{aligned}$$

The last condition will generally require that both $\boldsymbol{w}$ and $b - B$ are small, since $\partial\psi/\partial S^\alpha$ will not be proportional to $\epsilon^\beta_\alpha S^\beta$. However, for thin walled open section beams the proportionality is approximately satisfied and, therefore, we obtain in that case

$$w\frac{\partial\psi}{\partial S^\alpha} \approx -f^2\lambda^{-1}\epsilon^\beta_\alpha S^\beta(b-B).$$

Note that within the desired accuracy this equation even holds for other sections: in that case both the right-hand and left-hand side are very small. Substitution in the expression for the Green-Lagrange strain yields the (small) distortional strains:

$$\begin{aligned}
e^d_{11} &= -f\lambda^{-1}S^\beta\epsilon^\alpha_\beta(b_\alpha - B_\alpha) + \lambda^{-1}\chi\psi + \frac{1}{2}f^2\lambda^{-2}S^\alpha S^\alpha(b^2 - B^2),\\
\gamma^d_{(\alpha+1)1} &= \gamma_\alpha + \epsilon^\alpha_\beta f\lambda^{-1}S^\beta(b - B) + f^{-1}w\frac{\partial\psi}{\partial S^\alpha},\\
\gamma^d_{1(\alpha+1)} &= \gamma^d_{(\alpha+1)1},\\
e^d_{(\alpha+1)(\beta+1)} &= +\frac{1}{2}f^2\lambda^{-2}S^\alpha S^\beta(b - B)^2.
\end{aligned}$$

The total strains are obtained simply by addition as

$$\begin{aligned}
e_{11} &= \ln\lambda - f\lambda^{-1}S^\beta\epsilon^\alpha_\beta(b_\alpha - B_\alpha) + \lambda^{-1}\chi\psi + \frac{1}{2}f^2\lambda^{-2}S^\alpha S^\alpha(b^2 - B^2),\\
\gamma_{(\alpha+1)1} &= \gamma_\alpha + \epsilon^\alpha_\beta f\lambda^{-1}S^\beta(b - B) + f^{-1}w\frac{\partial\psi}{\partial S^\alpha},\\
\gamma_{1(\alpha+1)} &= \gamma_{(\alpha+1)1},\\
e_{(\alpha+1)(\beta+1)} &= \delta_{\alpha\beta}\ln(f) + \frac{1}{2}f^2\lambda^{-2}S^\alpha S^\beta(b - B)^2.
\end{aligned}$$

We assume that there are no stresses in the $(\alpha+1)(\beta+1)$ directions. Hence, the strains in these directions do not contribute to virtual work and do not need to be considered any further.

It is useful to split the total warping $\boldsymbol{w}$ in two parts: a part due to "free" warping w_f minus a part due to warping prevention w_p: $w = w_f - w_p$. We assume that the warping function ψ is chosen such that the free warping is related to the twist with the relation

$$w_f = f^2\lambda^{-1}(b - B) \qquad \text{and, hence,} \quad w_p = f^2\lambda^{-1}(b - B) - w.$$

This makes it possible to write the expression for $\gamma_{(\alpha+1)1}$ as

$$\gamma_{(\alpha+1)1} = \gamma_\alpha + f\lambda^{-1}\left(\epsilon^\alpha_\beta S^\beta + \frac{\partial\psi}{\partial S^\alpha}\right)(b - B) - f^{-1}w_p\frac{\partial\psi}{\partial S^\alpha}.$$

It is desirable to choose the cross-sectional resultants such that they are completely uncoupled. In addition, we assume that the axial strain variation across the section due to the second-order term in the twist is not significant. Therefore, we consider only the *average* axial strain due to the second-order twist term. Hence, we introduce the average axial strain

$$\overline{e} = \ln\lambda - f\lambda^{-1}S^\beta_c\epsilon^\alpha_\beta(b_\alpha - B_\alpha) + \lambda^{-1}\chi\overline{\psi} + \frac{1}{2}\left(\frac{I_p}{A} + S^\alpha_c S^\alpha_c\right)f^2\lambda^{-2}(b^2 - B^2),$$

where S_c^β are the coordinates of the centroid, I_p is the polar moment of inertia, and $\overline{\psi}$ is the average value of the warping function:

$$\overline{\psi} = \frac{1}{A}\int_A \psi \, dA.$$

Similarly, we introduce the average shear strain

$$\overline{\gamma}_\alpha = \gamma_\alpha + f\lambda^{-1}\left(\epsilon_\beta^\alpha S_c^\beta + \frac{1}{A}\int_A \frac{\partial \psi}{\partial S^\alpha}\, dA\right)(b - B) - f^{-1} w_p \frac{1}{A}\int_A \frac{\partial \psi}{\partial S^\alpha}\, dA.$$

This last expression can be simplified by the introduction of the shear center coordinates S_s^β, which are related to the warping function by

$$\epsilon_\beta^\alpha S_s^\beta = \epsilon_\beta^\alpha S_c^\beta + \frac{1}{A}\int_A \frac{\partial \psi}{\partial S^\alpha}\, dA, \quad \text{or} \quad S_s^\beta = S_c^\beta + \frac{1}{A}\int_A \epsilon_\beta^\alpha \frac{\partial \psi}{\partial S^\alpha}\, dA.$$

This yields

$$\overline{\gamma}_\alpha = \gamma_\alpha + f\lambda^{-1}\epsilon_\beta^\alpha S_s^\beta (b - B) + f^{-1} w_p \epsilon_\beta^\alpha (S_c^\beta - S_s^\beta).$$

Note that the average value is in fact the value at the shear center if warping prevention is absent ($w_p = 0$). However, for full warping prevention ($w_p = f^2\lambda^{-1}(b - B)$) the average value corresponds to the value obtained at the centroid.

Instead of the original warping function $\psi(S^\alpha)$ we now introduce a modified warping function $\Omega(S^\alpha)$ related to ψ by

$$\Omega(S^\alpha) \stackrel{\text{def}}{=} \psi(S^\alpha) - \overline{\psi}.$$

This function in fact represents the classical definition of a warping function with an area weighted average of zero. The average value $\overline{\psi}$ can be obtained from the classical warping function with the condition that $\psi(0) = 0$:

$$\overline{\psi} = \psi(S^\alpha) - \Omega(S^\alpha) = \psi(0) - \Omega(0) = -\Omega(0) = -\Omega_\circ.$$

The expression for the average axial strain then becomes

$$\overline{e} = \ln\lambda - f\lambda^{-1} S_c^\beta \epsilon_\beta^\alpha (b_\alpha - B_\alpha) - \lambda^{-1}\chi\,\Omega_\circ + \frac{1}{2}\left(\frac{I_p}{A} + S_c^\alpha S_c^\alpha\right) f^2\lambda^{-2}(b^2 - B^2),$$

and the location of the shear center is

$$S_s^\beta = S_c^\beta + \frac{1}{A}\int_A \epsilon_\beta^\alpha \frac{\partial \Omega}{\partial S^\alpha}\, dA.$$

The strains can, thus, be written in the form

$$
\begin{aligned}
e_{11} &= \overline{e} - f\lambda^{-1}(S^\beta - S_c^\beta)\epsilon_\beta^\alpha(b_\alpha - B_\alpha) + \lambda^{-1}\chi\,\Omega, \\
\gamma_{(\alpha+1)1} &= \overline{\gamma}_\alpha + f\lambda^{-1}\left(\epsilon_\beta^\alpha(S^\beta - S_s^\beta) + \frac{\partial\Omega}{\partial S^\alpha}\right)(b - B) + f^{-1}w_p\left(\epsilon_\beta^\alpha(S_s^\beta - S_c^\beta) - \frac{\partial\Omega}{\partial S^\alpha}\right).
\end{aligned}
$$

The second term in the expression for $\gamma_{(\alpha+1)1}$ is proportional to the shear strain field for pure elastic torsion:

$$
\gamma_\alpha^t(S^\gamma) \stackrel{\text{def}}{=} \epsilon_\beta^\alpha(S^\beta - S_s^\beta) + \frac{\partial\Omega}{\partial S^\alpha}.
$$

We use this definition to eliminate the gradient of Ω from the expression for the shear strain, which yields as final expression for the strains

$$
\begin{aligned}
e_{11} &= \overline{e} - f\lambda^{-1}(S^\beta - S_c^\beta)\epsilon_\beta^\alpha(b_\alpha - B_\alpha) + \lambda^{-1}\chi\Omega, \\
\gamma_{(\alpha+1)1} &= \overline{\gamma}_\alpha + f\lambda^{-1}\gamma_\alpha^t(b - B) + f^{-1}w_p\left(\epsilon_\beta^\alpha(S^\beta - S_c^\beta) - \gamma_\alpha^t\right).
\end{aligned}
$$

Virtual work

Since it was assumed that there are no stresses in the $(\alpha + 1)(\beta + 1)$ directions, the virtual work contribution is

$$
\begin{aligned}
\delta\Pi &= \int_V (\sigma_{11}\delta e_{11} + \tau_{(\alpha+1)1}\delta\gamma_{(\alpha+1)1})dV \\
&= \int_\ell d\ell \int_A (\sigma_{11}\delta e_{11} + \tau_{(\alpha+1)1}\delta\gamma_{(\alpha+1)1})dA.
\end{aligned}
$$

The strain variations are obtained by linearization of the expressions for the strains

$$
\begin{aligned}
\delta e_{11} &= \delta\overline{e} - f\lambda^{-1}(S^\beta - S_c^\beta)\epsilon_\beta^\alpha\delta b_\alpha + \lambda^{-1}\Omega\delta\chi, \\
\delta\gamma_{(\alpha+1)1} &= \delta\overline{\gamma}_\alpha + f\lambda^{-1}\gamma_\alpha^t\delta b + f^{-1}\left(\epsilon_\beta^\alpha(S^\beta - S_c^\beta) - \gamma_\alpha^t\right)\delta w_p,
\end{aligned}
$$

where all terms of the order of the "distortional" strains have been neglected. From the expressions for the average axial strain and average shear strain, we obtain the average axial and shear strain variations as

$$
\begin{aligned}
\delta\overline{e} &= \lambda^{-1}\delta\lambda - \lambda^{-1}fS_c^\beta\epsilon_\beta^\alpha\delta b_\alpha - \lambda^{-1}\Omega_\circ\delta\chi + f^2\lambda^{-2}\left(\frac{I_p}{A} + S_c^\alpha S_c^\alpha\right)b\,\delta b, \\
\delta\overline{\gamma}_\alpha &= \delta\gamma_\alpha + \lambda^{-1}fS_s^\beta\epsilon_\beta^\alpha\delta b + f^{-1}\epsilon_\beta^\alpha(S_c^\beta - S_s^\beta)\delta w_p,
\end{aligned}
$$

where $\delta w_p = f^2\lambda^{-1}\delta b - \delta w$ and again all terms of the order of the "distortional" strains have been neglected.

We now introduce the generalized section forces as defined below:

Axial force	$F = \int_A \sigma_{11} dA$
Shear forces	$F_\alpha = \int_A \tau_{(\alpha+1)1} dA$
Bending moments	$M^\alpha = \int_A -f(S^\beta - S^\beta{}_c)\epsilon^\alpha_\beta \sigma_{11} dA$
Twisting moment	$M_t = \int_A f\gamma^t_\alpha \tau_{(\alpha+1)1} dA$
Warping moment	$M_w = \int_A f\left(\epsilon^\alpha_\beta (S^\beta - S^\beta{}_c) - \gamma^t_\alpha\right) \tau_{(\alpha+1)1} dA$
Bimoment	$W = \int_A \Omega\, \sigma_{11} dA$

which transforms the virtual work contribution into

$$\delta\Pi = \int_\ell (F\delta\overline{e} + F_\alpha \delta\overline{\gamma}_\alpha + M^\alpha \lambda^{-1}\delta b_\alpha + M_t\lambda^{-1}\delta b + M_w f^{-2}\delta w_p + W\lambda^{-1}\delta\chi) d\ell.$$

Observe that the total torque $\boldsymbol{T}$ relative to the centroid of the section is the sum of the twisting moment and the warping moment:

$$T = \int_A f\epsilon^\alpha_\beta (S^\beta - S^\beta_c)\tau_{(\alpha+1)1} dA = M_t + M_w.$$

The rate of change of virtual work

To obtain the rate of change of virtual work, we first transform the integrations in the virtual work equation to the original volume such that

$$\delta\Pi = \int_{\ell^\circ} d\ell^\circ \int_{A^\circ} (\sigma_{11} f^2\lambda\delta e_{11} + \tau_{(\alpha+1)1} f^2\lambda\delta\gamma_{(\alpha+1)1}) dA^\circ.$$

The strain variations relative to the original state are then

$$\begin{aligned} f^2\lambda\delta e_{11} &= f^2\lambda\delta\overline{e} - f^3(S^\beta - S^\beta_c)\epsilon^\alpha_\beta \delta b_\alpha + f^2\Omega\delta\chi, \\ f^2\lambda\delta\gamma_{(\alpha+1)1} &= f^2\lambda\delta\overline{\gamma}_\alpha + f^3\gamma^t_\alpha\delta b + f\lambda\left(\epsilon^\alpha_\beta(S^\beta - S^\beta_c) - \gamma^t_\alpha\right)\delta w_p. \end{aligned}$$

The strain variations relative to the original state are

$$\begin{aligned} f^2\lambda\delta\overline{e} &= f^2\delta\lambda - f^3 S^\beta_c \epsilon^\alpha_\beta \delta b_\alpha - f^2\Omega_\circ\delta\chi + f^4\lambda^{-1}\left(\frac{I_p}{A} + S^\alpha_c S^\alpha_c\right) b\,\delta b, \\ f^2\lambda\delta\overline{\gamma}_\alpha &= f^2\lambda\delta\gamma_\alpha + f^3 S^\beta_s \epsilon^\alpha_\beta \delta b + f\lambda\epsilon^\alpha_\beta(S^\beta_c - S^\beta_s)\delta w_p. \end{aligned}$$

The rate of change of virtual work is then

$$
\begin{aligned}
\mathrm{d}\delta\Pi = \int_{\ell^\circ} d\ell^\circ \int_{A^\circ} \Big[& f^2\lambda(\mathrm{d}\sigma_{11}\delta e_{11} + \mathrm{d}\tau_{(\alpha+1)1}\delta\gamma_{(\alpha+1)1} \\
& + \sigma_{11}(2f\,\mathrm{d}f\,\delta\lambda - 3f^2 S^\beta \epsilon^\alpha_\beta \mathrm{d}f\,\delta b_\alpha + 2f\psi\,\mathrm{d}f\,\delta\chi + f^4\lambda^{-1}\left(\frac{I_p}{A} + S^\alpha_c S^\alpha_c\right)\mathrm{d}b\delta b \\
& + f^2\mathrm{d}\delta\lambda - f^3 S^\beta \epsilon^\alpha_\beta \mathrm{d}\delta b_\alpha + f^2\psi\,\mathrm{d}\delta\chi) \\
& + \tau_{(\alpha+1)1}(f^2\mathrm{d}\lambda\,\delta\gamma_\alpha + 2\lambda f\,\mathrm{d}f\,\delta\gamma_\alpha + 3f^2 S^\beta\epsilon^\alpha_\beta\,\delta b\,\mathrm{d}f + f\frac{\partial\psi}{\partial S^\alpha}\mathrm{d}\lambda\,\delta w \\
& + \lambda\frac{\partial\psi}{\partial S^\alpha}\mathrm{d}f\,\delta w + \lambda f^2\mathrm{d}\delta\gamma_\alpha + f^3 S^\beta\epsilon^\alpha_\beta\mathrm{d}\delta b + \lambda f\frac{\partial\psi}{\partial S^\alpha}\mathrm{d}\delta w)\Big] dA^\circ,
\end{aligned}
$$

where we have neglected terms of order $\boldsymbol{b}$. The changes in stress follow from the constitutive law

$$
\begin{Bmatrix} \mathrm{d}\sigma_{11} \\ \mathrm{d}\tau_{21} \\ \mathrm{d}\tau_{31} \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \begin{Bmatrix} \mathrm{d}e_{11} \\ \mathrm{d}\gamma_{21} \\ \mathrm{d}\gamma_{31} \end{Bmatrix}.
$$

We approximate $\mathrm{d}e_{11}$ and $\mathrm{d}\gamma_{11}$ with the same relations as used for virtual work:

$$
\begin{aligned}
\mathrm{d}e_{11} \quad & = \mathrm{d}\overline{e} - f\lambda^{-1}(S^\beta - S^\beta_c)\epsilon^\alpha_\beta \mathrm{d}b_\alpha + \lambda^{-1}\Omega\mathrm{d}\chi, \\
\mathrm{d}\gamma_{(\alpha+1)1} \quad & = \mathrm{d}\overline{\gamma}_\alpha + f\lambda^{-1}\gamma^t_\alpha \mathrm{d}b + f^{-1}\left(\epsilon^\alpha_\beta(S^\beta - S^\beta_c) - \gamma^t_\alpha\right)\mathrm{d}w_p,
\end{aligned}
$$

and for the average strain rates

$$
\begin{aligned}
\mathrm{d}\overline{e} \quad & = \lambda^{-1}\mathrm{d}\lambda - \lambda^{-1} f S^\beta_c \epsilon^\alpha_\beta \mathrm{d}b_\alpha - \lambda^{-1}\Omega_\circ \mathrm{d}\chi + f^2\lambda^{-2}\left(\frac{I_p}{A} + S^\alpha_c S^\alpha_c\right) b\,\mathrm{d}b, \\
\mathrm{d}\overline{\gamma}_\alpha & = \mathrm{d}\gamma_\alpha + \lambda^{-1} f S^\beta_s \epsilon^\alpha_\beta \mathrm{d}b + f^{-1}\epsilon^\alpha_\beta(S^\beta_c - S^\beta_s)\mathrm{d}w_p.
\end{aligned}
$$

We now neglect all terms that involve the product of a stress tensor; a variation in curvature, twist, or warping; and a change in axial strain of the cross-section. Using the previously obtained expressions for the second variations in $\lambda, b_\alpha, b, \gamma_\alpha$, and χ and transforming the results into the current state, this provides

$$
\begin{aligned}
\mathrm{d}\delta\Pi = \int_\ell d\ell \int_A \Big[& \mathrm{d}\sigma_{11}\delta e_{11} + \mathrm{d}\tau_{(\alpha+1)1}\delta\gamma_{(\alpha+1)1} \\
& + \sigma_{11}(2\lambda^{-1}\delta\lambda\, f^{-1}\mathrm{d}f + \lambda^{-1}\mathrm{d}\delta\lambda - \lambda^{-1} f S^\beta \epsilon^\alpha_\beta \mathrm{d}\delta b_\alpha + f^2\lambda^{-2}\left(\frac{I_p}{A} + S^\alpha_c S^\alpha_c\right)\mathrm{d}b\delta b) \\
& + \tau_{(\alpha+1)1}(\mathrm{d}\delta\gamma_\alpha + \lambda^{-1} f S^\beta \epsilon^\alpha_\beta \mathrm{d}\delta b + f^{-1}\frac{\partial\psi}{\partial S^\alpha}\mathrm{d}\delta w)\Big] dA.
\end{aligned}
$$

The incremental moments, forces, etc. are defined as

$$
\begin{aligned}
\mathrm{d}F &= \int_A \mathrm{d}\sigma_{11} dA, \\
\mathrm{d}F_\alpha &= \int_A \mathrm{d}\tau_{(\alpha+1)1} dA, \\
\mathrm{d}M^\alpha &= \int_A -f(S^\beta - S_c^\beta)\epsilon_\beta^\alpha \mathrm{d}\sigma_{11} dA, \\
\mathrm{d}M_t &= \int_A f\gamma_\alpha^t \mathrm{d}\tau_{(\alpha+1)1} dA, \\
\mathrm{d}M_w &= \int_A f\left(\epsilon_\beta^\alpha (S^\beta - S_c^\beta) - \gamma_\alpha^t\right) \mathrm{d}\tau_{(\alpha+1)1} dA, \\
\mathrm{d}W &= \int_A \Omega\, \mathrm{d}\sigma_{11} dA.
\end{aligned}
$$

For the determination of the initial stress stiffness, we assume that the second variations of the warping function and its derivative vanish: $\mathrm{d}\delta w = \mathrm{d}\delta\chi = 0$. Consequently,

$$
\mathrm{d}\delta w_p = f^2 \lambda^{-1} \mathrm{d}\delta b,
$$

and, therefore, only the torque relative to the centroid plays a role in the initial stress contribution to the rate of change of virtual work:

$$
\begin{aligned}
\mathrm{d}\delta\Pi = \int_\ell (&\mathrm{d}F\delta\overline{e} + \mathrm{d}F_\alpha \delta\overline{\gamma}_\alpha + \mathrm{d}M^\alpha \lambda^{-1}\delta b_\alpha + \mathrm{d}M_t \lambda^{-1}\delta b + \mathrm{d}M_w f^{-2}\delta w_p + \mathrm{d}W\lambda^{-1}\delta\chi \\
&+ F\mathrm{d}\delta\overline{e} + F_\alpha \mathrm{d}\delta\overline{\gamma}_\alpha + M^\alpha \lambda^{-1}\mathrm{d}\delta b_\alpha + T\lambda^{-1}\mathrm{d}\delta b) d\ell.
\end{aligned}
$$

The second variations of $\overline{e}$ and $\overline{\gamma}_\alpha$ contain contributions of the second variation in curvature and twist. These can be separated out by defining the bending moments and the torque relative to the origin of the cross-sectional coordinate system:

$$
\begin{aligned}
\overline{M}^\alpha &= M^\alpha + \int_A f S_c^\beta \epsilon_\beta^\alpha \sigma_{11} dA = M^\alpha + f S_c^\beta \epsilon_\beta^\alpha F, \\
\overline{T} &= T + \int_A f S_c^\beta \epsilon_\beta^\alpha \tau_{(\alpha+1)1} dA = T + f S_c^\beta \epsilon_\beta^\alpha F_\alpha.
\end{aligned}
$$

The expression for the rate of change of virtual work then takes the form

$$
\begin{aligned}
\mathrm{d}\delta\Pi = \int_\ell (&\mathrm{d}F\delta\overline{e} + \mathrm{d}F_\alpha \delta\overline{\gamma}_\alpha + \mathrm{d}M^\alpha \lambda^{-1}\delta b_\alpha + \mathrm{d}M_t \lambda^{-1}\delta b + \mathrm{d}M_w f^{-2}\delta w_p + \mathrm{d}W\lambda^{-1}\delta\chi + \\
&F(\lambda^{-1}\mathrm{d}\delta\lambda + 2\lambda^{-1}\delta\lambda\, f^{-1}\mathrm{d}f + f^2\lambda^{-2}\left(\frac{I_p}{A} + S_c^\alpha S_c^\alpha\right)\mathrm{d}b\delta b) + \\
&F_\alpha \mathrm{d}\delta\gamma_\alpha + \overline{M}^\alpha \lambda^{-1}\mathrm{d}\delta b_\alpha + \overline{T}\lambda^{-1}\mathrm{d}\delta b) d\ell.
\end{aligned}
$$

We propose to make the "cross-section size" a function of the stretch λ. For the thermal stretch of the section we use isotropic expansion ($f_\theta = \lambda_\theta$), and for the "mechanical" stretch of the section we assume an effective Poisson's ratio ν ($f_m = \lambda_m^{-\nu}$). The total cross-sectional stretch is

$$f = f_m f_\theta = \lambda_m^{-\nu} \lambda_\theta = \lambda^{-\nu} \lambda_\theta^{1+\nu},$$

so that

$$\mathrm{d}f = -\nu \lambda^{-1-\nu} \lambda_\theta^{1+\nu} \mathrm{d}\lambda.$$

Using this in the above expression for the rate of change of virtual work, we find

$$\begin{aligned} \mathrm{d}\delta\Pi = \int_\ell (& \mathrm{d}F \delta\overline{e} + \mathrm{d}F_\alpha \delta\overline{\gamma}_\alpha + \mathrm{d}M^\alpha \lambda^{-1} \delta b_\alpha + \mathrm{d}M_t \lambda^{-1} \delta b + \mathrm{d}M_w f^{-2} \delta w_p + \mathrm{d}W \lambda^{-1} \delta\chi + \\ & F(\lambda^{-1} \mathrm{d}\delta\lambda - 2\nu\lambda^{-2} \mathrm{d}\lambda\, \delta\lambda + f^2 \lambda^{-2} \left(\frac{I_p}{A} + S_c^\alpha S_c^\alpha \right) \mathrm{d}b \delta b) + \\ & F_\alpha \mathrm{d}\delta\gamma_\alpha + \overline{M}^\alpha \lambda^{-1} \mathrm{d}\delta b_\alpha + \overline{T} \lambda^{-1} \mathrm{d}\delta b) d\ell. \end{aligned}$$

This expression is symmetric if the material tensor $[\mathbf{D}]$ is symmetric.

Section integration

The formulation presented in the previous pages is valid for all possible beam types. However, different classes of beams will result in different final formulations. We consider three different classes of beams:

- Beams in which warping may be constrained. These beams generally have an open, thin walled section reinforced with some relatively solid parts or some relatively small closed cells and have a torsional constant that is considerably smaller than the polar moment of inertia. Hence, in the elastic range, the warping can be large, and warping prevention at the ends can contribute significantly to the torsional rigidity of the beam. In this case both the torsional shear stresses and the axial warping stresses can be of the same order of magnitude as the stresses due to axial forces and bending moments, and the complete theory must be used.
- Beams in which warping is unconstrained. These beams generally have a solid section or a closed, thin walled section and have a torsional constant that is of the same order of magnitude as the polar moment of inertia of the section. Hence, in the elastic range the warping is rather small, and it is assumed that warping prevention at the ends can be neglected. The axial warping stresses are assumed to be negligible, but the torsional shear stresses are assumed to be of the same order of magnitude as the stresses due to axial forces and bending moments. In this case the warping is dependent on the twist and can be eliminated as an independent variable, which leads to a considerably simplified formulation.
- Beams in which warping constraints dominate the torsional stiffness. These beams generally have an open, thin walled section and have a torsional constant that is much smaller than the polar moment of inertia. In the elastic range the warping is likely to be large, and warping constraints are essential to provide torsional stiffness for the beam. In this case the axial stresses may be of the same order of

magnitude as the stresses due to axial forces and bending moments, but the torsional shear stresses are relatively small. Hence, the warping can be coupled to the twist with a relatively stiff elastic constraint but cannot be eliminated because it must be possible to prevent warping at the nodes.

Examples of cross-sections for the last two classes will be derived after the general discussion of sections with and without warping prevention.

In Abaqus we neglect the effect of shear stresses due to transverse shear forces at individual material points. We will consequently always assume elastic behavior of the section in transverse shear, leading to the relations

$$F_\alpha = f_p GA\,\overline{\gamma}_\alpha, \qquad \mathrm{d}F_\alpha = f_p GA\,\mathrm{d}\overline{\gamma}_\alpha,$$

where F_α are the transverse shear forces, working at the shear center, and f_p is the "slenderness compensation factor" used to prevent the shear stiffness from becoming too big in slender beams. The slenderness compensation factor is defined as

$$f_p = \left(1 + 0.25\frac{\ell^2 A}{12I}\right)^{-1} \quad \text{for first-order Timoshenko beam elements and}$$

$$f_p = \left(1 + 0.25 \times 10^{-4}\frac{\ell^2 A}{12I}\right)^{-1} \quad \text{for all other beam elements,}$$

where ℓ is the length of the element and $\boldsymbol{I}$ is the larger of the moments of inertia I_{11} and I_{22}. Hence, the transverse shear terms do not need to be considered in any further detail.

The fact that the transverse shear forces are considered separately allows us to write

$$\begin{aligned}\gamma_{(\alpha+1)1} &= \gamma^F_{(\alpha+1)1} + \gamma^M_{(\alpha+1)1},\\ \tau_{(\alpha+1)1} &= \tau^F_{(\alpha+1)1} + \tau^M_{(\alpha+1)1},\end{aligned}$$

where $\gamma^F_{(\alpha+1)1}$ and $\tau^F_{(\alpha+1)1}$ are the strains and stresses due to transverse shear forces and $\gamma^M_{(\alpha+1)1}$ and $\tau^M_{(\alpha+1)1}$ are the strains and stresses due to a twisting around the shear center. Substitution in the expressions for the twisting and warping moments yields

$$\begin{aligned}M_t &= \int_A f\gamma^t_\alpha(\tau^F_{(\alpha+1)1} + \tau^M_{(\alpha+1)1})dA = \int_A f\gamma^t_\alpha\tau^M_{(\alpha+1)1}dA,\\ M_w &= \int_A f\left(\epsilon^\alpha_\beta(S^\beta - S^\beta_c) - \gamma^t_\alpha\right)(\tau^F_{(\alpha+1)1} + \tau^M_{(\alpha+1)1})dA\\ &= \epsilon^\alpha_\beta(S^\beta_s - S^\beta_c)F_\alpha + \int_A f\left(\epsilon^\alpha_\beta(S^\beta - S^\beta_c) - \gamma^t_\alpha\right)\tau^M_{(\alpha+1)1}dA,\end{aligned}$$

where we used the fact that γ^t_α was calculated based on application of a twisting moment around the shear center and, hence, does not do any work on the shear stresses due to transverse shear forces.

The warping function Ω is assumed to be determined based on isotropic, homogeneous elastic behavior of the section in shear. For this case the elastic energy due to twist is

$$U_t = \frac{1}{2} G \int_A \gamma^M_{(\alpha+1)1}\, \gamma^M_{(\alpha+1)1} dA.$$

For twist without warping constraints w_p vanishes and the energy per unit length of the beam is

$$U_t = \frac{1}{2} f^2 \lambda^{-2} (b-B)^2 G \int_A \gamma^t_\alpha\, \gamma^t_\alpha dA = \frac{1}{2} f^2 \lambda^{-2} (b-B)^2 GJ,$$

where we have introduced the torsion integral $\boldsymbol{J}$ given by

$$J = \int_A \gamma^t_\alpha\, \gamma^t_\alpha dA.$$

With complete warping prevention the $w_p = f^2 \lambda^{-1} (b-B)$ and the energy per unit length of the beam is

$$U_t = \frac{1}{2} f^2 \lambda^{-2} (b-B)^2 G \int_A \epsilon^\alpha_\beta (S^\beta - S^\beta_c) \epsilon^\alpha_\gamma (S^\gamma - S^\gamma_c) dA = \frac{1}{2} f^2 \lambda^{-2} (b-B)^2 G I_p,$$

where we have introduced the polar moment of inertia

$$I_p = \int_A (S^\beta - S^\beta_c)(S^\beta - S^\beta_c) dA = \int_A \epsilon^\alpha_\beta (S^\beta - S^\beta_c) \epsilon^\alpha_\gamma (S^\gamma - S^\gamma_c) dA.$$

For unconstrained warping $\tau_{(\alpha+1)1} = G \gamma^t_\alpha$. Since the twisting moment must be equal to

$$M_t = \int_A f (S^\beta - S^\beta_c) \epsilon^\alpha_\beta \tau_{(\alpha+1)1} dA = G \int_A f (S^\beta - S^\beta_c) \epsilon^\alpha_\beta \gamma^t_\alpha dA,$$

and since

$$M_t = \int_A f \gamma^t_\alpha \tau_{(\alpha+1)1} dA = G \int_A f \gamma^t_\alpha \gamma^t_\alpha dA,$$

it follows that

$$\int_A (S^\beta - S^\beta_c) \epsilon^\alpha_\beta \gamma^t_\alpha dA = \int_A \gamma^t_\alpha \gamma^t_\alpha dA = J.$$

Beams with unconstrained warping

For this beam type, warping prevention is not taken into consideration. Hence we assume $w_p = 0$. In addition we assume that axial strains due to warping can be neglected: $\chi\, \Omega \approx 0$.

For the strains at a material point this yields

$$\begin{aligned} e_{11} &= \overline{e} - (S^\beta - S^\beta_c) \epsilon^\alpha_\beta (f \lambda^{-1} b_\alpha - B_\alpha), \\ \gamma^M_{(\alpha+1)1} &= f \lambda^{-1} \gamma^t_\alpha (b - B), \end{aligned}$$

and for the variations in the strains

$$\begin{aligned} \delta e_{11} &= \delta\overline{e} - f\lambda^{-1}(S^\beta - S_c^\beta)\epsilon_\beta^\alpha\, \delta b_\alpha, \\ \delta\gamma_{(\alpha+1)1}^M &= f\lambda^{-1}\gamma_\alpha^t \delta b. \end{aligned}$$

Substitution in the virtual work statement yields

$$\delta\Pi = \int_\ell (F\delta\overline{e} + F_\alpha \delta\overline{\gamma}_\alpha + M^\alpha\lambda^{-1}\delta b_\alpha + M_t\lambda^{-1}\delta b)d\ell,$$

where the expressions derived earlier for $\boldsymbol{F}$, F_α, M^α, and M_t apply.

Although there is no warping prevention in the section, the warping moment M_w does not vanish. From the expression obtained earlier follows

$$M_w = \epsilon_\beta^\alpha(S_s^\beta - S_c^\beta)F_\alpha.$$

This yields for the torque around the centroid

$$T = \epsilon_\beta^\alpha(S_s^\beta - S_c^\beta)F_\alpha + M_t$$

and for the torque around the origin

$$\overline{T} = \epsilon_\beta^\alpha S_s^\beta F_\alpha + M_t.$$

For the rate of change of virtual work we obtain

$$\begin{aligned} \mathrm{d}\delta\Pi = \int_\ell (&\mathrm{d}F\delta\overline{e} + \mathrm{d}F_\alpha\delta\overline{\gamma}_\alpha + \mathrm{d}M^\alpha\lambda^{-1}\delta b_\alpha + \mathrm{d}M_t\lambda^{-1}\delta b + \\ &F(\lambda^{-1}\mathrm{d}\delta\lambda - 2\nu\lambda^{-2}\mathrm{d}\lambda\,\delta\lambda + f^2\lambda^{-2}\left(\frac{I_p}{A} + S_c^\alpha S_c^\alpha\right)\mathrm{d}b\delta b) + \\ &F_\alpha\mathrm{d}\delta\gamma_\alpha + \overline{M}^\alpha\lambda^{-1}\mathrm{d}\delta b_\alpha + \overline{T}\lambda^{-1}\mathrm{d}\delta b)d\ell. \end{aligned}$$

Beams with elastic torsion and constrained warping

Consider the case that the shear stresses are defined from the shear strains by linear elastic response, with a constant shear modulus G:

$$\tau_{(\alpha+1)1}^M = G\gamma_{(\alpha+1)1}^M = f\lambda^{-1}G\gamma_\alpha^t(b-B) + f^{-1}G\left(\epsilon_\beta^\alpha(S^\beta - S_c^\beta) - \gamma_\alpha^t\right).$$

This allows us to write for the twisting and warping moments:

$$
\begin{aligned}
M_t &= f^2\lambda^{-1}G(b-B)\int_A \gamma_\alpha^t\gamma_\alpha^t dA + G\,w_p\int_A \gamma_\alpha^t\left(\epsilon_\beta^\alpha(S^\beta - S_c^\beta) - \gamma_\alpha^t\right) dA \\
&= f^2\lambda^{-1}GJ(b-B) \\
M_w &= f^2\lambda^{-1}G(b-B)\int_A \gamma_\alpha^t\left(\epsilon_\beta^\alpha(S^\beta - S_c^\beta) - \gamma_\alpha^t\right) dA \\
&\quad + G\,w_p\int_A \left(\epsilon_\beta^\alpha(S^\beta - S_c^\beta) - \gamma_\alpha^t\right)\left(\epsilon_\gamma^\alpha(S^\gamma - S_c^\gamma) - \gamma_\alpha^t\right) dA \\
&= G(I_p - J)w_p.
\end{aligned}
$$

Substitution of these expressions in the part of the virtual work equation related to torsion yields

$$
\delta\Pi_t = \int_\ell (\lambda^{-1}M_t\delta b + f^{-2}M_w\delta w_p)d\ell = \int_\ell \left(f^2\lambda^{-2}GJ(b-B)\delta b + f^{-2}G(I_p-J)w_p\delta w_p\right) d\ell.
$$

Note that the torque $\boldsymbol{T}$ around the centroid is

$$
T = M_t + M_w = GJf^2\lambda^{-1}(b-B) + G(I_p-J)w_p.
$$

Hence, we can write virtual work in terms of the primary variables $\boldsymbol{b}$ and $\boldsymbol{w}$:

$$
\begin{aligned}
\delta\Pi_t &= \int_\ell \left(\lambda^{-1}T\delta b - f^{-2}M_w\delta w\right) dA \\
&= \int_\ell \left[\left(f^2\lambda^{-2}GJ(b-B) + \lambda^{-1}G(I_p-J)w_p\right)\delta b - f^{-2}G(I_p-J)w_p\delta w\right] d\ell.
\end{aligned}
$$

The complete virtual work equation has the form

$$
\delta\Pi = \int_\ell (F\delta\overline{e} + F_\alpha\delta\overline{\gamma}_\alpha + \lambda^{-1}M^\alpha\delta b_\alpha + f^2\lambda^{-2}GJ\,b\,\delta b + f^{-2}G(I_p-J)w_p\delta w_p + \lambda^{-1}W\delta\chi)d\ell,
$$

where $\boldsymbol{F}$, F_α, M^α and $\boldsymbol{W}$ are defined as before.

For the rate of change of virtual work we obtain similarly

$$
\begin{aligned}
\mathrm{d}\delta\Pi = \int_\ell (&\mathrm{d}F\delta\overline{e} + \mathrm{d}F_\alpha\delta\overline{\gamma}_\alpha + \mathrm{d}M^\alpha\lambda^{-1}\delta b_\alpha + f^2\lambda^{-2}GJ\,\mathrm{d}b\,\delta b + f^{-2}G(I_p-J)\mathrm{d}w_p\delta w_p + \\
&\mathrm{d}W\lambda^{-1}\delta\chi + F(\lambda^{-1}\mathrm{d}\delta\lambda - 2\nu\lambda^{-2}\mathrm{d}\lambda\,\delta\lambda + f^2\lambda^{-2}\left(\frac{I_p}{A} + S_c^\alpha S_c^\alpha\right)\mathrm{d}b\delta b) + \\
&F_\alpha\mathrm{d}\delta\gamma_\alpha + \overline{M}^\alpha\lambda^{-1}\mathrm{d}\delta b_\alpha + \overline{T}\lambda^{-1}\mathrm{d}\delta b)d\ell
\end{aligned}
$$

with

$$
\overline{T} = T + f\epsilon_\beta^\alpha S_c^\beta F_\alpha.
$$

We now discuss some specific section types incorporated in Abaqus.

Circular section

For this type of section, warping is absent. Hence,

$$\Omega = \frac{\partial \Omega}{\partial S^{\alpha}} = 0.$$

Solid noncircular sections

Solid sections such as rectangles or trapezoids are included in this category. The warping function ψ is a harmonic function and is subject to the condition that no shear stress component can act normal to the boundary of the cross-section. Although it is possible to determine the warping function ψ in this manner, we choose to work in terms of the Saint-Venant's stress function because of its simplicity. Following standard procedures we normalize this function so that the (elastic) shear strains can be derived directly from it. We introduce the function $\varphi(S^{\beta})$, which is differentiable in the cross-section and has the property that

$$\gamma_{\alpha}^{t} = \epsilon_{\alpha}^{\beta} \frac{\partial \varphi}{\partial S^{\beta}}.$$

The stress function is determined by solving the differential equation of the form

$$\nabla^2 \varphi = -2 \quad \text{in } A, \text{ with } \varphi = 0 \quad \text{on } S,$$

where S represents the boundary of the section. This boundary condition ensures that no shear stress component can act normal to the boundary.

For the solid noncircular sections this differential equation is solved numerically using a second-order isoparametric finite element. The torsional constant of the bar is then equal to twice the volume under the normalized stress function surface.

Closed, thin walled cross-sections

In this case we assume that the shear strain perpendicular to the section must vanish so that

$$(\epsilon_{\beta}^{\alpha} S^{\beta} + \frac{\partial \Omega}{\partial S^{\alpha}}) n_{\alpha} = 0.$$

Since $t_{\beta} = -\epsilon_{\beta}^{\alpha} n_{\alpha}$, this yields

$$\frac{\partial \Omega}{\partial n} = \frac{\partial \Omega}{\partial S^{\alpha}} n_{\alpha} = t_{\beta} S^{\beta},$$

which can be identically satisfied anywhere. The normalized shear strain along the section is

$$\overline{\gamma} = (\epsilon_{\beta}^{\alpha} S^{\beta} + \frac{\partial \Omega}{\partial S^{\alpha}})\, t_{\alpha} = n_{\beta} S^{\beta} + \frac{\partial \Omega}{\partial s},$$

where ***s*** is the distance along the section. Integrating this around the circumference yields

$$\oint \overline{\gamma} ds = \oint n_\alpha S^\alpha ds + \oint \frac{\partial \Omega}{\partial s} ds = 2A_C,$$

where A_C is the area enclosed by the section. With the assumption that the shear modulus is constant along the section, the total torsional elastic strain energy is

$$U_T = \oint \frac{1}{2} G \overline{\gamma}^2 t ds,$$

where ***t*** is the wall thickness. Minimization of the energy with the constraint enforced with a Lagrange multiplier μ yields

$$\delta P = \oint (G\overline{\gamma} h - \mu)\delta\overline{\gamma} ds - (\oint \overline{\gamma} ds - 2A_C)\delta\mu = 0.$$

Hence, $\overline{\gamma} = \dfrac{\mu}{Gh}$ and $\dfrac{\mu}{G}\oint \dfrac{1}{h} ds - A_C = 0$, which combine to define

$$\overline{\gamma} = \frac{2A_C}{h \oint \frac{1}{h}\, ds}.$$

This allows calculation of $\overline{\gamma}$ at any point in the section based on the section geometry.

Thin walled open sections

The most important sections that exhibit substantial warping are the thin walled open sections. For a single branch section we can conveniently express Ω as a function of the coordinate ***s*** along the section and the coordinate ***z*** perpendicular to the section. A suitable approximation for Ω is

$$\Omega(s, z) = \Omega_c(s) + z\ \eta(s),$$

where $\Omega_c(s)$ is the value at the centerline of the wall. Minimization of the torsional elastic energy yields

$$\int_{s_1}^{s_2} \int_{-h/2}^{h/2} \left[\left(t_\alpha \epsilon^\alpha_\beta (S^\beta - S^\beta_s) + \frac{d\,\Omega_c}{ds} + z\frac{d\eta}{ds} \right) \left(\frac{d\delta\Omega_c}{ds} + z\frac{d\delta\eta}{ds} \right) + (n_\alpha \epsilon^\alpha_\beta S^\beta + \eta)\delta\eta \right] dz\ ds = 0.$$

We now introduce S^β_0, which is the position of the middle of the wall. With $t_\alpha \epsilon^\alpha_\beta = n_\beta$ and $S^\beta = S^\beta_0 + z\ n_\beta$, and after carrying out the integration over ***z***, the minimization condition simplifies to

$$\int_{s_1}^{s_2} \left[h \left(n_\beta (S^\beta_0 - S^\beta_s) + \frac{d\,\Omega_c}{ds} \right) \frac{d\delta\Omega_c}{ds} + h \left(-t_\beta (S^\beta_0 - S^\beta_s) + \eta \right) \delta\eta + \frac{1}{12} h^3 \left(1 + \frac{d\eta}{ds} \right) \frac{d\delta\eta}{ds} \right] ds = 0.$$

Clearly, the dominant terms are of order ***h***. This yields the equations

$$\frac{d\Omega_c}{ds} = -n_\beta(S_0^\beta - S_s^\beta), \qquad \eta = t_\beta(S_0^\beta - S_s^\beta),$$

so that $\Omega(s,z)$ is

$$\Omega(s,z) = \int_0^s -n_\beta(S_0^\beta - S_s^\beta)ds + t_\beta(S_0^\beta - S_s^\beta)z + \Omega_s,$$

where Ω_s is the value of Ω at the start of the integration over the section. Observe that

$$\Omega_c(s) = \int_0^s -n_\beta(S_0^\beta - S_s^\beta)ds + \Omega_s$$

represents the *sectorial area* between two points on the midsection relative to the shear center. Therefore, it readily follows that

$$\int_0^{s_{end}} h\,\Omega_c S^\alpha ds = 0,$$

so there is no coupling between twist and bending in the section for linear elastic material behavior. As was discussed before, Ω_s must be chosen such that

$$\overline{\Omega} = \int_0^{s_{end}} h\,\Omega_c ds \Big/ \int_0^{s_{end}} h\,ds = 0,$$

which eliminates the coupling between twist and axial extension in the section for linear elasticity.

Note that coupling terms still exist but that they are incorporated in the generalized strain displacement relations. The coupling between twist and extension is governed by Ω_0, the value of the warping function at the origin of the cross-sectional coordinate system. If the origin is on the section, this value can be evaluated properly. If the origin is not on the section (which means that the node is not connected to the section), we assume that $\Omega_0 = 0$.

The torsion integral $\boldsymbol{J}$ is readily obtained as

$$\begin{aligned} J &= \int_{s_1}^{s_2}\int_{-h/2}^{h/2}\left[\left(n_\beta(S^\beta - S_s^\beta) + \frac{d\,\Omega}{ds}\right)^2 + \left(-t_\beta(S^\beta - S_s^\beta) + \frac{d\,\Omega}{dz}\right)^2\right]dz\,ds \\ &= \int_{s_1}^{s_2}\int_{-h/2}^{h/2}(2z)^2\,dz\,ds = \int_{s_1}^{s_2}\frac{1}{3}h^3\,ds, \end{aligned}$$

and the polar moment of inertia is given by the expression

$$I_p = \int_{s_1}^{s_2} h(S^\beta - S_c^\beta)(S^\beta - S_c^\beta)\,ds.$$

The above derivations cover single branch open sections. Multibranch open sections can be transformed into single branch open sections by connecting the end of one branch with the beginning of the next

branch with a section that has thickness $h = 0$. Such dummy sections do not yield any contribution to the area, the moments of inertia, or the torsion integral and, hence, have no influence on the results.

Reference

- "Beam modeling: overview," Section 29.3.1 of the Abaqus Analysis User's Guide

3.5.3 EULER-BERNOULLI BEAM ELEMENTS

Product: Abaqus/Standard

In these elements it is assumed that the internal virtual work rate is associated with axial strain and torsional shear only. Further, it is assumed that the cross-section does not deform in its plane or warp out of its plane, and that this cross-sectional plane remains normal to the beam axis. These are the classical assumptions of the Euler-Bernoulli beam theory, which provides satisfactory results for slender beams.

Let (S, g, h) be material coordinates such that S locates points on the beam axis and (g, h) measures distance in the cross-section. In addition, let $\mathbf{n}_1$, $\mathbf{n}_2$ be unit vectors normal to the beam axis in the current configuration: $\mathbf{n}_1 = \mathbf{n}_1(S)$, $\mathbf{n}_2 = \mathbf{n}_2(S)$. Then the position of a point of the beam in the current configuration is

$$\mathbf{x}^f = \mathbf{x} + g\mathbf{n}_1 + h\mathbf{n}_2,$$

where $\mathbf{x} = \mathbf{x}(S)$ is the point on the beam axis of the cross-section containing $\mathbf{x}^f$. Then

$$\frac{d\mathbf{x}^f}{dS} = \frac{d\mathbf{x}}{dS} + g\frac{d\mathbf{n}_1}{dS} + h\frac{d\mathbf{n}_2}{dS},$$

and so length on the fiber at (S, g, h) is measured in the current configuration as

$$\begin{aligned}(dl^f)^2 &= \frac{d\mathbf{x}^f}{dS}\cdot\frac{d\mathbf{x}^f}{dS}(dS)^2 \\ &= \left(\frac{d\mathbf{x}}{dS} + g\frac{d\mathbf{n}_1}{dS} + h\frac{d\mathbf{n}_2}{dS}\right)\cdot\left(\frac{d\mathbf{x}}{dS} + g\frac{d\mathbf{n}_1}{dS} + h\frac{d\mathbf{n}_2}{dS}\right)(dS)^2.\end{aligned}$$

Now since the beam is slender, we will neglect terms of second-order in g and h, the distance measuring material coordinates in the cross-section. Thus,

$$(dl^f)^2 = \left(\frac{d\mathbf{x}}{dS}\cdot\frac{d\mathbf{x}}{dS} + 2g\frac{d\mathbf{x}}{dS}\cdot\frac{d\mathbf{n}_1}{dS} + 2h\frac{d\mathbf{x}}{dS}\cdot\frac{d\mathbf{n}_2}{dS}\right)(dS)^2. \tag{3.5.3–1}$$

Strain measures

The internal virtual work rate associated with axial stress is

$$\delta W_1^I = \int_{L^f}\int_A \sigma^f\,\delta\varepsilon^f\,dA\,dL^f,$$

where σ^f and $\delta\varepsilon^f$ are any material stress and strain measures associated with axial deformation at the point (S, g, h) of the beam, since strains are assumed to be small. For this purpose we will use Green's strain so that

$$\varepsilon^f = \frac{1}{2}[(\lambda^f)^2 - 1],$$

where $(\lambda^f)^2 = (dl^f)^2/(dL^f)^2$, the square of the ratio of current configuration length to reference configuration length in the axial direction on the fiber. From Equation 3.5.3–1 and its equivalent in the reference configuration, we have

$$\begin{aligned}\varepsilon^f dL^f = \frac{1}{2}\Bigg[&\left(\frac{d\mathbf{x}}{dS}\cdot\frac{d\mathbf{x}}{dS} + 2g\frac{d\mathbf{x}}{dS}\cdot\frac{d\mathbf{n}_1}{dS} + 2h\frac{d\mathbf{x}}{dS}\cdot\frac{d\mathbf{n}_2}{dS}\right)\\ &\cdot\left(\frac{d\mathbf{x}}{dS}\cdot\frac{d\mathbf{x}}{dS} + 2g\frac{d\mathbf{X}}{dS}\cdot\frac{d\mathbf{N}_1}{dS} + 2h\frac{d\mathbf{x}}{dS}\cdot\frac{d\mathbf{N}_2}{dS}\right)^{-1} - 1\Bigg]\\ &\cdot\left(\frac{d\mathbf{X}}{dS}\cdot\frac{d\mathbf{X}}{dS} + 2g\frac{d\mathbf{X}}{dS}\cdot\frac{d\mathbf{N}_1}{dS} + 2h\frac{d\mathbf{X}}{dS}\cdot\frac{d\mathbf{N}_2}{dS}\right)^{\frac{1}{2}} dS.\end{aligned}$$

Again, neglecting all but first-order terms in g and h because of the slenderness assumption, this becomes

$$\begin{aligned}\varepsilon^f dL^f \approx \Bigg[&\varepsilon + gG^{-1}\left(\frac{d\mathbf{x}}{dS}\cdot\frac{d\mathbf{n}_1}{dS} - (1+\varepsilon)\frac{d\mathbf{X}}{dS}\cdot\frac{d\mathbf{N}_1}{dS}\right)\\ &+ hG^{-1}\left(\frac{d\mathbf{x}}{dS}\cdot\frac{d\mathbf{n}_2}{dS} - (1+\varepsilon)\frac{d\mathbf{X}}{dS}\cdot\frac{d\mathbf{N}_2}{dS}\right)\Bigg]dL,\end{aligned}$$

where

$G = \dfrac{d\mathbf{X}}{dS}\cdot\dfrac{d\mathbf{X}}{dS}$ and $\varepsilon = \dfrac{1}{2}(\lambda^2 - 1) = \dfrac{1}{2}\left[\left(\dfrac{dl}{dL}\right)^2 - 1\right]$ is the Green's strain of the beam axis.

This simplification allows us to write the internal virtual work rate associated with axial stress as

$$\delta W_1^I = \int_{L^f}\int_A \sigma^f \delta\varepsilon^f \, dA \, dL^f = \int_L \int_A \sigma^f \delta\tilde{\varepsilon}^f \, dA \, dL,$$

where

$$\tilde{\varepsilon}^f = \varepsilon - gK_2 + hK_1,$$

with

$$\begin{aligned}\varepsilon &= \frac{1}{2}(\lambda^2 - 1) \text{ as Green's strain of the beam axis,}\\ K_2 &= -G^{-1}\left(\frac{d\mathbf{x}}{dS}\cdot\frac{d\mathbf{n}_1}{dS} - (1+\varepsilon)\frac{d\mathbf{X}}{dS}\cdot\frac{d\mathbf{N}_1}{dS}\right)\end{aligned}$$

and

$$K_1 = -G^{-1}\left(\frac{d\mathbf{x}}{dS}\cdot\frac{d\mathbf{n}_2}{dS} - (1+\varepsilon)\frac{d\mathbf{X}}{dS}\cdot\frac{d\mathbf{N}_2}{dS}\right).$$

Now the cross-sectional base vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ are assumed to remain normal to the beam axis, so

$$\frac{d\mathbf{x}}{dS}\cdot\mathbf{n}_1 = \frac{d\mathbf{x}}{dS}\cdot\mathbf{n}_2 = 0.$$

Hence,

$$\frac{d\mathbf{x}}{dS}\cdot\frac{d\mathbf{n}_1}{dS} = -\frac{d^2\mathbf{x}}{dS^2}\cdot\mathbf{n}_1, \text{ and } \frac{d\mathbf{x}}{dS}\cdot\frac{d\mathbf{n}_2}{dS} = -\frac{d^2\mathbf{x}}{dS^2}\cdot\mathbf{n}_2.$$

So we have

$$K_2 = G^{-1}\left(\mathbf{n}_1\cdot\frac{d^2\mathbf{x}}{dS^2} - (1+\varepsilon)\mathbf{N}_1\cdot\frac{d^2\mathbf{X}}{dS^2}\right)$$

and

$$K_1 = -G^{-1}\left(\mathbf{n}_2\cdot\frac{d^2\mathbf{x}}{dS^2} - (1+\varepsilon)\mathbf{N}_2\cdot\frac{d^2\mathbf{X}}{dS^2}\right).$$

This defines the generalized strains associated with axial stretch. For torsional strain the internal virtual work rate is

$$\delta W_2^I = \int_L\int_A \gamma_\alpha^t \tau_\alpha^f \delta e_1\, dA dL,$$

where

$$e_1 = \mathbf{n}_2\cdot\frac{d\mathbf{n}_1}{dS} - \mathbf{N}_2\cdot\frac{d\mathbf{N}_1}{dS}$$

and γ_α^t is the proportionality constant between shear strains and torsional strains (see "Beam element formulation," Section 3.5.2, for details). For computational simplicity the form of the torsional strains is taken to be

$$e_1 = \mathbf{n}_2\cdot\frac{d\tilde{\mathbf{n}}_1}{dS} - \mathbf{N}_2\cdot\frac{d\mathbf{N}_1}{dS},$$

where

$$\tilde{\mathbf{n}}_1 = \mathbf{C}(p_1\boldsymbol{\omega}^1 + p_2\boldsymbol{\omega}^2)\cdot\mathbf{N}_1,$$

and

$$p_1 = \frac{1}{2}(1-g),$$
$$p_2 = \frac{1}{2}(1+g),$$
$$-1 \le g \le 1.$$

This assumes a linear interpolation of rotation $\boldsymbol{\omega}$ along the beam. Thus, the generalized strain measures for these beams are

ε	axial strain,
K_1 and K_2	the beam curvature change measures, and
e_1	the torsional strain.

With these measures, the internal virtual work rate can be written

$$\delta W^I = \int_L \left[\delta\varepsilon \int_A \sigma^f \, dA - \delta K_2 \int_A g\sigma^f \, dA + \delta K_1 \int_A h\sigma^f \, dA + \delta e_1 \int_A \gamma_\alpha^t \tau_\alpha^f dA \right] dL.$$

Internal virtual work rate Jacobian

For the Jacobian matrix of the overall Newton method, Equation 2.1.1–3, the variation of this internal work rate with respect to nodal displacement variations must be formed. The constitutive theory is written as

$$\left\{ \begin{array}{c} d\sigma^f \\ d\tau^f \end{array} \right\} = [\, H \,] \left\{ \begin{array}{c} d\varepsilon^f \\ d\gamma^f \end{array} \right\},$$

and so

$$d\delta W^I = \int_L \left[\left\lfloor \delta\varepsilon \; \delta K_1 \; \delta K_2 \; \delta e_1 \right\rfloor [\, B \,] \left\{ \begin{array}{c} d\varepsilon \\ dK_1 \\ dK_2 \\ de_1 \end{array} \right\} + d\delta\varepsilon \int_A \sigma^f dA + d\delta K_1 \int_A h\sigma^f dA \right.$$

$$\left. - d\delta K_2 \int_A g\sigma^f dA + d\delta e_1 \int_A \gamma_\alpha^f \tau_\alpha^f dA \right] dL,$$

where $[\, B \,]$ is the section stiffness matrix obtained by integration over the cross-section. See "Beam element formulation," Section 3.5.2, for more information on section integration.

First variations of strains

Taking the variations of the above strain definitions gives directly

$$\delta\varepsilon = G^{-1}\frac{d\mathbf{x}}{dS}\cdot\frac{d\delta\mathbf{u}}{dS},$$
$$\delta K_2 = G^{-1}\left(\delta\mathbf{n}_1\cdot\frac{d^2\mathbf{x}}{dS^2}+\mathbf{n}_1\cdot\frac{d^2\delta\mathbf{u}}{dS^2}-\delta\varepsilon\mathbf{N}_1\cdot\frac{d^2\mathbf{X}}{dS^2}\right)$$
$$\delta K_1 = G^{-1}\left(\delta\mathbf{n}_2\cdot\frac{d^2\mathbf{x}}{dS^2}+\mathbf{n}_2\cdot\frac{d^2\delta\mathbf{u}}{dS^2}-\delta\varepsilon\mathbf{N}_2\cdot\frac{d^2\mathbf{X}}{dS^2}\right)$$

and

$$\delta e_1 = \delta\mathbf{n}_2\cdot\frac{d\tilde{\mathbf{n}}_1}{dS}+\mathbf{n}_2\cdot\frac{d\delta\tilde{\mathbf{n}}_1}{dS}.$$

In these expressions we need $\delta\mathbf{n}_1$ and $\delta\mathbf{n}_2$ as well as $\delta\tilde{\mathbf{n}}_1$; these are now derived. From the expression for $\tilde{\mathbf{n}}_1$, namely

$$\tilde{\mathbf{n}}_1 = \mathbf{C}(p_1\boldsymbol{\omega}^1+p_2\boldsymbol{\omega}^2)\cdot\mathbf{N}_1,$$

another ancillary vector $\overline{\mathbf{n}}_1$, normal to the tangent, is defined by

$$\overline{\mathbf{n}}_1 = \tilde{\mathbf{n}}_1 - \tilde{\mathbf{n}}_1\cdot\mathbf{t}\mathbf{t}$$

so that

$$\mathbf{n}_1 = \overline{\mathbf{n}}_1/|\overline{\mathbf{n}}_1|.$$

In addition,

$$\mathbf{t} = g^{-\frac{1}{2}}\frac{d\mathbf{x}}{dS}$$

so that

$$\delta\mathbf{t} = g^{-\frac{1}{2}}(\mathbf{I}-\mathbf{t}\mathbf{t})\cdot\frac{d\delta\mathbf{u}}{dS}.$$

Since $(\mathbf{t}_1\ \mathbf{n}_1\ \mathbf{n}_2)$ form an orthonormal triad,

$$\begin{aligned}\delta\mathbf{n}_1 &= \delta\mathbf{n}_1\cdot(\mathbf{n}_2\mathbf{n}_2+\mathbf{t}\mathbf{t})\\ &= \delta\mathbf{n}_1\cdot\mathbf{n}_2\mathbf{n}_2+\delta\mathbf{n}_1\cdot\mathbf{t}\mathbf{t}\\ &= \delta\mathbf{n}_1\cdot\mathbf{n}_2\mathbf{n}_2-g^{-\frac{1}{2}}\mathbf{n}_1\cdot\frac{d\delta\mathbf{u}}{dS}\mathbf{t},\end{aligned}\qquad(3.5.3\text{–}2)$$

because $\mathbf{n}_1\cdot\mathbf{t}=0$. From the definition of $\mathbf{n}_1$, it is straightforward to show that

$$\delta\mathbf{n}_1 = \frac{1}{|\overline{\mathbf{n}}_1|}(\mathbf{I}-\mathbf{n}_1\mathbf{n}_1)\cdot\delta\overline{\mathbf{n}}.$$

So

$$\begin{aligned}\delta\mathbf{n}_1\cdot\mathbf{n}_2 &= \frac{1}{|\overline{\mathbf{n}}_1|}\mathbf{n}_2\cdot\delta\mathbf{n}_1\\ &= \frac{1}{|\overline{\mathbf{n}}_1|}\mathbf{n}_2\cdot[\delta\tilde{\mathbf{n}}_1-(\delta\tilde{\mathbf{n}}_1\cdot\mathbf{t})\mathbf{t}-(\tilde{\mathbf{n}}_1\cdot\mathbf{t})\delta\mathbf{t}]\\ &= \frac{1}{|\overline{\mathbf{n}}_1|}\left[\mathbf{n}_2\cdot\delta\tilde{\mathbf{n}}_1-g^{-\frac{1}{2}}(\tilde{\mathbf{n}}_1\cdot\mathbf{t})\mathbf{n}_2\cdot\frac{d\delta\mathbf{u}}{dS}\right]\end{aligned}$$

and

$$\delta\tilde{\mathbf{n}}_1=\delta\boldsymbol{\omega}\times\tilde{\mathbf{n}}_1 \text{ with } \quad \boldsymbol{\omega} \quad \text{interpolated linearly.}$$

Thus,

$$\delta\mathbf{n}_1\cdot\mathbf{n}_2=\frac{1}{|\overline{\mathbf{n}}_1|}\left(\delta\boldsymbol{\omega}\cdot\tilde{\mathbf{n}}_1\times\mathbf{n}_2-g^{-\frac{1}{2}}\tilde{\mathbf{n}}_1\cdot\mathbf{t}\mathbf{n}_2\cdot\frac{d\delta\mathbf{u}}{dS}\right).$$

We can also write

$$\tilde{\mathbf{n}}_1\times\mathbf{n}_2=\tilde{\mathbf{n}}_1\mathbf{t}-\tilde{\mathbf{n}}_1\cdot\mathbf{t}\,\mathbf{n}_1$$

and

$$|\overline{\mathbf{n}}_1|=\overline{\mathbf{n}}_1\cdot\mathbf{n}_1=\tilde{\mathbf{n}}_1\cdot\mathbf{n}_1.$$

Combining terms appropriately,

$$\delta\mathbf{n}_1\cdot\mathbf{n}_2=\delta\boldsymbol{\omega}\cdot\mathbf{t}-\frac{\tilde{\mathbf{n}}_1\cdot\mathbf{t}}{\tilde{\mathbf{n}}_1\cdot\mathbf{n}_1}\left[\delta\boldsymbol{\omega}\cdot\mathbf{n}_1+g^{-\frac{1}{2}}\mathbf{n}_2\cdot\frac{d\delta\mathbf{u}}{dS}\right].$$

Hence, from Equation 3.5.3–2

$$\delta\mathbf{n}_1=\delta\boldsymbol{\omega}\cdot\mathbf{t}\mathbf{n}_2-\frac{\tilde{\mathbf{n}}_1\cdot\mathbf{t}}{\tilde{\mathbf{n}}_1\cdot\mathbf{n}_1}\left[\delta\boldsymbol{\omega}\cdot\mathbf{n}_1+g^{-\frac{1}{2}}\mathbf{n}_2\cdot\frac{d\delta\mathbf{u}}{dS}\right]\mathbf{n}_2-g^{-\frac{1}{2}}\mathbf{n}_1\cdot\frac{d\delta\mathbf{u}}{dS}\mathbf{t}.$$

In a similar manner one can show that

$$\delta\mathbf{n}_2=-\delta\boldsymbol{\omega}\cdot\mathbf{t}\mathbf{n}_1-\frac{\tilde{\mathbf{n}}_1\cdot\mathbf{t}}{\tilde{\mathbf{n}}_1\cdot\mathbf{n}_1}\left[\delta\boldsymbol{\omega}\cdot\mathbf{n}_1+g^{-\frac{1}{2}}\mathbf{n}_1\cdot\frac{d\delta\mathbf{u}}{dS}\right]\mathbf{n}_1-g^{-\frac{1}{2}}\mathbf{n}_2\cdot\frac{d\delta\mathbf{u}}{dS}\mathbf{t}.$$

The first variations of strain become

$$
\begin{aligned}
\delta\varepsilon &= G^{-1}\frac{d\mathbf{x}}{dS}\cdot\frac{d\delta\mathbf{u}}{dS}\\
\delta K_2 &= G^{-1}\left[\mathbf{n}_2\cdot\frac{d^2\mathbf{x}}{d\,S^2}\left(\mathbf{t}-\frac{\tilde{\mathbf{n}}_1\cdot\mathbf{t}}{\tilde{\mathbf{n}}_1\cdot\mathbf{n}_1}\mathbf{n}_1\right)\cdot\delta\boldsymbol{\omega}\right.\\
&\quad - g^{-\frac{1}{2}}\left(\mathbf{t}\cdot\frac{d^2\mathbf{x}}{d\,S^2}\mathbf{n}_1+\frac{\tilde{\mathbf{n}}_1\cdot\mathbf{t}}{\tilde{\mathbf{n}}_1\cdot\mathbf{n}_1}\mathbf{n}_2\cdot\frac{d^2\mathbf{x}}{d\,S^2}\right)\mathbf{n}_2\cdot\frac{d\delta\mathbf{u}}{dS}\\
&\quad \left.+\,\mathbf{n}_1\frac{d^2\delta\mathbf{u}}{dS^2}-\delta\varepsilon\mathbf{N}_1\cdot\frac{d^2\mathbf{X}}{dS^2}\right]\\
\delta K_1 &= G^{-1}\left[\mathbf{n}_1\cdot\frac{d^2\mathbf{x}}{d\,S^2}\left(\mathbf{t}-\frac{\tilde{\mathbf{n}}_1\cdot\mathbf{t}}{\tilde{\mathbf{n}}_1\cdot\mathbf{n}_1}\mathbf{n}_1\right)\cdot\delta\boldsymbol{\omega}\right.\\
&\quad + g^{-\frac{1}{2}}\left(\mathbf{t}\cdot\frac{d^2\mathbf{x}}{d\,S^2}\mathbf{n}_1+\frac{\tilde{\mathbf{n}}_1\cdot\mathbf{t}}{\tilde{\mathbf{n}}_1\cdot\mathbf{n}_1}\mathbf{n}_1\cdot\frac{d^2\mathbf{x}}{d\,S^2}\right)\mathbf{n}_2\cdot\frac{d\delta\mathbf{u}}{dS}\\
&\quad \left.-\,\mathbf{n}_2\frac{d^2\delta\mathbf{u}}{dS^2}-\delta\varepsilon\mathbf{N}_2\cdot\frac{d^2\mathbf{X}}{dS^2}\right]
\end{aligned}
$$

and

$$
\begin{aligned}
\delta e_1 &= \left(\frac{d\tilde{\mathbf{n}}_1}{dS}\cdot\mathbf{n}_1\frac{\tilde{\mathbf{n}}_1\cdot\mathbf{t}}{\tilde{\mathbf{n}}_1\cdot\mathbf{n}_1}-\frac{d\tilde{\mathbf{n}}_1}{dS}\mathbf{t}\right)\mathbf{n}_1\cdot\delta\boldsymbol{\omega}\\
&\quad + g^{-\frac{1}{2}}\left[\frac{\tilde{\mathbf{n}}_1\cdot\mathbf{t}}{\tilde{\mathbf{n}}_1\cdot\mathbf{n}_1}\frac{d\tilde{\mathbf{n}}_1}{dS}\cdot\mathbf{n}_1-\frac{d\tilde{\mathbf{n}}_1}{dS}\cdot\mathbf{t}\right]\mathbf{n}_2\cdot\frac{d\delta\mathbf{u}}{dS}\\
&\quad + \tilde{\mathbf{n}}_1\times\mathbf{n}_2\cdot\frac{d\delta\boldsymbol{\omega}}{dS}
\end{aligned}
$$

so that

$$\mathbf{n}_1=\overline{\mathbf{n}}_1/|\overline{\mathbf{n}}_1|.$$

In addition,

$$\mathbf{t}=g^{-\frac{1}{2}}\frac{d\mathbf{x}}{dS}$$

so that

$$\delta\mathbf{t}=g^{-\frac{1}{2}}(\mathbf{I}-\mathbf{t}\mathbf{t})\cdot\frac{d\delta\mathbf{u}}{dS},$$

since $(\mathbf{t}_1\ \mathbf{n}_1\ \mathbf{n}_2)$ form an orthonormal triad,

$$
\begin{aligned}
\delta\mathbf{n}_1 &= \delta\mathbf{n}_1\cdot(\mathbf{n}_2\mathbf{n}_2+\mathbf{t}\mathbf{t})\\
&= \delta\mathbf{n}_1\cdot\mathbf{n}_2\mathbf{n}_2+\delta\mathbf{n}_1\cdot\mathbf{t}\mathbf{t}\\
&= \delta\mathbf{n}_1\cdot\mathbf{n}_2\mathbf{n}_2-g^{-\frac{1}{2}}\mathbf{n}_1\cdot\frac{d\delta\mathbf{u}}{dS}\mathbf{t},
\end{aligned}
$$

because

$$\mathbf{n}_1 \cdot \mathbf{t} = 0.$$

Second variations of strains

The second variation of the axial strain is simply

$$d\delta\varepsilon = G^{-1} \frac{d\delta\mathbf{u}}{dS} \cdot \frac{dd\mathbf{u}}{dS}.$$

To compute the second variations of bending strain, we need expressions for $d\delta\mathbf{n}_1,\ d\delta\mathbf{n}_2$. These are obtained by approximating

$$\begin{aligned}
\delta\tilde{\mathbf{n}}_1 &\approx \mathbf{n}_2\mathbf{t} \cdot \delta\boldsymbol{\omega} - \mathbf{t}\,\mathbf{n}_1 \cdot \frac{d\delta\mathbf{u}}{dS} \\
\delta\tilde{\mathbf{n}}_2 &\approx -\mathbf{n}_1\mathbf{t} \cdot \delta\boldsymbol{\omega} - \mathbf{t}\,\mathbf{n}_1 \cdot \frac{d\delta\mathbf{u}}{dS} \\
\delta\tilde{\mathbf{t}} &\approx \delta\boldsymbol{\omega} \cdot \mathbf{n}_2 - \delta\boldsymbol{\omega} \cdot \mathbf{n}_1\,\mathbf{n}_2
\end{aligned}$$

from which

$$d\delta\tilde{\mathbf{n}}_1 = d\mathbf{n}_2\mathbf{t} \cdot \delta\boldsymbol{\omega} + \mathbf{n}_2 d\mathbf{t} \cdot \delta\boldsymbol{\omega} - d\mathbf{t}\,\mathbf{n}_1 \cdot \frac{d\delta\mathbf{u}}{dS} - \mathbf{t}\,d\mathbf{n}_1 \cdot \frac{d\delta\mathbf{u}}{dS}.$$

Using these expressions, the second variations of bending strains are written as

$$\begin{aligned}
d\delta K_1 =& G^{-1}\Bigg[\mathbf{n}_2 \cdot \frac{d^2\mathbf{x}}{dS^2}\mathbf{t} \cdot d\boldsymbol{\omega}\,\mathbf{t} \cdot \delta\boldsymbol{\omega} \\
&- \mathbf{t} \cdot \frac{d^2\mathbf{x}}{dS^2}\left(\mathbf{n}_1 \cdot \frac{dd\mathbf{u}}{dS}\,\mathbf{t} \cdot \delta\boldsymbol{\omega} + \mathbf{n}_1 \cdot \frac{d\delta\mathbf{u}}{dS}\,\mathbf{t} \cdot d\boldsymbol{\omega}\right) \\
&+ \mathbf{n}_1 \cdot \frac{d^2 d\mathbf{u}}{dS^2}\,\mathbf{t} \cdot \delta\boldsymbol{\omega} + \mathbf{t} \cdot \frac{d^2 d\mathbf{u}}{dS^2}\,\mathbf{n}_2 \cdot \frac{d\delta\mathbf{u}}{dS} \\
&+ \mathbf{n}_1 \cdot \frac{d^2\delta\mathbf{u}}{dS^2}\,\mathbf{t} \cdot d\boldsymbol{\omega} + \mathbf{t} \cdot \frac{d^2\delta\mathbf{u}}{dS^2}\,\mathbf{n}_2 \cdot \frac{dd\mathbf{u}}{dS} \\
&+ d\delta\varepsilon\mathbf{N}_2 \cdot \frac{d^2\mathbf{X}}{dS^2} \\
&+ \mathbf{n}_1 \cdot \frac{d^2\mathbf{x}}{dS^2}\left(\mathbf{n}_2 \cdot d\boldsymbol{\omega}\,\mathbf{n}_1 \cdot \delta\boldsymbol{\omega} - \mathbf{n}_1 \cdot d\boldsymbol{\omega}\mathbf{n}_2 \cdot \delta\boldsymbol{\omega} + \mathbf{n}_2 \cdot \frac{d\delta\mathbf{u}}{dS}\,\mathbf{n}_2 \cdot d\boldsymbol{\omega}\right) \\
&- \mathbf{n}_2 \cdot \frac{d^2\mathbf{x}}{dS^2}\left(\mathbf{n}_2 \cdot \frac{d\delta\mathbf{u}}{dS}\,\mathbf{n}_1 \cdot d\boldsymbol{\omega}\right) \\
&- \mathbf{t} \cdot \frac{d^2\mathbf{x}}{dS^2}\left(\mathbf{n}_2 \cdot \frac{dd\mathbf{u}}{dS}\,\mathbf{n}_2\,\mathbf{t} \cdot \frac{d\delta\mathbf{u}}{dS}\right)\Bigg]
\end{aligned}$$

and

$$
\begin{aligned}
d\delta K_2 = G^{-1}\Bigg[& -\mathbf{n}_1 \cdot \frac{d^2\mathbf{x}}{dS^2}\mathbf{t}\cdot\delta\boldsymbol{\omega}\,\mathbf{t}\cdot d\boldsymbol{\omega} \\
& - \mathbf{t}\cdot\frac{d^2\mathbf{x}}{dS^2}\left(\mathbf{n}_2\cdot\frac{d\delta\mathbf{u}}{dS}\,\mathbf{t}\cdot d\boldsymbol{\omega} + \mathbf{n}_2\cdot\frac{d d\mathbf{u}}{dS}\,\mathbf{t}\cdot\delta\boldsymbol{\omega}\right) \\
& + \mathbf{n}_2\cdot\frac{d^2 d\mathbf{u}}{dS^2}\,\mathbf{t}\cdot\delta\boldsymbol{\omega} - \mathbf{t}\cdot\frac{d^2 d\mathbf{u}}{dS^2}\,\mathbf{n}_1\cdot\frac{d\delta\mathbf{u}}{dS} \\
& + \mathbf{n}_2\cdot\frac{d^2\delta\mathbf{u}}{dS^2}\,\mathbf{t}\cdot d\boldsymbol{\omega} - \mathbf{t}\cdot\frac{d^2\delta\mathbf{u}}{dS^2}\,\mathbf{n}_1\cdot\frac{d d\mathbf{u}}{dS} \\
& - d\delta\varepsilon\mathbf{N}_1\cdot\frac{d^2\mathbf{X}}{dS^2} \\
& + \mathbf{n}_2\cdot\frac{d^2\mathbf{x}}{dS^2}\left(\mathbf{n}_2\cdot d\boldsymbol{\omega}\,\mathbf{n}_1\cdot\delta\boldsymbol{\omega} - \mathbf{n}_1\cdot d\boldsymbol{\omega}\,\mathbf{n}_2\cdot\delta\boldsymbol{\omega} + \mathbf{n}_1\cdot\frac{d\delta\mathbf{u}}{dS}\,\mathbf{n}_1\cdot d\boldsymbol{\omega}\right) \\
& - \mathbf{n}_1\cdot\frac{d^2\mathbf{x}}{dS^2}\left(\mathbf{n}_1\cdot\frac{d\delta\mathbf{u}}{dS}\,\mathbf{n}_2\cdot d\boldsymbol{\omega}\right) \\
& - \mathbf{t}\cdot\frac{d^2\mathbf{x}}{dS^2}\left(\mathbf{n}_1\cdot\frac{d d\mathbf{u}}{dS}\,\mathbf{n}_1\,\mathbf{t}\cdot\frac{d\delta\mathbf{u}}{dS}\right)\Bigg].
\end{aligned}
$$

For the torsional strain contribution to the initial stress matrix, we approximate

$$
\begin{aligned}
\delta e_2 &\approx \mathbf{t}\cdot\frac{d\delta\boldsymbol{\omega}}{dS} \\
d\delta e_1 &\approx d\mathbf{t}\cdot\frac{d\delta\boldsymbol{\omega}}{dS} = d\boldsymbol{\omega}\times\mathbf{t}\cdot\frac{d\delta\boldsymbol{\omega}}{dS}.
\end{aligned}
$$

This matrix is again unsymmetric.

Interpolation

In the virtual work equation the strains include second derivatives of displacement. For this reason continuity of rotation as well as displacement is needed so that the Hermitian polynomial interpolation functions are the minimum interpolation order needed. These are used here. The Hermite cubic is written in terms of the function value and its derivative at the ends of the interval:

$$
u(g) = N_1(g)u^1 + N_2(g)u^2 + N_3(g)\frac{du^1}{dg} + N_4(g)\frac{du^2}{dg}\ \ , -1 < g < 1,
$$

with node 1 at $g = -1$ and node 2 at $g = +1$.

These functions are used in Abaqus to interpolate the components of displacement $\mathbf{u}$ and the initial position vector $\mathbf{x}$, so that the elements are basically isoparametric. In addition, rotation of $(\mathbf{n}_1\ \mathbf{n}_2)$ about the beam axis, ϕ_a, is interpolated linearly. This interpolation is unsatisfactory for the user, because the nodal variables are

$$(u_x\ u_y\ u_z\ \frac{du_x}{dS}\ \frac{du_y}{dS}\ \frac{du_z}{dS}\ \phi_a).$$

The last four of these variables are difficult to work with; furthermore, making them the same in all elements sharing the same node causes excessive constraint of axial stretch in these elements, especially if the beam axis is not continuous through the node, as in a frame structure or "T" junction.

To avoid this difficulty, the following procedure is adopted. At a node the tangent to the beam axis is

$$\mathbf{t} = \frac{d\mathbf{x}}{dl} = \frac{d\mathbf{x}}{dS}\frac{dS}{dl},$$

so

$$\frac{d\mathbf{x}}{dS} = \frac{dl}{dS}\mathbf{t}.$$

Now suppose we store as degrees of freedom at the node,

$$(u_x,\ u_y, u_z,\ \omega_x,\ \omega_y,\ \omega_z,\ dl/dS),$$

where $\boldsymbol{\omega}$ is the rotation definition introduced above. Since the initial geometry and hence $\mathbf{T}$, the initial direction of the beam axis, is known, $\mathbf{t} = \mathbf{C}\cdot\mathbf{T}$, where $\mathbf{C}$ is the rotation matrix defined by $\boldsymbol{\omega}$, and hence

$$\frac{d\mathbf{x}}{dS} = \frac{dl}{dS}\mathbf{C}\cdot\mathbf{T} \tag{3.5.3–3}$$

is defined by these variables and the initial geometry. Furthermore, ϕ_a is directly available from $\boldsymbol{\omega}$ and $\mathbf{t}$. Thus, the above set is a satisfactory set of nodal variables. To eliminate the unwanted axial strain constraint, in Abaqus the stretch dl/dS at the node of each such element is taken as an internal variable, local to the element (a third internal node is created for this purpose, and so it is not shared with neighboring elements.)

It should be remarked that the above transformation (Equation 3.5.3–3) is nonlinear. This leads to some complications—for example, the d'Alembert forces no longer have the simple form $M_{MN}\ddot{\overline{u}}^N$; rather, a matrix $\overline{M}_{MN} = Q^{MP}M_{PQ}Q^{QN}$ replaces M_{MN} where Q^{MP} and Q^{QN} use the transformation (Equation 3.5.3–3) and its appropriate time derivatives. The resulting Jacobian is nonsymmetric; Abaqus ignores the nonsymmetric terms.

Integration

The cross-section integration has already been discussed in the context of the shear beams—it is the same for these beams. Along the beam axis, the integration schemes are as described below.

Stiffness and internal forces

Three Gauss points are used. Two Gauss points are not sufficient because the torsional strain is not independent.

Mass and distributed loads

Three Gauss points are used. Rotary inertia is not included in the mass, except for rotation of the section about the beam axis, where it is included to avoid singularity in perfectly straight beams.

Reference

- "Beam modeling: overview," Section 29.3.1 of the Abaqus Analysis User's Guide

3.5.4 HYBRID BEAM ELEMENTS

Product: Abaqus/Standard

The hybrid beam elements in Abaqus/Standard are designed to handle very slender situations, where the axial stiffness of the beam is very large compared to the bending stiffness; and so a mixed method, where axial force is treated as an independent unknown, is required. For the shear beams mixed elements are provided where the transverse shear forces are also treated as independent unknowns. This section discusses the basis of these mixed methods.

Axial and bending behavior

The internal virtual work of the beam can be written

$$\delta W_1^I = \int_L (N\delta\varepsilon + M_1\delta K_1 + M_2\delta K_2 + M_3\delta e_1)dL.$$

Alternatively, we can introduce an independent axial force variable, $\tilde{N}$, and write

$$\delta W_2^I = \int_L (\tilde{N}\delta\varepsilon + M_1\delta K_1 + M_2\delta K_2 + M_3\delta e_1 + \delta\lambda(N-\tilde{N})dL,$$

where $\delta\lambda$ is a Lagrange multiplier introduced to impose the constraint $N = \tilde{N}$. A linear combination of these expressions is

$$\delta W_C^I = \rho\delta W_1^I + (1-\rho)\delta W_2^I, \quad \text{where } \rho \text{ is a parameter that will be defined later.}$$

Then

$$\delta W_C^I = \int_L \Big[(\rho N + (1-\rho)\tilde{N})\delta\varepsilon + M_1\delta K_1 + M_2\delta K_2 + M_3\delta e_1 + (1-\rho)\delta\lambda(N-\tilde{N})\Big]dL.$$

The contribution of this term to the Newton scheme is then

$$\begin{aligned}
&\int_L \Big[(\rho dN + (1-\rho)d\tilde{N})\delta\varepsilon + dM_1\delta K_1 + dM_2\delta K_2 + dM_3\delta e_1 + (1-\rho)\delta\lambda(dN - d\tilde{N}) \\
&\quad + \overline{N}d\delta\varepsilon + M_1 d\delta K_1 + M_2 d\delta K_2 + M_3 d\delta e_1\Big]\, dL \\
&= -\int_L \Big[\overline{N}\delta\varepsilon + M_1\delta K_1 + M_2\delta K_2 + M_3\delta e_1 + (1-\rho)\delta\lambda(N-\tilde{N})\Big]\, dL,
\end{aligned}$$

where

$$\overline{N} = \rho N + (1-\rho)\tilde{N}.$$

The tangent stiffness of the section behavior gives

$$\begin{Bmatrix} d\tilde{N} \\ dM_1 \\ dM_2 \\ dM_3 \end{Bmatrix} = \begin{bmatrix} A_{00} & A_{01} & A_{02} & A_{03} \\ & A_{11} & A_{12} & A_{13} \\ \text{sym} & & A_{22} & A_{23} \\ & & & A_{33} \end{bmatrix} \begin{Bmatrix} d\varepsilon \\ dK_1 \\ dK_2 \\ de_1 \end{Bmatrix}.$$

If $L^2\, A_{00} < A_{11},\ A_{22}$ (where L is the element length), then the beam is flexible axially and the mixed formulation is unnecessary. Otherwise, we assume that an inverse of the first equation above defines $d\varepsilon$ from $d\tilde{N}$:

$$d\varepsilon = \frac{1}{A_{00}}(d\tilde{N} - A_{01}dK_1 - A_{02}dK_2 - A_{03}de_1),$$

and so

$$dM_1 = \Big(A_{11} - \frac{A_{01}^2}{A_{00}}\Big)dK_1 + \Big(A_{12} - \frac{A_{01}A_{02}}{A_{00}}\Big)dK_2 + \Big(A_{13} - \frac{A_{01}A_{03}}{A_{00}}\Big)de_1 + \frac{A_{01}}{A_{00}}d\tilde{N}$$
$$dM_2 = \Big(A_{12} - \frac{A_{01}A_{02}}{A_{00}}\Big)dK_1 + \Big(A_{22} - \frac{A_{02}^2}{A_{00}}\Big)dK_2 + \Big(A_{23} - \frac{A_{02}A_{03}}{A_{00}}\Big)de_1 + \frac{A_{02}}{A_{00}}d\tilde{N}$$

$$dM_3 = \Big(A_{13} - \frac{A_{01}A_{03}}{A_{00}}\Big)dK_1 + \Big(A_{23} - \frac{A_{02}A_{03}}{A_{00}}\Big)dK_2 + \Big(A_{33} - \frac{A_{03}^2}{A_{00}}\Big)de_1 + \frac{A_{03}}{A_{00}}d\tilde{N}.$$

Now using the first tangent section stiffness multiplied by ρ and the second multiplied by $1 - \rho$, the Newton contribution of the element becomes

$$\int_L \lfloor \delta\varepsilon\delta K_1\delta K_2\delta e_1 A_{00}\delta\lambda \rfloor \left[\ \tilde{A}\ \right] \begin{Bmatrix} d\varepsilon \\ dK_1 \\ dK_2 \\ de_1 \\ d\tilde{N} \end{Bmatrix} dL$$
$$+ \int_L (\tilde{N}d\delta\varepsilon + M_1 d\delta K_1 + M_2 d\delta K_2 + M_3 d\delta e_1)dL$$
$$= -\int_L \left[\tilde{N}\delta\varepsilon + M_1\delta K_1 + M_2\delta K_2 + M_3\delta e_1 + A_{00}\delta\lambda(1-\rho)\left(\frac{N-\tilde{N}}{A_{00}}\right)\right] dL,$$

where $\left[\ \tilde{A}\ \right]$ is

$$\begin{bmatrix} \rho A_{00} & \rho A_{01} & \rho A_{02} & \rho A_{03} & 1-\rho \\ & A_{11}-(1-\rho)\frac{A_{01}^2}{A_{00}} & A_{12}-(1-\rho)\frac{A_{01}A_{02}}{A_{00}} & A_{13}-(1-\rho)\frac{A_{01}A_{03}}{A_{00}} & (1-\rho)\frac{A_{01}}{A_{00}} \\ & & A_{22}-(1-\rho)\frac{A_{02}^2}{A_{00}} & A_{23}-(1-\rho)\frac{A_{02}A_{03}}{A_{00}} & (1-\rho)\frac{A_{02}}{A_{00}} \\ & \text{symm} & & A_{33}-(1-\rho)\frac{A_{03}^2}{A_{00}} & (1-\rho)\frac{A_{03}}{A_{00}} \\ & & & & -(1-\rho)\frac{1}{A_{00}} \end{bmatrix}.$$

The variable $\tilde{N}$ is taken as an independent value at each integration point in the element. We choose ρ as $\tilde{\rho}/A_{00}$, where $\tilde{\rho}$ is a small value. With this choice, by ensuring that the variables $\tilde{N}$ are eliminated after the displacement variables of each element, the Gaussian elimination scheme has no difficulty with solving the equations.

Transverse shear

In the mixed elements that allow transverse shear (B21H, B22H, B31H, B32H), the transverse shear constraints are imposed by treating the shear forces as independent variables, using the following formulation. The internal virtual work associated with transverse shear is

$$\delta W_1^{TS} = \int_L T_i \delta\gamma_i \, dL \qquad i = 1,\ 2,$$

where T_1 and T_2 are shear forces on the section, and $\delta\gamma_1$ and $\delta\gamma_2$ are variations of transverse shear strain. The virtual work can also be written by introducing independent shear force variables $\tilde{T}_1$ and $\tilde{T}_2$, as

$$\delta W_2^{TS} = \int \{\tilde{T}_i \delta\gamma_i + \delta\lambda_i (T_i - \tilde{T}_i)\} dL,$$

where the $\delta\lambda_i$ are Lagrange multipliers. As in the axial case, we take a linear combination of these two forms,

$$\delta W^{TS} = \rho \delta W_1^{TS} + (1-\rho)\delta W_2^{TS},$$

where ρ will be defined later. This gives

$$\delta W^{TS} = \int_L \left\{\overline{T}_i \delta\gamma_i + \delta\lambda_i (1-\rho)(T_i - \tilde{T}_i)\right\} dL,$$

where

$$\overline{T}_i = \rho T_i + (1-\rho)\tilde{T}_i.$$

The contribution of this term to the Newton scheme is

$$\int_L \left\{(\rho dT_i + (1-\rho)d\tilde{T}_i)\delta\gamma_i + \delta\lambda_i(1-\rho)(dT_i - d\tilde{T}_i) + \tilde{T}_i d\delta\gamma_i\right\}dL$$

$$= -\int_L \left\{\overline{T}_i\delta\gamma_i + \delta\lambda_i(1-\rho)(T_i - \tilde{T}_i)\right\}dL.$$

Abaqus treats transverse shear elastically, so $T_i = GA\gamma_i$, where GA is constant. Then the Newton contribution is

$$\int_L \left\{(\rho GA\, d\gamma_i + (1-\rho)d\tilde{T}_i)\delta\gamma_i + \delta\lambda_1(1-\rho)(GA\, d\gamma_i - d\tilde{T}_i) + \overline{T}_i\, d\delta\gamma_i\right\}dL$$

$$= -\int_L \left\{\overline{T}_i\delta\gamma_i + \delta\lambda_i(1-\rho)(T_i - \overline{T}_i)\right\}dL.$$

We now define $\delta\tilde{T}_i = GA\delta\lambda_i$ and choose $\rho = \tilde{\rho}/GA$, where $\tilde{\rho}$ is a small value compared to GA, to give

$$\int_L \left\{\tilde{\rho}\, d\gamma_i + (1-\tilde{\rho}/GA)d\tilde{T}_i)\delta\gamma_i + (1-\tilde{\rho}/GA)(d\gamma_i - \frac{1}{GA}d\tilde{T}_i)\delta T_i + \overline{T}_i d\delta\gamma_i\right\}dL$$

$$= -\int_L \left\{\overline{T}_i\delta\gamma_i + (1-\tilde{\rho}/GA)(\gamma_i - \tilde{T}_i/GA)\delta\tilde{T}_i\right\}dL.$$

Reference

- "Choosing a beam element," Section 29.3.3 of the Abaqus Analysis User's Guide

3.5.5 MASS AND INERTIA FOR TIMOSHENKO BEAMS

Products: Abaqus/Standard Abaqus/Explicit

Mass and inertia properties for Timoshenko beams (including PIPE elements) in Abaqus may come from two separate sources. The first source is the beam's own density and the cross-section geometry. The second source comes from any additional mass and inertia properties per element length that may be applied at specified locations on the beam cross-section. When the mass and inertia are given only from the first source, the user has the choice of requesting either an isotropic approximation or the exact rotary inertia formulation for the beam mass matrix. When the isotropic formulation is used, the mass of the beam per length is applied to the beam's node that is located at the origin of the cross-section axis and the offset between the beam's node and the center of mass for the cross-section (if it is nonzero) is neglected in the mass matrix formulation.

Let ρ_B be the beam density. Mass and the rotary inertia about the center of mass in the beam's cross-section coordinate system are defined as

$$
\begin{aligned}
m_B &= \int_V \rho_B \, dV, \\
I_{11} &= \int_V \rho_B (x_2)^2 \, dV, \\
I_{22} &= \int_V \rho_B (x_1)^2 \, dV, \\
I_{12} &= -\int_V \rho_B x_1 x_2 \, dV, \\
I_{33} &= I_{11} + I_{22}.
\end{aligned}
$$

Here, x_1 and x_2 are measured relative to the center of mass of the cross-section.

For the isotropic (approximate) formulation the mass matrix for the element takes the form

$$
\begin{bmatrix} \mathbf{m} & 0 \\ 0 & I_{33}\mathbf{1} \end{bmatrix}.
$$

In two dimensions $I_{33} = I_{11}$.

In all expressions in this section the mass matrix $\mathbf{m}$ that applies to the translational degrees of freedom is lumped for 2-node beams and consistent for 3-node beams.

When the exact formulation is used, any offset between the beam's node and the center of mass for the cross-section will produce coupling between the translational degrees of freedom and rotational degrees of freedom in the mass matrix for the element.

Let $m_A(x_\alpha)$ define the added mass per beam length. The combined beam mass is defined as

$$
m = m_B + \int_l m_A \, dl.
$$

Let $\mathbf{r}$ be the vector between the center of mass C and some point with current coordinates $\mathbf{x}$,

$$\mathbf{r} = \mathbf{x}_C - \mathbf{x}.$$

For a rigid body the velocity of any point in the body is given by

$$\dot{\mathbf{u}} = \dot{\mathbf{u}}_C - \boldsymbol{\omega} \times \mathbf{r},$$

where $\boldsymbol{\omega} = \dot{\boldsymbol{\phi}}$ is the angular velocity of the body. Taking the time derivative of this expression, the acceleration is

$$\ddot{\mathbf{u}} = \ddot{\mathbf{u}}_C - \dot{\boldsymbol{\omega}} \times \mathbf{r} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}).$$

The local or strong form of the equilibrium equations represents the balance of linear momentum and balance of angular momentum; these two equilibrium equations are

$$\begin{aligned} m\frac{d}{dt}\left(\dot{\mathbf{u}} + \boldsymbol{\omega} \times \mathbf{r}\right) &= \bar{\mathbf{F}}\,, \\ \frac{d}{dt}\left[\mathbf{I}\cdot\boldsymbol{\omega} + m\mathbf{r} \times \left(\dot{\mathbf{u}} + \boldsymbol{\omega} \times \mathbf{r}\right)\right] &= \bar{\mathbf{M}} + \mathbf{r} \times \bar{\mathbf{F}}\,, \end{aligned}$$

where $\bar{\mathbf{F}}$ and $\bar{\mathbf{M}}$ are external forces acting at the center of mass and external moment and $\mathbf{I}$ is the rotary inertia tensor.

The variational or weak form of equilibrium is

$$\delta W_{int} + \delta W_{ext} = 0\,.$$

Taking the time derivative in the equilibrium equations, the internal or d'Alembert force contribution is

$$\begin{aligned} -\delta W_{int} = \int_V \rho\, \delta\mathbf{u}\cdot\ddot{\mathbf{u}}\, dV = m\left[\ddot{\mathbf{u}} + \dot{\boldsymbol{\omega}} \times \mathbf{r} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r})\right]\cdot\delta\mathbf{u} \\ + \left[\mathbf{I}\cdot\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}\cdot\boldsymbol{\omega} + m\left((\boldsymbol{\omega} \times \mathbf{r}) \times \dot{\mathbf{u}} + \mathbf{r} \times \ddot{\mathbf{u}} + \left(r^2\mathbf{1} - \mathbf{r}\mathbf{r}\right)\cdot\dot{\boldsymbol{\omega}} - (\boldsymbol{\omega}\cdot\mathbf{r})\,(\boldsymbol{\omega} \times \mathbf{r})\right)\right]\cdot\delta\boldsymbol{\theta}, \end{aligned}$$

where $\delta\mathbf{u}$ is the variation of the position of a point in the body and $\delta\boldsymbol{\theta}$ is the variation of the rotation of the rigid body reference node. The external loading contribution is

$$\delta W_{ext} = \bar{\mathbf{F}}\cdot\delta\mathbf{u} + \left(\bar{\mathbf{M}} + \mathbf{r} \times \mathbf{F}\right)\cdot\delta\boldsymbol{\theta}.$$

For linear theory all nonlinear terms are neglected, so the internal force contribution simplifies to

$$\begin{aligned} -\delta W_{int} = \int_V \rho\, \delta\mathbf{u}\cdot\ddot{\mathbf{u}}\, dV = m\left[\ddot{\mathbf{u}} + \dot{\boldsymbol{\omega}} \times \mathbf{r}\right]\cdot\delta\mathbf{u} \\ + \left[\mathbf{I}\cdot\dot{\boldsymbol{\omega}} + m\left(\mathbf{r} \times \ddot{\mathbf{u}} + \left(r^2\mathbf{1} - \mathbf{r}\mathbf{r}\right)\cdot\dot{\boldsymbol{\omega}}\right)\right]\cdot\delta\boldsymbol{\theta} \end{aligned}$$

and leads to the following mass matrix for all linear and linear perturbation analyses:

$$\begin{bmatrix} \mathbf{m} & -\mathbf{m}\cdot\hat{\mathbf{r}} \\ \mathbf{m}\cdot\hat{\mathbf{r}} & \mathbf{I}_{\mathbf{eff}} \end{bmatrix},$$

where $\hat{\mathbf{r}}$ denotes a skew symmetric matrix and $\mathbf{I}_{\mathbf{eff}} = \mathbf{I} + r^2\mathbf{m} - \mathbf{m}\cdot\mathbf{rr}$.

When the inertia of a rigid body is used with implicit time integration, the Jacobian contribution of δW_{int} is required. It can be written in the form

$$-d\delta W_{int} = m\delta\mathbf{u}\cdot d\mathbf{U} + \delta\boldsymbol{\theta}\cdot\left(d\mathbf{K_1} + m d\mathbf{K_2}\right),$$

where the following notation was used

$$d\mathbf{U} = d\ddot{\mathbf{u}} + d\dot{\boldsymbol{\omega}}\times\mathbf{r} + \dot{\boldsymbol{\omega}}\times(d\boldsymbol{\theta}\times\mathbf{r}) + d\boldsymbol{\omega}\times(\boldsymbol{\omega}\times\mathbf{r}) + \boldsymbol{\omega}\times(d\boldsymbol{\omega}\times\mathbf{r}) + \boldsymbol{\omega}\times(\boldsymbol{\omega}\times(d\boldsymbol{\theta}\times\mathbf{r})),$$

$$\begin{aligned} d\mathbf{K_1} =& \mathbf{I}\cdot d\dot{\boldsymbol{\omega}} + d\boldsymbol{\omega}\times\mathbf{I}\cdot\boldsymbol{\omega} + \boldsymbol{\omega}\times\mathbf{I}\cdot d\boldsymbol{\omega} + d\boldsymbol{\theta}\times\mathbf{I}\cdot\dot{\boldsymbol{\omega}} + \mathbf{I}\cdot(\dot{\boldsymbol{\omega}}\times d\boldsymbol{\theta}) + (\boldsymbol{\omega}\cdot\mathbf{I}\cdot\boldsymbol{\omega})d\boldsymbol{\theta} \\ & - (\boldsymbol{\omega}\cdot d\boldsymbol{\theta})\mathbf{I}\cdot\boldsymbol{\omega} + \boldsymbol{\omega}\times\mathbf{I}\cdot(\boldsymbol{\omega}\times d\boldsymbol{\theta}), \end{aligned}$$

$$\begin{aligned} d\mathbf{K_2} =& \left[\mathbf{r}\dot{\mathbf{u}} - (\mathbf{r}\cdot\dot{\mathbf{u}})\mathbf{1} - (\boldsymbol{\omega}\times\mathbf{r})\mathbf{r} + (\boldsymbol{\omega}\cdot\mathbf{r})\hat{\mathbf{r}}\right]\cdot d\boldsymbol{\omega} \\ & + \left[-(\boldsymbol{\omega}\cdot\mathbf{r})\hat{\dot{\mathbf{u}}} - (\mathbf{r}\times\dot{\mathbf{u}})\boldsymbol{\omega} + \mathbf{r}\ddot{\mathbf{u}} - (\mathbf{r}\cdot\ddot{\mathbf{u}})\mathbf{1} + (\mathbf{r}\cdot\dot{\boldsymbol{\omega}})\hat{\mathbf{r}} - \mathbf{rr}\cdot\hat{\dot{\boldsymbol{\omega}}} + (\boldsymbol{\omega}\times\mathbf{r})\,\boldsymbol{\omega}\cdot\hat{\mathbf{r}}\right. \\ & \left. - (\boldsymbol{\omega}\cdot\mathbf{r})\left((\boldsymbol{\omega}\cdot\mathbf{r})\mathbf{1} - \mathbf{r}\boldsymbol{\omega}\right)\right]\cdot d\boldsymbol{\theta} + (\mathbf{r}\boldsymbol{\omega} - \boldsymbol{\omega}\mathbf{r})\cdot d\dot{\mathbf{u}} + \hat{\mathbf{r}}\cdot d\ddot{\mathbf{u}} + (r^2\mathbf{1} - \mathbf{rr})\cdot d\dot{\boldsymbol{\omega}}. \end{aligned}$$

In the Jacobian formulation for 3-node beams, a consistent mass matrix is used for translational degrees of freedom and a lumped mass matrix is used for rotational degrees of freedom and the terms that couple translational and rotational degrees of freedom.

3.5.6 MESHED BEAM CROSS-SECTIONS

Product: Abaqus/Standard

Overview

The beam theory introduced in "Beam element formulation," Section 3.5.2, applies to homogeneous beams (made out of a single material) and assumes that the shear center of the beam cross-section is either known or can be easily calculated. However, if the beam cross-section is arbitrarily shaped and/or the beam is made of more than one material layer, finding the cross-section shear center and warping function are no longer trivial tasks. To perform these tasks, the cross-section has to be numerically integrated using a finite element discretization over the two-dimensional cross-section region. The nodal degrees of freedom of the finite element cross-section model represent warping displacements (in general, in and out-of-plane warping degrees of freedom) that allow the shear center and beam torsional stiffness to be determined. Numerical integration of this meshed section also provides the stiffness and inertia statistics: integrated axial stiffness $\sum_i E_i A_i$, integrated bending stiffness $(\sum_i E_i I_i)_{\alpha\beta}$, integrated shear stiffness $(\sum_i G_i A_i)_{\alpha\beta}$, total mass per unit length $\sum_i \rho_i A_i$, and rotary inertia $(\sum_i \rho_i I_i)_{\alpha\beta}$.

The warping function and the shear center are derived for shear flexible Timoshenko beams under the following assumptions:

- Cross-sections are solid or closed and thin-walled and have a torsional constant that is of the same order of magnitude as the polar moment of inertia of the section. Hence, in the elastic range the warping is small, and it is assumed that warping prevention at the ends can be neglected. The axial warping stresses are assumed to be negligible, but the torsional shear stresses are assumed to be of the same order of magnitude as the stresses due to axial forces and bending moments. In this case the warping is dependent on the twist and can be eliminated as an independent variable, which leads to a considerably simplified formulation. Hence, the theory is based on a solid cross-section with unconstrained warping. Using the notation from "Beam element formulation," Section 3.5.2, we assume that $w_p = 0$ and the axial strains due to warping can be neglected: $\chi\,\Omega \approx 0$.
- Beams can be made out of linear elastic materials either with isotropic properties or orthotropic shear properties defined by two shear moduli G_1 and G_2 given in two perpendicular directions. The stress-strain relationship for the elastic orthotropic material in the beam cross-section axis yields

$$\begin{Bmatrix} \sigma_{11} \\ \tau_{21} \\ \tau_{31} \end{Bmatrix} = \begin{bmatrix} E & 0 & 0 \\ & G_1(\cos\alpha)^2 + G_2(\sin\alpha)^2 & (G_1 - G_2)\cos\alpha\sin\alpha \\ sym & & G_1(\sin\alpha)^2 + G_2(\cos\alpha)^2 \end{bmatrix} \begin{Bmatrix} e_{11} \\ \gamma_{21} \\ \gamma_{31} \end{Bmatrix},$$

where σ_{11} represents the beam's axial stress, τ_{21} and τ_{31} represent two shear stresses, and the angle α is a user-defined material orientation. For the isotropic material properties the above relationship becomes

$$\left\{ \begin{array}{c} \sigma_{11} \\ \tau_{21} \\ \tau_{31} \end{array} \right\} = \left[\begin{array}{ccc} E & 0 & 0 \\ & G & 0 \\ sym & & G \end{array} \right] \left\{ \begin{array}{c} e_{11} \\ \gamma_{21} \\ \gamma_{31} \end{array} \right\}.$$

- Material fibers are aligned with or perpendicular to the beam axis; hence, in-plane warping can be neglected and the out-of-plane degree of freedom is the only unknown warping value. This assumption can be inaccurate if the beam consists of materials with very different stiffness properties.

Defining the shear center and warping function

At a given stage in the deformation history of the beam, the position of a material point in the cross-section is given by the expression

$$\hat{\mathbf{x}}(S, S^\alpha) = \mathbf{x}(S) + f(S) S^\alpha \mathbf{n}_\alpha(S) + w(S)\psi(S^\alpha)\mathbf{t}(S).$$

Applying the assumptions made for meshed sections, the expressions for the axial and transverse shear strain components simplify to

$$e_{11} = \bar{e} - S^\alpha \epsilon^\beta_\alpha b_\beta + \chi\psi,$$

$$\gamma_{(\alpha+1)1} = \bar{\gamma}_\alpha + b(S^\beta \epsilon^\alpha_\beta + \frac{\partial \psi}{\partial S^\alpha}).$$

Express these strain components relative to the centroid and the shear center strains, respectively, as

$$e_{11} = e^c - (S^\alpha - S^\alpha_c)\epsilon^\beta_\alpha b_\beta + \chi\psi,$$

$$\gamma_{(\alpha+1)1} = \bar{\gamma}^s_\alpha + b\left\{ (S^\beta - S^\beta_s)\epsilon^\alpha_\beta + \frac{\partial \psi}{\partial S^\alpha} \right\},$$

where S^α_c is a section centroid, S^α_s is a section shear center, e^c is the axial strain at the centroid, and $\bar{\gamma}^s_\alpha$ is the shear strain at the shear center.

The elastic energy in the beam is

$$\Pi = \frac{1}{2}\int_\ell \int_A (E e^2_{11} + G_{\alpha\beta}\gamma_{(\alpha+1)1}\gamma_{(\beta+1)1}) d\ell dA.$$

Using the strain definitions relative to the section strains at the centroid and the shear center, the elastic energy can be written as

$$\Pi = \frac{1}{2}\int_\ell \int_A \{E(e^c - (S^\alpha - S_c^\alpha)\epsilon_\alpha^\beta b_\beta + \chi\psi)^2 + G_{\alpha\beta}(\bar{\gamma}_\alpha^s + b\psi_{,\alpha} + b(S^\delta - S_s^\delta)\epsilon_\delta^\alpha)(\bar{\gamma}_\beta^s + b\psi_{,\beta} + b(S^\delta - S_s^\delta)\epsilon_\delta^\beta)\} d\ell dA.$$

Although we assume no warping prevention (i.e., $\chi = 0$), the above energy leads to the following condition that requires the warping function to be orthogonal to the axial and bending energies:

$$\int_A E\left\{e^c - b_\alpha \epsilon_\beta^\alpha (S^\beta - S_c^\beta)\right\}\psi dA = 0.$$

The cross-section centroid is defined as the point about which the coupling between axial and bending vanishes. Hence, the centroid location follows from

$$e^c b_\alpha \epsilon_\beta^\alpha \int_A E(S^\beta - S_c^\beta) dA = 0,$$

and

$$S_c^\alpha = \frac{\int_A S^\alpha E dA}{\int_A E dA}.$$

The shear center is defined as the point about which the coupling between twist and transverse shear vanishes. Hence, the following term is zero in the elastic energy:

$$\int_A G_{\alpha\beta}\bar{\gamma}_\alpha^s b(\psi_{,\beta} + (S^\delta - S_s^\delta)\epsilon_\delta^\beta) dA = 0.$$

Let us express the warping function as a sum of three parts: a warping function ψ_0 superimposed on the unknown rigid translation C_0 and rigid rotation about the yet unknown shear center S_s^α. This assumption can be written as

$$\psi = \psi_0 + C_0 - S_s^\delta \epsilon_\alpha^\delta (S^\alpha - S_c^\alpha).$$

Substituting the above into the expression for elastic energy, using the property of the shear center, and minimizing the energy with respect to the warping function, we get

$$\int_A \frac{\partial \delta\psi}{\partial S^\alpha} G_{\alpha\beta}\left(\frac{\partial \psi_0}{\partial S^\beta} + \epsilon_\delta^\beta S^\delta\right) dA = 0.$$

This equation is solved numerically over the meshed section and gives the value of $\psi_0(S_\alpha)$.

Recall that the warping function satisfies the orthogonality condition

$$\int_A \left\{e_c - b_\alpha \epsilon_\delta^\alpha (S^\delta - S_c^\delta)\right\} E\psi dA = 0.$$

Substituting ψ, grouping axial and bending terms, and using the centroid definition, we get

$$\int_A \left\{ e_c E(\psi_0 + C_0) - b_\alpha \epsilon^\alpha_\beta (S^\beta - S^\beta_c) E[\psi_0 - S^\delta_s \epsilon^\delta_\alpha (S^\alpha - S^\alpha_c)] \right\} dA = 0.$$

This expression must be true for any value of axial strain e_c and curvatures b_α, so we can write two separate equations that provide constant C_0 and shear center components S^α_s:

$$\int_A E\left(\psi_0 + C_0\right) dA = 0,$$

$$\int_A \epsilon^\alpha_\beta (S^\beta - S^\beta_c) E\left\{\psi_0 - S^\delta_s \epsilon^\delta_\alpha (S^\alpha - S^\alpha_c)\right\} dA = 0.$$

Hence,

$$C_0 = -\frac{\int_A E\psi_0 dA}{\int_A E dA},$$

$$S^\delta_s = (EI)^{-1}_{\alpha\delta} \int_A \epsilon^\alpha_\beta (S^\beta - S^\beta_c) E\psi_0 dA.$$

Finally, the section integrated stiffness properties are defined as

$$(EA) = \int_A E dA,$$

$$(EI)_{\alpha\beta} = \int_A E\epsilon^\alpha_\gamma (S^\gamma - S^\gamma_c)\epsilon^\beta_\delta (S^\delta - S^\delta_c) dA,$$

$$(GA)_{\alpha\beta} = \int_A G_{\alpha\beta} dA,$$

$$(GJ) = \int_A G_{\alpha\beta}(\psi_{,\alpha} + \epsilon^\alpha_\gamma (S^\gamma - S^\gamma_s))(\psi_{,\beta} + \epsilon^\beta_\delta (S^\delta - S^\delta_s)) dA.$$

The integrated inertia properties are

$$(\rho A) = \int_A \rho dA,$$

$$(\rho I)_{\alpha\beta} = \int_A \rho\epsilon^\alpha_\gamma (S^\gamma - S^\gamma_m)\epsilon^\beta_\delta (S^\delta - S^\delta_m) dA,$$

where S^α_m is the center of mass given by the equation

$$S^\alpha_m = \frac{\int_A S^\alpha \rho dA}{\int_A \rho dA}.$$

We assume elastic section behavior in transverse shear and we neglect the effect at the individual material points (shear strain and stress is averaged over the section). This leads to the following relationships for transverse shear stiffness:

$$\bar{K}^{ts}_{\alpha\beta} = f_p K^{ts}_{\alpha\beta},$$

$$f_p = 1/(1 + \xi \frac{l^2(EA)}{12(EI)_{av}}),$$

$$(EI)_{av} = 1/3((EI)_{11} + (EI)_{22} + (EI)_{12}).$$

$$K^{ts}_{\alpha\beta} = k(GA)_{\alpha\beta},$$

where k equals 1.0 for meshed cross-sections and ξ depends on the finite element interpolation.

3.6 Shell elements

- "Shell element overview," Section 3.6.1
- "Axisymmetric shell elements," Section 3.6.2
- "Shear flexible small-strain shell elements," Section 3.6.3
- "Triangular facet shell elements," Section 3.6.4
- "Finite-strain shell element formulation," Section 3.6.5
- "Small-strain shell elements in Abaqus/Explicit," Section 3.6.6
- "Axisymmetric shell element allowing asymmetric loading," Section 3.6.7
- "Transverse shear stiffness in composite shells and offsets from the midsurface," Section 3.6.8
- "Rotary inertia for 5 degree of freedom shell elements," Section 3.6.9

3.6.1 SHELL ELEMENT OVERVIEW

The Abaqus shell element library provides elements that allow the modeling of curved, intersecting shells that can exhibit nonlinear material response and undergo large overall motions (translations and rotations). Abaqus shell elements can also model the bending behavior of composites.

The library is divided into three categories consisting of general-purpose, thin, and thick shell elements. Thin shell elements provide solutions to shell problems that are adequately described by classical (Kirchhoff) shell theory, thick shell elements yield solutions for structures that are best modeled by shear flexible (Mindlin) shell theory, and general-purpose shell elements can provide solutions to both thin and thick shell problems. All shell elements use bending strain measures that are approximations to those of Koiter-Sanders shell theory (Budiansky and Sanders, 1963). While Abaqus/Standard provides shell elements in all three categories, Abaqus/Explicit provides only general-purpose shell elements. For most applications the general-purpose shell elements should be the user's first choice from the element library. However, for specific applications it may be possible to obtain enhanced performance by choosing one of the thin or thick shell elements. It should also be noted that not all Abaqus shell elements are formulated for large-strain analysis.

The general-purpose shell elements are axisymmetric elements SAX1, SAX2, and SAX2T and three-dimensional elements S3, S4, S3R, S4R, S4RS, S3RS, and S4RSW, where S4RS, S3RS, and S4RSW are small-strain elements that are available only in Abaqus/Explicit. The general-purpose elements provide robust and accurate solutions in all loading conditions for thin and thick shell problems. Thickness change as a function of in-plane deformation is allowed in their formulation. They do not suffer from transverse shear locking, nor do they have any unconstrained hourglass modes. With the exception of the small-strain elements, all of these elements consider finite membrane strains. No hourglass control is required for the axisymmetric general-purpose shells, nor in the bending and membrane response of the fully integrated element S4. The membrane kinematics of S4 are based on an assumed-strain formulation that provides accurate solutions for in-plane bending behavior. The Abaqus/Explicit elements S3RS, S4RS, and S4RSW are well-suited for many impact dynamics problems, including structures undergoing large-scale buckling behavior, which involve small-strains but large rotations and severe bending. These elements use simplified methods for strain calculation and hourglass control and offer significant advantages in computational speed.

Thin shell elements are available only in Abaqus/Standard. STRI3 and STRI65 are triangular small-strain, thin shell elements; S4R5, S8R5, and S9R5 comprise the quadrilateral small-strain, thin shell elements, while SAXA is a finite-strain, thin shell element suitable for modeling axisymmetric geometries subjected to arbitrary loadings. Thin shell elements may provide enhanced performance for large problems where reducing the number of degrees of freedom through the use of five degree of freedom shells is desirable. However, they should be used only for the modeling of thin structures that exhibit at most weak nonlinearities in problems where rotation degree of freedom output is not required and for situations where the shell surface and the displacement field are smooth so that higher accuracy can be achieved with the use of second-order shells. SAXA elements very effectively model axisymmetric structures undergoing asymmetric deformation when only a few circumferential Fourier modes describe the circumferential variation of the deformation accurately.

The Discrete Kirchhoff (DK) constraint, which refers to the satisfaction of the Kirchhoff constraint at discrete points on the shell surface, is imposed in all thin shell elements in Abaqus. For element type STRI3 the constraint is imposed analytically and involves no transverse shear strain energy calculation. Solutions

obtained with these elements converge to those corresponding to classical shell theory. For element types STRI65, S4R5, S8R5, S9R5, and SAXA the discrete Kirchhoff constraint is imposed numerically where the transverse shear stiffness acts as a penalty that enforces the constraint.

Shell behavior that can be properly described with shear flexible shell theory and results in smooth displacement fields can be analyzed accurately with the second-order Abaqus/Standard thick shell element S8R. Nonnegligible transverse shear flexibility is required for this element to function properly; hence, the element is suitable for the analysis of composite and sandwich shells. Irregular meshes of S8R elements converge very poorly because of severe transverse shear locking; therefore, this element is recommended for use in regular mesh geometries for thick shell applications.

Thickness change

In geometrically nonlinear analyses in Abaqus/Standard the cross-section thickness of finite-strain shell elements changes as a function of the membrane strain based on a user-defined "effective section Poisson's ratio," ν. In Abaqus/Explicit the thickness change is based on the "effective section Poisson's ratio" for all shell elements in large-deformation analyses, unless the user has specified that the thickness change should be based on the element material definition. The thickness change based on the "effective section Poisson's ratio" is calculated as follows.

In plane stress $\sigma_{33} = 0$; linear elasticity gives

$$\epsilon_{33} = -\frac{\nu}{1-\nu}(\epsilon_{11} + \epsilon_{22}).$$

Treating these as logarithmic strains,

$$\ln\left(\frac{t}{t_0}\right) = -\frac{\nu}{1-\nu}\left(\ln\left(\frac{l_1}{l_1^0}\right) + \ln\left(\frac{l_2}{l_2^0}\right)\right) = -\frac{\nu}{1-\nu}\ln\left(\frac{A}{A^0}\right),$$

where A is the area on the shell's reference surface. This nonlinear analogy with linear elasticity leads to the thickness change relationship:

$$\frac{t}{t_0} = \left(\frac{A}{A_0}\right)^{-\frac{\nu}{1-\nu}}.$$

For $\nu = 0.5$ the material is incompressible; for $\nu = 0.0$ the section thickness does not change.

Reference

- "Shell elements: overview," Section 29.6.1 of the Abaqus Analysis User's Guide

3.6.2 AXISYMMETRIC SHELL ELEMENTS

Products: Abaqus/Standard Abaqus/Explicit

These two shell elements are axisymmetric versions of the shells described in the previous section and use the "reduced-integration penalty" method of Hughes et al. (1977). While these are shell elements, they are also simple extensions of the two-dimensional beam elements B21 and B22. The extension is the inclusion of the hoop terms. These elements are thus one-dimensional, deforming in a radial plane. The Cartesian coordinates in this plane are r (radius) and z (axial position). Distance along the shell reference surface in such a plane is measured by the material coordinate S (see Figure 3.6.2–1).

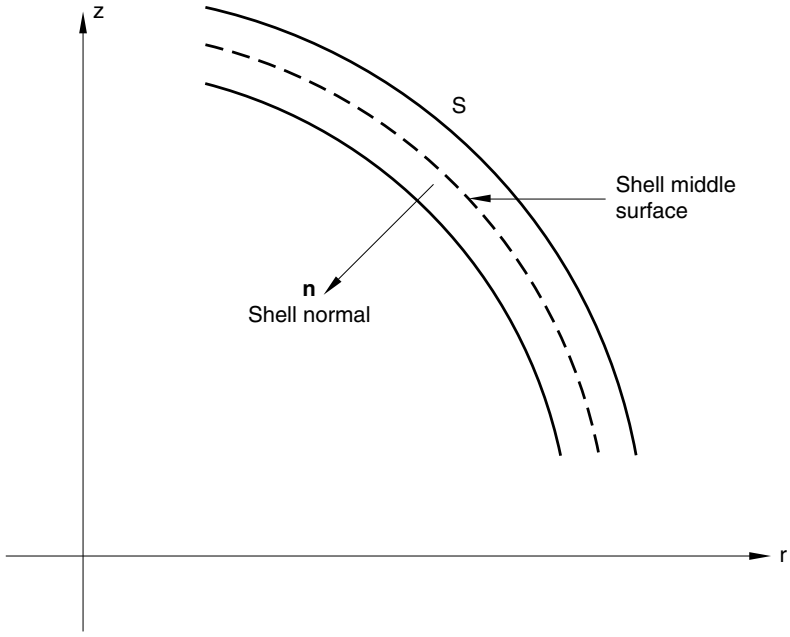


Figure 3.6.2–1 Axisymmetric shell.

Interpolation and integration

The 2-node element (SAX1) uses one-point integration of the linear interpolation function for the distribution of loads. The mass matrix is lumped. The 3-node element (SAX2) uses two-point integration of a quadratic interpolation function for the stiffness and three-point integration of a quadratic interpolation function for the distribution of loads. SAX2 uses a consistent mass matrix. All

integrations use the Gauss method. The integration through the thickness follows the usual numerical or exact scheme used in Abaqus.

Theory

This shell theory allows for finite strains and rotations of the shell. The strain measure used is chosen to give a close approximation (accurate to second-order terms) to log strain. Thus, the theory is intended for direct application to cases involving inelastic or hypoelastic deformation where the stress-strain behavior is given in terms of Kirchhoff stress ("true" stress in the usual engineering literature) and log strain, such as metal plasticity. The theory is approximate, but the approximations are not rigorously justified: they are introduced for simplicity and seem reasonable. These approximations are as follows:

a. A "thinness" assumption is made. This means that, at all times, only terms up to first order with respect to the thickness direction coordinate are included.

b. The thinning of the shell caused by stretching parallel to its reference surface is assumed to be uniform through the thickness and defined by an incompressibility condition on the reference surface of the shell. Obviously this is a relatively coarse approximation, especially in the case where a shell is subjected to pure bending. It is adopted because it is simple and models the effect of thinning associated with membrane straining: this is considered to be of primary importance in the type of applications envisioned, such as the failure of pipes and vessels subjected to over-pressurization.

c. The thinning of the shell is assumed to occur smoothly—that is to say, gradients of the thinning with respect to position on the reference surface are assumed to be negligible. This means that localization effects, such as necking of the shell, are only modeled in a very coarse way. Again, the reason for adopting this approximation is simplicity—details of localization effects are not important to the type of application for which the elements are designed.

d. All stresses except those parallel to the reference surface are neglected; and, for the nonnegligible stresses, plane stress theory is assumed. As with (c) above, this precludes detailed localization studies, but introduces considerable simplification into the formulation.

e. Plane sections remain plane. This has been shown to be consistent with the thinness assumption, (a) above, for most material models. Here it is simply assumed without further justification.

f. Transverse shears are assumed to be small, and the material response to such deformation is assumed to be linear elastic. Transverse shear is introduced because the elements used are of the "reduced integration, penalty" type (see Hughes et al., 1977, for example). In these elements position on the reference surface and rotation of lines initially orthogonal to the reference surface are interpolated independently: the transverse shear stiffness is then viewed as a penalty term imposing the necessary constraint at selected (reduced integration) points. This transverse shear stiffness is the actual elastic value for relatively thick shells. For thinner cases the penalty must be reduced for numerical reasons—this is done in Abaqus in the manner described in Hughes et al. (1977).

The theory is now described in detail. The concepts are taken from various sources, most especially Budiansky and Sanders (1963) and Rodal and Witmer (1979). The position of a material point in the shell is given by

$$\mathbf{x}^1 = \mathbf{x} + \lambda_t \eta\, \mathbf{n}, \tag{3.6.2–1}$$

where

$\mathbf{x}(\theta^1,\ \theta^2)$	is the position of a point on the reference surface of the shell;
$\mathbf{n}(\theta^1, \theta^2)$	is a unit vector in the "thickness" direction, this direction being initially orthogonal to the reference surface;
λ_t	is the stretch of the shell in the thickness direction;
η	measures position with respect to the thickness direction, in the reference configuration; and
$(\theta^1,\ \theta^2)$	are material coordinates in the reference surface.

The assumptions listed above imply that $\lambda_t = \lambda_t(\theta^1,\ \theta^2)$ only and that $\partial\lambda_t/\partial\theta^1,\ \partial\lambda_t/\partial\theta^2$ are small quantities. Equation 3.6.2–1 is written at the end of an increment, and at the start of an increment the same equation is written as

$$\mathbf{X}^1 = \mathbf{X} + \lambda_t^o \eta\, \mathbf{N}. \tag{3.6.2–2}$$

The metric at the end of an increment is

$$\begin{aligned} \frac{\partial \mathbf{x}^1}{\partial\theta^\alpha} \cdot \frac{\partial \mathbf{x}^1}{\partial\theta^\beta} &= \left(\frac{\partial \mathbf{x}}{\partial\theta^\alpha} + \lambda_t \eta \frac{\partial \mathbf{n}}{\partial\theta^\alpha}\right) \cdot \left(\frac{\partial \mathbf{x}}{\partial\theta^\beta} + \lambda_t n \frac{\partial \mathbf{n}}{\partial\theta^\beta}\right) \\ \approx & \frac{\partial \mathbf{x}}{\partial\theta^\alpha} \cdot \frac{\partial \mathbf{x}}{\partial\theta^\beta} + \eta\lambda_t \left(\frac{\partial \mathbf{x}}{\partial\theta^\alpha} \cdot \frac{\partial \mathbf{n}}{\partial\theta^\beta} + \frac{\partial \mathbf{n}}{\partial\theta^\alpha} \cdot \frac{\partial \mathbf{x}}{\partial\theta^\beta}\right) \\ = & g_{\alpha\beta} + \eta\lambda_t b_{\alpha\beta}, \ \text{ say,} \end{aligned} \tag{3.6.2–3}$$

where

$$g_{\alpha\beta} = \frac{\partial \mathbf{x}}{\partial\theta^\alpha} \cdot \frac{\partial \mathbf{x}}{\partial\theta^\beta} \text{ is the metric of the reference surface,}$$

and

$$b_{\alpha\beta} = \frac{\partial \mathbf{x}}{\partial\theta^\alpha} \cdot \frac{\partial \mathbf{n}}{\partial\theta^\beta} + \frac{\partial \mathbf{n}}{\partial\theta^\alpha} \cdot \frac{\partial \mathbf{x}}{\partial\theta^\beta}$$

is an approximation to the curvature tensor (second fundamental form) of the reference surface. $b_{\alpha\beta}$ would be precisely the curvature tensor as it is usually defined if

$$\mathbf{n} \cdot \frac{\partial \mathbf{x}}{\partial\theta^\alpha} = 0.$$

This is only approximately true for these elements, because a small transverse shear is allowed.

At the start of the increment the same quantities are

$$\frac{\partial \mathbf{X}^1}{\partial \theta^\alpha} \cdot \frac{\partial \mathbf{X}^1}{\partial \theta^\beta} = G_{\alpha\beta} + \eta \lambda_t^o B_{\alpha\beta}. \tag{3.6.2–4}$$

Axisymmetric shells undergoing axisymmetric deformations have the great simplification that principal directions do not rotate. Thus, by assuming that θ^1 and θ^2 are oriented in these principal directions (θ^1 is meridional and θ^2 is circumferential), the stretch ratios that occur within the increment in these directions are written as

$$\Delta\lambda_\alpha^1 = \left(\frac{\partial \mathbf{x}^1}{\partial \theta^\alpha} \cdot \frac{\partial \mathbf{x}^1}{\partial \theta^\alpha} \,/\, \frac{\partial \mathbf{X}^1}{\partial \theta^\alpha} \cdot \frac{\partial \mathbf{X}^1}{\partial \theta^\alpha} \right)^{-\frac{1}{2}},$$

where from this point onward the summation convention has been dropped for indexes α and β. Using Equation 3.6.2–3 and Equation 3.6.2–4 and truncating to first order in η then gives

$$\Delta\lambda_\alpha^1 \approx \Delta\lambda_\alpha (1 + \eta \lambda_t \Delta k_{\alpha\alpha}), \tag{3.6.2–5}$$

where

$$\Delta\lambda_\alpha = (g_{\alpha\alpha} \,/\, G_{\alpha\alpha})^{\frac{1}{2}}$$

and

$$\Delta k_{\alpha\alpha} = \lambda_t \frac{b_{\alpha\alpha}}{g_{\alpha\alpha}} - \frac{\lambda_t^o}{\Delta_t} \frac{B_{\alpha\alpha}}{G_{\alpha\alpha}}.$$

The incremental strain, $\Delta\varepsilon_{\alpha\alpha}^1$, is defined as

$$\Delta\varepsilon_{\alpha\alpha}^1 = 2 \left(\frac{\Delta\lambda_\alpha^1 - 1}{\Delta\lambda_\alpha^1 + 1} \right).$$

Because this expression approximates the increment of log strain correctly to second-order terms, it can be thought of as a central difference approximation for the rate of deformation. This expression is used because we anticipate that strain increments of a maximum of 20 percent per increment will be used: at that magnitude the difference between this definition of incremental strain and the increment of log strain is about 1%, which seems to be acceptable (4 % of the increment). At lower—and probably more typical—values of strain increment, the error is very much less. Again expanding to first order in the thickness direction coordinate, η, we obtain

$$\Delta\varepsilon_{\alpha\alpha}^1 \approx \Delta\varepsilon_{\alpha\alpha} + \eta \frac{4\Delta\lambda_\alpha}{(1 + \Delta\lambda_\alpha)^2} \Delta k_{\alpha\alpha},$$

where $\Delta\varepsilon_{\alpha\alpha}$ is the incremental strain of the reference surface—the membrane strain. Now consider the term

$$\frac{4\Delta\lambda_\alpha}{(1+\Delta\lambda_\alpha)^2}.$$

Write $\Delta\lambda_\alpha = 1 + e$, where e represents the change in length per unit length that occurs within the increment (the "nominal strain" with respect to the configuration at the beginning of the increment).

Then

$$\frac{4\Delta\lambda_\alpha}{(1+\Delta\lambda_\alpha)^2} = \frac{1+e}{(1+e/2)^2} = (1+e)(1-e+\frac{3}{4}e^2-\frac{1}{2}e^3+\cdots)$$
$$= (1-\frac{1}{4}e^2+\frac{1}{4}e^3-\cdots).$$

Again, if $e \approx 20$ percent, this means that

$$\frac{4\Delta\lambda_\alpha}{(1+\Delta\lambda_\alpha)^2} \approx 1 - .01 + 0(e^3),$$

and so once again using the argument that practical applications will involve strain increments of no more than a few percent, we approximate

$$\frac{4\Delta\lambda_\alpha}{(1+\Delta\lambda_\alpha)^2} \approx 1.$$

This then gives

$$\Delta\varepsilon^1_{\alpha\alpha} \approx \Delta\varepsilon_{\alpha\alpha} + \eta\lambda_t\Delta k_{\alpha\alpha}. \tag{3.6.2–6}$$

The stretch ratio in the thickness direction is assumed to be defined by the following relation on the reference surface:

$$\lambda_t = (\lambda_1\lambda_2)^{-\nu/1-\nu}\lambda_t^{\text{th}},$$

where λ_t^{th} is the thickness stretch ratio caused by thermal expansion.

From the definition of $\Delta k_{\alpha\alpha}$,

$$\Delta k_{\alpha\alpha} = \frac{b_{\alpha\alpha}}{g_{\alpha\alpha}} - \frac{1}{\lambda_t}\frac{B_{\alpha\alpha}}{G_{\alpha\alpha}}. \tag{3.6.2–7}$$

The transverse shear strains are written as

$$\Delta\gamma_\alpha = \frac{\partial\mathbf{x}}{\partial\theta^\alpha}\cdot\mathbf{n} - \frac{\partial\mathbf{X}}{\partial\theta^\alpha}\cdot\mathbf{N}. \tag{3.6.2–8}$$

This simple form is used because these strains are always assumed to be small. This completes the statement of the incremental strain definitions, and so—together with a virtual work statement

to represent equilibrium—a theory is available. However, it is necessary to satisfy the minimum requirement that the theory provide constant strain under appropriate motions. This is essential if the theory is to be suitable for many practical cases, most especially those involving thermal loading. Interestingly, the theory in Rodal and Witmer (1979) appears to violate this requirement. To achieve this, a modified incremental curvature change measure is defined as

$$\Delta\tilde{k}_{\alpha\beta} = \Delta k_{\alpha\beta} + \mathrm{sym}(\rho^{\gamma}_{\alpha}\Delta\varepsilon_{\gamma\beta}),$$

where

$$\rho^{\beta}_{\alpha} \text{ is a tensor, defined as follows.}$$

We know that the radii of curvature of the α-line at the end and at the beginning of an increment are given by

$$\frac{1}{r_\alpha} = \frac{b_{\alpha\alpha}}{g_{\alpha\alpha}} \text{ at the end of an increment,}$$

and

$$\frac{1}{R_\alpha} = \frac{B_{\alpha\alpha}}{G_{\alpha\alpha}} \text{ at the beginning of the increment.}$$

In these expressions, as in the following development, no summation is implied by a repeated index α. If the α-line is stretched uniformly by $\Delta\lambda_\alpha$ during the increment, we require that

$$r_\alpha = \Delta\lambda_\alpha R_\alpha;$$

and, further, such uniform stretch of the shell must give constant strain so that since we assume

$$\Delta\varepsilon^{1}_{\alpha\alpha} = \Delta\varepsilon_{\alpha\alpha} + \eta\lambda_t\Delta\tilde{k}_{\alpha\alpha},$$

we need

$$\Delta\tilde{k}_{\alpha\alpha} = 0$$

under such circumstances. In this motion

$$\begin{aligned}\Delta k_{\alpha\alpha} &= \frac{1}{r_\alpha} - \frac{1}{\Delta\lambda_t R_\alpha} = \frac{1}{R_\alpha}\left(\frac{1}{\Delta\lambda_\alpha} - \frac{1}{\Delta\lambda_t}\right)\\ &= \frac{1}{R_\alpha}\left(\frac{1}{\Delta\lambda_\alpha} - \frac{1}{\Delta\lambda_t}\right).\end{aligned}$$

Defining

$$\Delta\tilde{k}_{\alpha\alpha} = \Delta k_{\alpha\alpha} + \left(\frac{1}{\Delta\lambda_t} - \frac{1}{\Delta\lambda_\alpha}\right) t^o \frac{B_{\alpha\alpha}}{G_{\alpha\alpha}} \tag{3.6.2–9}$$

and assuming

$$\Delta\varepsilon^1_{\alpha\alpha} = \Delta\varepsilon_{\alpha\alpha} + \eta\lambda_t \Delta\tilde{k}_{\alpha\alpha} \tag{3.6.2–10}$$

satisfies the requirement. Equation 3.6.2–9 may be simplified by substituting in the definition of $\Delta k_{\alpha\alpha}$ in Equation 3.6.2–7 to give

$$\Delta\tilde{k}_{\alpha\alpha} = \frac{b_{\alpha\alpha}}{g_{\alpha\alpha}} - \frac{1}{\Delta\lambda_\alpha}\frac{B_{\alpha\alpha}}{G_{\alpha\alpha}},$$

and so

$$\Delta\tilde{k}_{\alpha\alpha} = \frac{1}{g_{\alpha\alpha}}(b_{\alpha\alpha} - \Delta\lambda_\alpha B_{\alpha\alpha}). \tag{3.6.2–11}$$

The formulation is completed by the assumption that the virtual work equation can be written

$$\int_{\theta^1,\theta^2} \left\{ \lambda_t^o \int_h \sigma^{\alpha\beta}\delta\varepsilon^1_{\alpha\beta}\, d\eta + T^\alpha \delta\gamma_\alpha \right\} (G_{11}G_{22})^{\frac{1}{2}} d\theta^1 d\theta^2 = \delta W^E, \tag{3.6.2–12}$$

where

$\sigma^{\alpha\beta}$ are the Kirchhoff stresses at a point;

(θ^1, θ^2) in the shell, defined by plane stress theory using the summation of the strain increments in Equation 3.6.2–10 to define the strain at this point;

$\delta\varepsilon^1_{\alpha\beta}$ are the variations of the strain increments in Equation 3.6.2–10;

T^α are the transverse shear forces per unit area, defined by $T^\alpha = kGh\gamma_\alpha$; where

- γ_α are the transverse shear strains from Equation 3.6.2–8,
- h is the original thickness of the shell,
- kG is the elastic transverse shear stiffness (reduced according to the suggestions of Hughes et al. (1977) if the shell is too thin, to avoid numerical problems);

and

δW^E is the virtual external work rate.

This completes the statement of the formulation.

Reference

- "Axisymmetric shell element library," Section 29.6.9 of the Abaqus Analysis User's Guide

3.6.3 SHEAR FLEXIBLE SMALL-STRAIN SHELL ELEMENTS

Product: Abaqus/Standard

This section discusses the formulation of the small-strain shear flexible elements in Abaqus/Standard, which are quadrilaterals (S4R5, S8R5, S9R5, and S8R), except for the 6-node triangle STRI65. The essential idea of these elements is that the position of a point in the shell reference surface—$\mathbf{x}$—and the components of a vector $\mathbf{n}$—which is approximately normal to the reference surface—are interpolated independently. The kinematics of the shell theory then consist of measuring membrane strain on the reference surface from the derivatives of $\mathbf{x}$ with respect to position on the surface and bending strain from the derivatives of $\mathbf{n}$; the strain measures that are used for this purpose are approximations to Koiter-Sanders theory strains (Budiansky and Sanders, 1963). The transverse shear strains are measured as the changes in the projections of $\mathbf{n}$ onto tangents to the shell's reference surface. For these element types the strain measures are suitable for large rotations but small strains, and the change in the shell's thickness caused by deformation is neglected.

Notation

A typical piece of shell surface is shown in Figure 3.6.3–1.

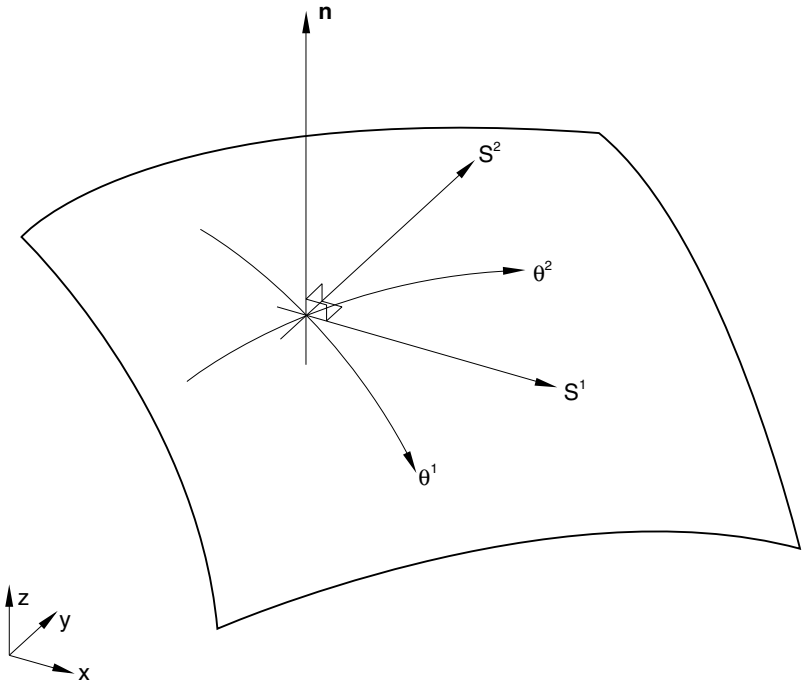


Figure 3.6.3–1 Shell reference surface.

Let $(\theta^1,\ \theta^2)$ be a set of Gaussian surface coordinates on the shell reference surface. Since these coordinates are only needed locally at an integration point, we use the element's isoparametric coordinates as these coordinates. $\mathbf{x}(\theta^1, \theta^2)$ is the current position of a point on the interpolated reference surface, and $\mathbf{X}(\theta^1, \theta^2)$ is the initial position of the same point. The unit vector

$$\mathbf{N} = \frac{\partial \mathbf{X}}{\partial \theta^1} \times \frac{\partial \mathbf{X}}{\partial \theta^2} \Bigg/ \sqrt{\frac{\partial \mathbf{X}}{\partial \theta^1} \times \frac{\partial \mathbf{X}}{\partial \theta^2}}$$

is the unit normal to the interpolated reference surface in the initial configuration. This vector gives a "sidedness" to the surface—one surface of the shell is the "top" surface (in the positive direction along $\mathbf{N}$ from the shell's reference surface) and the other is the bottom surface. The vector corresponding to $\mathbf{N}$ in the current configuration, $\mathbf{n}$, will be made approximately normal to the reference surface in the current configuration by imposing the Kirchhoff constraint discretely.

In the rest of this section Greek indices will be used to indicate values associated with the (two-dimensional) reference surface and so will sum over the range 1, 2 under the summation convention.

First, we establish convenient directions for stress and strain output. These will be local material directions, indistinguishable (to the order of approximation) from corotational directions, since we assume strains are small. The standard convention used throughout Abaqus for such local directions on a surface is as follows.

It is most convenient to choose orthogonal directions. Define

$$\mathbf{T}_2 = \frac{\mathbf{N} \times \mathbf{i}}{\sqrt{\mathbf{N} \times \mathbf{i}}}$$

so long as $N_1 < \cos 0.1^\circ$, where $\mathbf{i}$ is a unit vector in the global X-direction; otherwise,

$$\mathbf{T}_2 = \frac{\mathbf{N} \times \mathbf{k}}{\sqrt{\mathbf{N} \times \mathbf{k}}},$$

where $\mathbf{k}$ is a unit vector in the global Z-direction. Then define

$$\mathbf{T}_1 = \mathbf{T}_2 \times \mathbf{N}.$$

Let

$$dS^\beta = \mathbf{T}_\beta \cdot \frac{\partial \mathbf{X}}{\partial \theta^\alpha} d\theta^\alpha,$$

so that the dS^α are locally defined distance measuring coordinates at each material point. The transformation

$$\frac{\partial}{\partial S^\alpha} = \frac{\partial \theta^\beta}{\partial S^\alpha} \frac{\partial}{\partial \theta^\beta}$$

transforms locally with respect to surface coordinates. Here

$$\frac{\partial \theta^\alpha}{\partial S^\beta} = \begin{bmatrix} \mathbf{T}_1 \cdot \partial \mathbf{X}/\partial \theta^1 & \mathbf{T}_1 \cdot \partial \mathbf{X}/\partial \theta^2 \\ \mathbf{T}_2 \cdot \partial \mathbf{X}/\partial \theta^1 & \mathbf{T}_2 \cdot \partial \mathbf{X}/\partial \theta^2 \end{bmatrix}^{-1}.$$

Stress and strain components are formed in the (dS^1, dS^2) directions.

Surface measures

The following surface measures are defined. The metric of the deformed surface is

$$g_{\alpha\beta} = \frac{\partial \mathbf{x}}{\partial S^\alpha} \cdot \frac{\partial \mathbf{x}}{\partial S^\beta},$$

and an approximation to the curvature tensor (the second fundamental form) is

$$b_{\alpha\beta} = -\frac{1}{2}\left(\frac{\partial \mathbf{n}}{\partial S^\alpha} \cdot \frac{\partial \mathbf{x}}{\partial S^\beta} + \frac{\partial \mathbf{n}}{\partial S^\beta} \cdot \frac{\partial \mathbf{x}}{\partial S^\alpha} \right)$$

(this is only an approximation because $\mathbf{n}$ is not exactly normal to the surface in the current configuration).

The corresponding measures associated with the original reference surface are the metric

$$G_{\alpha\beta} = \frac{\partial \mathbf{X}}{\partial S^\alpha} \cdot \frac{\partial \mathbf{X}}{\partial S^\beta}$$

and the approximation to the curvature

$$B_{\alpha\beta} = -\frac{1}{2}\left(\frac{\partial \mathbf{N}}{\partial S^\alpha} \cdot \frac{\partial \mathbf{X}}{\partial S^\beta} + \frac{\partial \mathbf{N}}{\partial S^\beta} \cdot \frac{\partial \mathbf{X}}{\partial S^\alpha} \right).$$

The vectors $\partial \mathbf{N}/\partial S^\alpha$ are defined from the derivatives of the interpolation functions and the "normals" at the nodes. These nodal normals are calculated as average values of the normals to the surfaces of all elements abutting the node. Abaqus determines if the surface is intended to be smooth at the node (the criterion is that the angle between the normals at the node should be less than 20°). If the surface is not calculated as smooth, separate normals are set up in the different surface branches at the node. Thus, $B_{\alpha\beta}$ should be a reasonable approximation to the second fundamental form of the original reference surface.

Displacements

The nodal variables for shell elements are the displacements of the shell's reference surface, $\mathbf{u} = \mathbf{x} - \mathbf{X}$, and the normal direction, $\mathbf{n}$. Since $\mathbf{n}$ is defined to be a unit vector, only two independent values are needed to define $\mathbf{n}$, so that this type of shell element needs only five degrees of freedom per node. In Abaqus this issue is addressed in two ways. At nodes in a smooth shell surface in those elements that naturally have five degrees of freedom per node, Abaqus stores the values of the projections of the change in $\mathbf{n}$ projected onto two orthogonal directions in the shell surface at the start of the increment to define $\mathbf{n}$. Otherwise, Abaqus stores the usual rotation triplet, $\boldsymbol{\omega}$, at the node. This latter method leaves a redundant degree of freedom if the node is on a smooth surface. A small stiffness is introduced locally at the node to constrain this extra degree of freedom to a measure of the same rotation of the shell's reference surface.

Interpolation

The same bipolynomial interpolation functions are used for all components of $\mathbf{u}$, $\mathbf{X}$, $\mathbf{N}$, and $\mathbf{n}$. The shear flexible shell elements in the library use bilinear interpolation (four nodes), fully biquadratic interpolation (nine nodes), and "serendipity" quadratics (eight nodes).

Strains

The reference surface membrane strains are

$$\epsilon_{\alpha\beta} = \frac{1}{2}(g_{\alpha\beta} - G_{\alpha\beta}).$$

The curvature change is

$$\kappa_{\alpha\beta} = B_{\alpha\beta} - b_{\alpha\beta} + \frac{1}{2}(B^{\gamma}_{\alpha}\epsilon_{\gamma\beta} + B^{\gamma}_{\beta}\epsilon_{\gamma\alpha}).$$

The transverse shears are

$$\gamma_{3\alpha} = \mathbf{n} \cdot \mathbf{t}_{\alpha},$$

where

$$\mathbf{t}_{\alpha} = \frac{\partial \mathbf{x}}{\partial S^{\alpha}} \Bigg/ \sqrt{\frac{\partial \mathbf{x}}{\partial S^{\alpha}} \cdot \frac{\partial \mathbf{x}}{\partial S^{\alpha}}} \quad \text{(no sum on } \alpha\text{)}$$

is a unit vector, tangent to the dS^{α} line in the current surface.

In addition to these strains, when six degrees of freedom are used at the nodes of the elements, the extra rotation degree of freedom is constrained with a penalty, as follows.

When such a node is the corner node of an element, define $\mathbf{T}_1$, $\mathbf{T}_2$, $\mathbf{N}_1$, dS^1, and dS^2 in the element as above. Notice that these will be different in each element at the node, since the interpolated surface is not generally continuous. Then the strain to be penalized is defined as

$$\gamma_{SRC} = \frac{1}{2}(\mathbf{t}_1 \cdot \overline{\mathbf{t}}_2 - \mathbf{t}_2 \cdot \overline{\mathbf{t}}_1),$$

where

$$\overline{\mathbf{t}}_{\alpha} = \mathbf{C} \cdot \mathbf{T}_{\alpha}$$

is the rotated tangent direction, as defined by the rotation values at the node, and

$$\mathbf{t}_{\alpha} = \frac{\partial \mathbf{x}}{\partial S^{\alpha}} \Bigg/ \sqrt{\frac{\partial \mathbf{x}}{\partial S^{\alpha}} \cdot \frac{\partial \mathbf{x}}{\partial S^{\alpha}}} \quad \text{(no sum on } \alpha\text{)}$$

is the rotated tangent direction defined by the motion of the interpolated reference surface at the node.

At each midside node in the original configuration, define $\mathbf{N}$ as the average surface normal for the elements of this surface branch at the nodes (there will be at most two such elements) and $\mathbf{T}$ as the tangent to the edge. Then define

$$\overline{\mathbf{t}} = \mathbf{C} \cdot \mathbf{T} \quad \text{and} \quad \mathbf{n} = \mathbf{C} \cdot \mathbf{N}$$

as rotated values of $\mathbf{T}$ and $\mathbf{N}$, as defined by the rotation values at the node. The vector

$$\mathbf{p} = \overline{\mathbf{t}} \times \mathbf{n}$$

is then normal to $\mathbf{n}$ and to the edge.

The strain to be penalized at these midside nodes is then defined as

$$\gamma_{SRM} = \mathbf{t} \cdot \mathbf{p},$$

where

$$\mathbf{t} = \frac{\partial \mathbf{x}}{\partial S} \Bigg/ \sqrt{\frac{\partial \mathbf{x}}{\partial S} \cdot \frac{\partial \mathbf{x}}{\partial S}}$$

is the tangent to the edge of the element in the current position of the reference surface.

Penalties

The transverse shear strains are calculated at a set of reduced integration points and have the following stiffness associated with them:

$$K_{3\alpha} = G_{3\alpha} h \Delta(\text{area})/(1 + q\Delta(\text{area})/h^2),$$

where the $G_{3\alpha}$ are the elastic moduli associated with transverse shear. The $G_{3\alpha}$ are defined directly by the user or are computed from the elastic moduli given for the layers of the shell. $\boldsymbol{h}$ is the shell thickness; $\Delta(\text{area})$ is the value of reference surface area associated with this integration point in the numerical integration scheme; $\boldsymbol{q}$ is a numerical factor, currently set to $1/4 \times 10^{-4}$. (See Hughes et al., 1977, for a discussion of such factors.) Transverse shears are always treated elastically: nonlinear material calculations in shells are based on plane stress theory, using the membrane and bending strains to define the strain on the surface parallel to the shell's reference surface at each integration point through the shell's thickness.

When rotation constraints are required at nodes that use six degrees of freedom, the penalty used is

$$K = kGh\Delta(\text{area})/(1 + q\Delta(\text{area})/h^2).$$

This is the same as the transverse shear constraint, except that $\Delta(\text{area})$ is here an area "assigned" to the node and the factor $\boldsymbol{k}$ is introduced. This (small) factor has been chosen based on numerical

experiments, to be large enough to avoid singularities yet small enough to avoid adding significantly to the stiffness of the model.

These strain measures, with the interpolation specified above, give zero strain for any general rigid body motion

$$\begin{aligned}\mathbf{x} &= \mathbf{X} + \mathbf{u}_1 + (\mathbf{C}_1 - \mathbf{I}) \cdot (\mathbf{X} - \mathbf{X}_1)\\ \mathbf{n} &= \mathbf{C}_1 \cdot \mathbf{N},\end{aligned}$$

where $\mathbf{u}_1$, $\mathbf{C}_1$, and $\mathbf{X}_1$ are constant.

First variations of strain

The first variations of the strains are

$$\begin{aligned}\delta\epsilon_{\alpha\beta} &= \frac{1}{2}\delta g_{\alpha\beta}\\ \delta\kappa_{\alpha\beta} &= -\delta b_{\alpha\beta} + \frac{1}{2}(B^\gamma_\alpha \delta\epsilon_{\gamma\beta} + B^\gamma_\beta \delta\epsilon_{\gamma\alpha})\\ \delta\gamma_{3\alpha} &= \delta\mathbf{n}\cdot\mathbf{t}_\alpha + \mathbf{n}\cdot\delta\mathbf{t}_\alpha\\ \delta\gamma_{SRC} &= \frac{1}{2}(\delta\mathbf{t}_1\cdot\overline{\mathbf{t}}_2 - \delta\mathbf{t}_2\cdot\overline{\mathbf{t}}_1 + \mathbf{t}_1\cdot\delta\overline{\mathbf{t}}_2 - \mathbf{t}_2\cdot\delta\overline{\mathbf{t}}_1)\\ \delta\gamma_{SRM} &= \delta\overline{\mathbf{t}}_1\cdot\mathbf{p} + \delta\mathbf{p}\cdot\overline{\mathbf{t}},\end{aligned}$$

where

$$\begin{aligned}\delta g_{\alpha\beta} &= \frac{\partial\delta\mathbf{u}}{\partial S^\alpha}\cdot\frac{\partial\mathbf{x}}{\partial S^\beta} + \frac{\partial\mathbf{x}}{\partial S^\alpha}\cdot\frac{\partial\delta\mathbf{u}}{\partial S^\beta}\\ \delta b_{\alpha\beta} &= -\frac{1}{2}\left(\frac{\partial\delta\mathbf{n}}{\partial S^\alpha}\cdot\frac{\partial\mathbf{x}}{\partial S^\beta} + \frac{\partial\delta\mathbf{n}}{\partial S^\beta}\cdot\frac{\partial\mathbf{x}}{\partial S^\alpha} + \frac{\partial\mathbf{n}}{S^\alpha}\frac{\partial\delta\mathbf{u}}{\partial S^\beta} + \frac{\partial\mathbf{n}}{\partial S^\beta}\cdot\frac{\partial\delta\mathbf{u}}{\partial S^\alpha}\right)\\ \delta\mathbf{n} &= \delta\boldsymbol{\omega}\times\mathbf{n}\\ \delta\overline{\mathbf{t}}_\alpha &= \delta\boldsymbol{\omega}\times\overline{\mathbf{t}}_\alpha\\ \delta\mathbf{t}_\alpha &= \left(\frac{\partial\mathbf{x}}{\partial S^\alpha}\cdot\frac{\partial\mathbf{x}}{\partial S^\alpha}\right)^{-\frac{1}{2}}[\mathbf{I} - \mathbf{t}_\alpha\,\mathbf{t}_\alpha]\cdot\frac{\partial\delta\mathbf{u}}{\partial S^\alpha}\quad(\text{no sum on}\alpha)\end{aligned}$$

and at the midside nodes

$$\delta\mathbf{p} = \delta\boldsymbol{\omega}\times\mathbf{p}.$$

Second variations of strains

In forming the initial stress matrix we approximate by neglecting $d\delta\mathbf{n}$, $d\delta\mathbf{t}$, etc, to simplify the expressions and reduce the cost of forming the matrix. Numerical experiments have suggested that,

at least for the problems tested, this does not significantly affect the convergence rate. With this approximation,

$$d\delta\epsilon_{\alpha\beta} = \frac{1}{2}\left(\frac{\partial\delta\mathbf{u}}{\partial S^\alpha}\cdot\frac{\partial d\mathbf{u}}{\partial S^\beta} + \frac{\partial\delta\mathbf{u}}{\partial S^\beta}\cdot\frac{\partial d\mathbf{u}}{\partial S^\alpha}\right)$$

$$\begin{aligned} d\delta\kappa_{\alpha\beta} =& \frac{1}{2}\left(\frac{\partial\boldsymbol{\omega}}{\partial S^\alpha}\times\mathbf{n}\cdot\frac{\partial d\mathbf{u}}{\partial S^\beta} + \delta\boldsymbol{\omega}\times\frac{\partial\mathbf{n}}{\partial S^\alpha}\cdot\frac{\partial d\mathbf{u}}{\partial S^\beta}\right.\\ &+ \frac{\partial\boldsymbol{\omega}}{\partial S^\beta}\times\mathbf{n}\cdot\frac{\partial d\mathbf{u}}{\partial S^\alpha} + \delta\boldsymbol{\omega}\times\frac{\partial\mathbf{n}}{\partial S^\beta}\cdot\frac{\partial d\mathbf{u}}{\partial S^\alpha}\\ &+ \frac{\partial\boldsymbol{\omega}}{\partial S^\alpha}\times\mathbf{n}\cdot\frac{\partial\delta\mathbf{u}}{\partial S^\beta} + d\boldsymbol{\omega}\times\frac{\partial\mathbf{n}}{\partial S^\alpha}\cdot\frac{\partial\delta\mathbf{u}}{\partial S^\beta}\\ &\left.+ \left(\frac{\partial\boldsymbol{\omega}}{\partial S^\beta}\times\mathbf{n}\cdot\frac{\partial\delta\mathbf{u}}{\partial S^\alpha} + d\boldsymbol{\omega}\times\frac{\partial\mathbf{n}}{\partial S^\beta}\cdot\frac{\partial\mathbf{u}}{\partial S^\alpha}\right)\right.\\ &+ \frac{1}{2}(B^\gamma_\alpha\, d\delta\epsilon_{\gamma\beta} + B^\gamma_\beta\, d\delta\epsilon_{\gamma\alpha}) \end{aligned}$$

$$\begin{aligned} d\gamma_{3\alpha} =& g_\alpha^{-\frac{1}{2}}\delta\boldsymbol{\omega}\times\mathbf{n}\cdot(\mathbf{I}-\mathbf{t}_\alpha\,\mathbf{t}_\alpha)\cdot\frac{\partial d\mathbf{u}}{\partial S^\alpha}\\ &+ g_\alpha^{-\frac{1}{2}}d\boldsymbol{\omega}\times\mathbf{n}\cdot(\mathbf{I}-\mathbf{t}_\alpha\,\mathbf{t}_\alpha)\cdot\frac{\partial\mathbf{u}}{\partial S^\alpha} \quad \text{(no sum on } \alpha) \end{aligned}$$

$$\begin{aligned} d\delta\gamma_{SRC} =& g_1^{-\frac{1}{2}}\delta\boldsymbol{\omega}\times\overline{\mathbf{t}}_1\cdot(\mathbf{I}-\mathbf{t}_2\,\mathbf{t}_2)\cdot\frac{\partial d\mathbf{u}}{\partial S^2}\\ &- g_2^{-\frac{1}{2}}\delta\boldsymbol{\omega}\times\overline{\mathbf{t}}_2\cdot(\mathbf{I}-\mathbf{t}_1\,\mathbf{t}_1)\cdot\frac{\partial d\mathbf{u}}{\partial S^1}\\ &+ g_1^{-\frac{1}{2}}d\boldsymbol{\omega}\times\overline{\mathbf{t}}_1\cdot(\mathbf{I}-\mathbf{t}_2\,\mathbf{t}_2)\cdot\frac{\partial\delta\mathbf{u}}{\partial S^2}\\ &- g_2^{-\frac{1}{2}}d\boldsymbol{\omega}\times\overline{\mathbf{t}}_2\cdot(\mathbf{I}-\mathbf{t}_1\,\mathbf{t}_1)\cdot\frac{\partial\delta\mathbf{u}}{\partial S^1} \end{aligned}$$

and

$$\begin{aligned} d\delta\gamma_{SRM} =& g^{-\frac{1}{2}}d\boldsymbol{\omega}\times\mathbf{p}\cdot(\mathbf{I}-\mathbf{t}\,\mathbf{t})\cdot\frac{\partial\delta\mathbf{u}}{\partial S^2}\\ &+ g^{-\frac{1}{2}}\delta\boldsymbol{\omega}\times\mathbf{p}\cdot(\mathbf{I}-\mathbf{t}\,\mathbf{t})\cdot\frac{\partial d\mathbf{u}}{\partial S^1}. \end{aligned}$$

Internal virtual work rate

For these shell elements the internal virtual work rate is assumed to be

$$\begin{aligned}\delta W^I = &\int_A \int_h \sigma^{\alpha\beta} \delta\varepsilon^f_{\alpha\beta}\, dz dA \\ &+ \sum_r K_I^{(r)} \gamma_{3\alpha}^{(r)} \delta\gamma_{3\alpha}^{(r)} \\ &+ \sum_{n^c} K_{II}^{(n)} \gamma_{SRC}^{(n)} \delta\gamma_{SRC}^{(n)} \\ &+ \sum_{n^m} K_{III}^{(n)} \gamma_{SRM}^{(n)} \delta\gamma_{SRM}^{(n)},\end{aligned}$$

where $K_I^{(r)}$, $K_{II}^{(n)}$, and $K_{III}^{(n)}$ are the transverse shear stiffness and the penalties defined above and r indicates the integration points at which transverse shears are calculated, n^c indicates corner nodes at which six degrees of freedom are used, and n^m indicates midside nodes at which six degrees of freedom are used. Here $\varepsilon^f_{\alpha\beta}$ and $\sigma^{\alpha\beta}$ are the strain and stress in the (dS^α, dS^β) material directions in a surface offset by a distance z from the reference surface. The usual Kirchhoff assumption is adopted:

$$\varepsilon^f_{\alpha\beta} = \epsilon_{\alpha\beta} + z\kappa_{\alpha\beta},$$

so that the first term above is

$$\int_A \left(\delta\epsilon_{\alpha\beta} \int_h \sigma^{\alpha\beta}\, dz + \delta\kappa_{\alpha\beta} \int_h z\sigma^{\alpha\beta}\, dz \right) dA.$$

The thickness direction integrations are performed numerically in Abaqus. The integration scheme is a Simpson's rule, of user-chosen order. The shell can also be considered layered, with different properties at each layer and a different integration scheme assigned (by the user) to each layer.

Pressure load stiffness

The load stiffness associated with pressure loading is often important in shells, especially in eigenvalue buckling estimates on elastic shells. In Abaqus/Standard the pressure load stiffness is implemented as a symmetric form, thus assuming that the pressure magnitude is constant over the surface and neglecting free edge effects. See Hibbitt (1979) and Mang (1980) for details.

The load stiffness is obtained in such a form as follows. The external virtual work associated with pressure is

$$\delta W^e = \int_A \mathbf{p} \cdot \delta\mathbf{u}\, dA,$$

where $\mathbf{p}$ is the pressure load per unit area, given in terms of the (externally prescribed) pressure magnitude, p, as

$$\mathbf{p}\, dA = p \frac{\partial \mathbf{x}}{\partial \theta^1} \times \frac{\partial \mathbf{x}}{\partial \theta^2}\, d\theta^1 d\theta^2.$$

Thus,

$$\delta W^e = \int_A p \left(\frac{\partial \mathbf{x}}{\partial \theta^1} \times \frac{\partial \mathbf{x}}{\partial \theta^1} \right) \cdot \delta \mathbf{u} \, d\theta^1 d\theta^2.$$

The change in this term caused by change of displacement of the shell (the "load stiffness") is

$$d\delta W^e = \int_A p \, \delta \mathbf{u} \cdot \left(\frac{\partial \mathbf{x}}{\partial \theta^1} \times \frac{\partial d\mathbf{u}}{\partial \theta^2} - \frac{\partial \mathbf{x}}{\partial \theta^2} \times \frac{\partial d\mathbf{u}}{\partial \theta^1} \right) d\theta^1 d\theta^2,$$

since we assume that pressure magnitude, $\boldsymbol{p}$, is externally prescribed and has no dependence on position, $\mathbf{x}$. Neglecting free edge effects, and assuming the magnitude $\boldsymbol{p}$ is uniform, results in the symmetric form

$$\begin{aligned} d\delta W^e = \int_A \frac{1}{2} p \Bigg[& \frac{\partial \mathbf{x}}{\partial \theta^1} \cdot \left(\frac{\partial d\mathbf{u}}{\partial \theta^2} \times \delta \mathbf{u} + \frac{\partial \delta \mathbf{u}}{\partial \theta^2} \times d\mathbf{u} \right) \\ & - \frac{\partial \mathbf{x}}{\partial \theta^2} \cdot \left(\frac{\partial d\mathbf{u}}{\partial \theta^1} \times \delta \mathbf{u} + \frac{\partial \delta \mathbf{u}}{\partial \theta^1} \times d\mathbf{u} \right) \Bigg] d\theta^1 d\theta^2. \end{aligned}$$

This is the pressure load stiffness provided in Abaqus.

Reference

- "Shell elements: overview," Section 29.6.1 of the Abaqus Analysis User's Guide

3.6.4 TRIANGULAR FACET SHELL ELEMENTS

Product: Abaqus/Standard

Element type STRI3 in Abaqus/Standard is a facet shell—a plate element used to approximate a shell. The element has three nodes, each with six degrees of freedom. The strains are based on thin plate theory, using a small-strain approximation. Arbitrary rigid body rotations are accounted for exactly by formulating the deformation of the element in a local coordinate system that rotates with the element. The element also satisfies the patch test, so that it will produce reliable results with appropriate meshes.

The bending of the element is based on a discrete Kirchhoff approach to plate bending, using Batoz's interpolation functions (Batoz et al., 1980). This formulation satisfies the Kirchhoff constraints all around the boundary of the triangle and provides linear variation of curvature throughout the element. However, the membrane strains are assumed constant within the element. In addition, a curved shell is approximated by this element as a set of facets formed by the planes defined by the three nodes of each element. For these reasons it is necessary to use a reasonably well refined mesh in most applications.

Kinematics

A local orthonormal basis system, $\mathbf{T}^1$ and $\mathbf{T}^2$, is defined in the plane of each element in the reference configuration, using the standard Abaqus convention. S^1 and S^2 measure distance along $\mathbf{T}^1$ and $\mathbf{T}^2$ in the reference configuration.

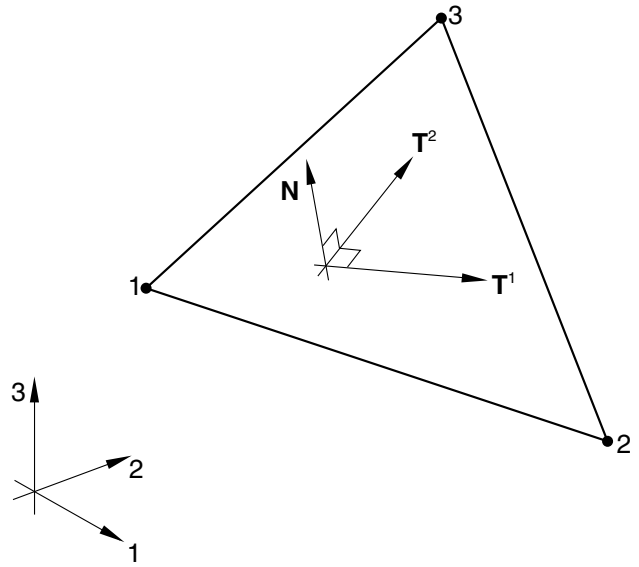


Figure 3.6.4–1 Triangular facet shell in the reference configuration.

The membrane strains are then defined as

$$\varepsilon_{\alpha\beta} = \frac{1}{2}(g_{\alpha\beta} - G_{\alpha\beta}), \quad \alpha \text{ and } \beta = 1, 2$$

where

$$g_{\alpha\beta} = \frac{\partial \mathbf{x}}{\partial S^\alpha} \cdot \frac{\partial \mathbf{x}}{\partial S^\beta}$$

is the metric in the current configuration, and

$$G_{\alpha\beta} = \frac{\partial \mathbf{X}}{\partial S^\alpha} \cdot \frac{\partial \mathbf{X}}{\partial S^\beta}$$

is the metric in the reference configuration.

Here $\mathbf{x}$ and $\mathbf{X}$ are the spatial coordinates of a point in the current and reference configurations, respectively. Curvature changes are defined incrementally. To account for large rigid body rotations we use a local coordinate system that rotates with the plane defined by the three nodes of the element. The basis vectors chosen for this local system are $\mathbf{t}^1 = \partial \mathbf{x}/\partial S^1$ and $\mathbf{t}^2 = \partial \mathbf{x}/\partial S^2$. Since the membrane strains are assumed to be small, these vectors will be approximately orthonormal. The components of incremental rotation of the normal to the plate are defined as $\Delta\omega_1$ about $\mathbf{t}^1$ and $\Delta\omega_2$ about $\mathbf{t}^2$. The incremental displacement of the reference surface of the plate along the normal to the plane of its nodes is defined as Δw. (Note that Δw will be zero at the nodes at all times because the plane containing $\mathbf{t}^1$ and $\mathbf{t}^2$ always passes through the nodes.) The Kirchhoff constraints are, approximately,

$$-\Delta\omega_2 + \frac{\partial \Delta w}{\partial S^1} = 0$$

and

$$\Delta\omega_1 + \frac{\partial \Delta w}{\partial S^2} = 0.$$

Batoz (1980) assumes that $\Delta\omega_1$ and $\Delta\omega_2$ vary quadratically over the element and that Δw is defined independently along each of the three sides of the element as a cubic function. The Kirchhoff constraints are then imposed at the corners and at the middle of each element edge along the direction of the edge to give

$$\Delta\omega_1 = H_1^P \Delta\nu^P$$

and

$$\Delta\omega_2 = H_2^P \Delta\nu^P,$$

where $\Delta\nu^P$ is the array

$$\left\lfloor \Delta\theta_1^1,\ \Delta\theta_2^1,\ \Delta\theta_1^2,\ \Delta\theta_2^2,\ \Delta\theta_1^3,\ \Delta\theta_2^3 \right\rfloor .$$

In the above expressions $H_1^P(g,h)$ and $H_2^P(g,h)$ are interpolation functions that are defined by Batoz (1980), and the incremental rotation components at the nodes, $\Delta\theta_\alpha^N$, are defined as

$$\Delta\theta_\alpha^N = \mathbf{n}^N \cdot \mathbf{t}^\alpha,$$

where

$$\mathbf{n}^N = \mathbf{C}(\Delta\phi_i^N) \cdot \mathbf{N},$$

and $\Delta\phi_i^N$ are the increments of the rotational degrees of freedom at the node N, $\mathbf{C}$ is the rotation matrix defined by $\Delta\phi_i^N$, and $\mathbf{N}$ is the normal to the plane of the element's nodes at the beginning of the increment. Finally, the incremental curvature change measures are defined as

$$\Delta k_{11} = \frac{\partial \Delta\omega_2}{\partial S^1}$$
$$\Delta k_{22} = -\frac{\partial \Delta\omega_1}{\partial S^2}$$
and
$$\Delta k_{12} = -\frac{\partial \Delta\omega_1}{\partial S^1} + \frac{\partial \Delta\omega_2}{\partial S^2}.$$

The three membrane strains and three curvature strains complete the basic kinematic description of the element, except that the use of six degrees of freedom per node introduces a spurious rotation at each node (only two incremental rotations at each node appear in the above equations—the rotation about the normal to the plane of the element's nodes does not enter). To deal with this problem, we define a generalized strain to be penalized with a small stiffness at each node as

$$\gamma^i = \mathbf{b}_1 \cdot \mathbf{a}_1 - \mathbf{b}_2 \cdot \mathbf{a}_2,$$

where

$$\mathbf{a}_1 = \mathbf{C} \cdot \mathbf{A}_1$$
$$\mathbf{a}_2 = \mathbf{C} \cdot \mathbf{A}_2$$
$$\mathbf{A}_1 = \mathbf{N} \times \overline{\mathbf{A}}_1$$
$$\mathbf{A}_2 = \overline{\mathbf{A}}_2 \times \mathbf{N}$$
$$\overline{\mathbf{A}}_1 = (\mathbf{X}_j - \mathbf{X}_i)/|\mathbf{X}_j - \mathbf{X}_i|$$
$$\overline{\mathbf{A}}_2 = (\mathbf{X}_k - \mathbf{X}_i)/|\mathbf{X}_k - \mathbf{X}_i|$$
$$\mathbf{b_1} = (\mathbf{x}_j - \mathbf{x}_i)/|\mathbf{x}_j - \mathbf{x}_i|$$
$$\mathbf{b_2} = (\mathbf{x}_k - \mathbf{x}_i)/|\mathbf{x}_k - \mathbf{x}_i|$$

and j, k are the node numbers in cyclic order forming the two sides of the triangle at the node i.

First variations of strain

The first variations of strain are

$$\begin{aligned}
\delta\varepsilon_{\alpha\beta} &= \frac{1}{2}\left(\frac{\partial \mathbf{x}}{\partial S^\alpha}\cdot\frac{\partial \delta\mathbf{x}}{\partial \mathbf{S}^\beta}+\frac{\partial \mathbf{x}}{\partial S^\beta}\cdot\frac{\partial \delta\mathbf{x}}{\partial S^\alpha}\right)\\
\delta K_{11} &= \frac{\partial(\delta\Delta\omega_2)}{\partial S^1}\\
\delta K_{22} &= -\frac{\partial(\delta\Delta\omega_1)}{\partial S^2}\\
\delta K_{12} &= \left(\frac{\partial(\delta\Delta\omega_2)}{\partial S^2}-\frac{\partial(\delta\Delta\omega_1)}{\partial S^1}\right),
\end{aligned}$$

where

$$\begin{aligned}
\delta\Delta\omega_1 &= H_1^P\delta\Delta\nu^P\\
\delta\Delta\omega_2 &= H_2^P\delta\Delta\nu^P,
\end{aligned}$$

and in $\Delta\delta\nu^P$,

$$\delta\Delta\theta_\alpha^N = \delta\boldsymbol{\phi}^N\times\mathbf{n}^N\cdot\mathbf{t}^\alpha+\mathbf{n}^N\cdot\delta\mathbf{t}^\alpha.$$

Also, for the "strain" used to introduce the extra stiffness at the nodes to avoid singularity caused by the component of rotation about the normal,

$$\delta\gamma^i = \delta\mathbf{b}_1\cdot\mathbf{a}_1-\delta\mathbf{b}_2\cdot\mathbf{a}_2+\mathbf{b}_1\cdot\delta\boldsymbol{\omega}\times\mathbf{a}_1-\mathbf{b}_2\cdot\delta\boldsymbol{\omega}\times\mathbf{a}_2.$$

Second variations of strain

The second variations of strain are

$$\begin{aligned}
d\delta\varepsilon_{\alpha\beta} &= \frac{1}{2}\left(\frac{\partial \delta\mathbf{x}}{\partial S^\alpha}\cdot\frac{\partial d\mathbf{x}}{\partial S^\beta}+\frac{\partial \delta\mathbf{x}}{\partial S^\beta}\cdot\frac{\partial d\mathbf{x}}{\partial S^\alpha}\right)\\
d\delta K_{11} &= \frac{\partial d\delta\omega_2}{\partial S^1}\\
d\delta K_{22} &= -\frac{\partial(d\delta\omega_1)}{\partial S^2}\\
d\delta K_{12} &= \left(\frac{\partial d\delta\omega_2}{\partial S^2}-\frac{\partial d\delta\omega_1}{\partial S^1}\right),
\end{aligned}$$

where

$$
\begin{aligned}
d\Delta\delta\theta_\alpha^N = & -\delta\boldsymbol{\omega}^N \cdot d\boldsymbol{\omega}^N \, \mathbf{n}^N \cdot \mathbf{t}^\alpha + \delta\boldsymbol{\omega}^N \cdot \mathbf{n}^N \, d\boldsymbol{\omega}^N \cdot \mathbf{t}^\alpha \\
& + \delta\boldsymbol{\omega}^N \cdot \mathbf{n}^N \times \left(\frac{\partial g^1}{\partial S^1}(d\mathbf{x}^2 - d\mathbf{x}^1) + \frac{\partial g^2}{\partial S^1}(d\mathbf{x}^3 - d\mathbf{x}^1) \right) \\
& - \left(\frac{\partial g^1}{\partial S^1}(\delta\mathbf{x}^2 - \delta\mathbf{x}^1) + \frac{\partial g^2}{\partial S^1}(\delta\mathbf{x}^3 - \delta\mathbf{x}^1) \right) \times \mathbf{n}^N \cdot d\boldsymbol{\omega}^N
\end{aligned}
$$

and

$$
\begin{aligned}
d\delta\gamma^i = & \, \delta\mathbf{b}_i \cdot d\mathbf{a}_1 - \delta\mathbf{b}_2 \cdot d\mathbf{a}_2 + d\mathbf{b}_1 \cdot \delta\boldsymbol{\omega} \times \mathbf{a}_1 + \mathbf{b}_1 \cdot \delta\boldsymbol{\omega} \times d\mathbf{a}_1 \\
& - d\mathbf{b}_2 \cdot \delta\boldsymbol{\omega} \times \mathbf{a}_2 - b_2 \cdot \delta\boldsymbol{\omega} \times d\mathbf{a}_2 .
\end{aligned}
$$

Here g^1 and g^2 are coordinates in the plane of the element, normalized so that the nodes of the element are at (0,0), (1,0) and (0,1).

Internal virtual work rate

The internal virtual work rate is defined as

$$
\delta W^I = \int_A \int_h \sigma^{\alpha\beta} \delta\varepsilon^f_{\alpha\beta} \, dz \, dA + \sum_{\text{nodes}(i)} K^i \, \gamma^i \, \delta\gamma^i ,
$$

where $\varepsilon^f_{\alpha\beta} = \varepsilon_{\alpha\beta} + z k_{\alpha\beta}$ is the strain at a point, $\boldsymbol{f}$, away from the reference surface; $\sigma^{\alpha\beta}$ are the stress components at $\boldsymbol{f}$; $\boldsymbol{h}$ is the shell thickness; and K^i is the penalty stiffness used to constrain the spurious rotation.

The formulation now proceeds as for the shell elements described in “Shear flexible small-strain shell elements,” Section 3.6.3, using a 3-point integration scheme in the plane of the element.

Reference

- “Shell elements: overview,” Section 29.6.1 of the Abaqus Analysis User’s Guide

3.6.5 FINITE-STRAIN SHELL ELEMENT FORMULATION

Products: Abaqus/Standard Abaqus/Explicit

This section describes the formulation of the quadrilateral finite-membrane-strain element S4R, the triangular element S3R and S3 obtained through degeneration of S4R, and the fully integrated finite-membrane-strain element S4.

Geometric description

At a given stage in the deformation history of the shell, the position of a material point in the shell is defined by

$$\mathbf{x}(S_i) = \overline{\mathbf{x}}(S_\alpha) + \overline{f}_{33}(S_\alpha)\mathbf{t}_3(S_\alpha)S_3,$$

where the subscript i and other Roman subscripts range from 1 to 3. Subscripts α and other lowercase Greek subscripts which describe the quantities in the reference surface of the shell range from 1 to 2. In the above equation $\mathbf{t}_3$ is the normal to the reference surface of the shell. The gradient of the position is

$$\frac{\partial \mathbf{x}}{\partial S_\beta} = \frac{\partial \overline{\mathbf{x}}}{\partial S_\beta} + \overline{f}_{33}\frac{\partial \mathbf{t}_3}{\partial S_\beta}S_3, \qquad \frac{\partial \mathbf{x}}{\partial S_3} = \overline{f}_{33}\mathbf{t}_3,$$

where we have neglected derivatives of $\overline{f}_{33}$ with respect to S_β. Note that in the above S_α are local surface coordinates that are assumed to be orthogonal and distance measuring in the reference state. S_3 is the coordinate in the thickness direction, distance measuring and orthogonal to S_α in the reference state. The thickness increase factor $\overline{f}_{33}$ is assumed to be independent of S_3.

In the deformed state we define local, orthonormal shell directions $\mathbf{t}_i$ such that

$$\mathbf{t}_i \cdot \mathbf{t}_j = \delta_{ij}, \qquad \mathbf{t}_i\mathbf{t}_i = \mathbf{I},$$

where δ_{ij} is the Kronecker delta and $\mathbf{I}$ is the identity tensor of rank 2. Summation convention is used for repeated subscripts. The in-plane components of the gradient of the position are obtained as

$$f_{\alpha\beta} = \mathbf{t}_\alpha \cdot \frac{\partial \mathbf{x}}{\partial S_\beta} = \overline{f}_{\alpha\beta} + B_{\alpha\beta}\overline{f}_{33}S_3,$$

where we have introduced the reference surface deformation gradient

$$\overline{f}_{\alpha\beta} \stackrel{\text{def}}{=} \mathbf{t}_\alpha \cdot \left.\frac{\partial \mathbf{x}}{\partial S_\beta}\right|_{S_3=0} = \mathbf{t}_\alpha \cdot \frac{\partial \overline{\mathbf{x}}}{\partial S_\beta}$$

and the reference surface normal gradient

$$B_{\alpha\beta} \stackrel{\text{def}}{=} \mathbf{t}_\alpha \cdot \frac{\partial \mathbf{t}_3}{\partial S_\beta}.$$

In the original (reference) configuration we denote the position by $\mathbf{X}$ ($\overline{\mathbf{X}}$ for the reference surface) and the direction vectors by $\mathbf{T}_i$, which yields

$$\mathbf{X}(S_i) = \overline{\mathbf{X}}(S_\alpha) + \mathbf{T}_3(S_\alpha) S_3.$$

The gradient of the position is

$$\frac{\partial \mathbf{X}}{\partial S_\beta} = \frac{\partial \overline{\mathbf{X}}}{\partial S_\beta} + \frac{\partial \mathbf{T}_3}{\partial S_\beta} S_3, \qquad \frac{\partial \mathbf{X}}{\partial S_3} = \mathbf{T}_3,$$

and the in-plane components of the gradient are obtained as

$$f^\circ_{\alpha\beta} = \mathbf{T}_\alpha \cdot \frac{\partial \mathbf{X}}{\partial S_\beta} = \delta_{\alpha\beta} + B^\circ_{\alpha\beta} S_3,$$

where we have assumed that the in-plane direction vectors follow from the surface coordinates with

$$\mathbf{T}_\beta = \left. \frac{\partial \mathbf{X}}{\partial S_\beta} \right|_{S_3 = 0} = \frac{\partial \overline{\mathbf{X}}}{\partial S_\beta}$$

and defined the original reference surface normal gradient,

$$B^\circ_{\alpha\beta} \stackrel{\text{def}}{=} \mathbf{T}_\alpha \cdot \frac{\partial \mathbf{T}_3}{\partial S_\beta}.$$

The original reference surface normal gradient is obtained in the finite element formulation from the interpolation of the nodal normals with the shape functions. In the deformed configuration it is not derived from the nodal normals but is updated independently based on the gradient of the incremental rotations.

Parametric interpolation

The position of the points in the shell reference surface is described in terms of discrete nodal positions with parametric interpolation functions $N^I(\xi_\alpha)$. The functions are $C_\circ$ continuous, and ξ_α are nonorthogonal, nondistance measuring parametric coordinates. For the reference surface positions one, thus, obtains

$$\overline{\mathbf{x}}(\xi_\alpha) = N^I(\xi_\alpha) \overline{\mathbf{x}}^I, \qquad \overline{\mathbf{X}}(\xi_\alpha) = N^I(\xi_\alpha) \overline{\mathbf{X}}^I.$$

The gradients of the position with respect to ξ_β are

$$\frac{\partial \overline{\mathbf{x}}}{\partial \xi_\beta} = \frac{\partial N^I}{\partial \xi_\beta} \overline{\mathbf{x}}^I, \qquad \frac{\partial \overline{\mathbf{X}}}{\partial \xi_\beta} = \frac{\partial N^I}{\partial \xi_\beta} \overline{\mathbf{X}}^I.$$

Note that uppercase Roman superscripts such as I denote nodes of an element and that repeated superscripts imply summation over all nodes of an element.

Now consider the original configuration. The unit normal to the shell reference surface is readily obtained as

$$\mathbf{T}_3 = \left(\frac{\partial \overline{\mathbf{X}}}{\partial \xi_1} \times \frac{\partial \overline{\mathbf{X}}}{\partial \xi_2} \right) \Big/ \left| \frac{\partial \overline{\mathbf{X}}}{\partial \xi_1} \times \frac{\partial \overline{\mathbf{X}}}{\partial \xi_2} \right|.$$

Subsequently, we define two orthonormal tangent vectors $\mathbf{T}_\alpha$ and distance measuring coordinates S_α along these vectors. The derivatives of these coordinates with respect to ξ_β follow from

$$\frac{\partial S_\alpha}{\partial \xi_\beta} = \mathbf{T}_\alpha \cdot \frac{\partial \overline{\mathbf{X}}}{\partial \xi_\beta} = \mathbf{T}_\alpha \cdot \overline{\mathbf{X}}^I \frac{\partial N^I}{\partial \xi_\beta}.$$

The gradient of ξ_α with respect to S_β is readily obtained by inversion:

$$\frac{\partial \xi_\alpha}{\partial S_\beta} = \left[\frac{\partial S_\alpha}{\partial \xi_\beta} \right]^{-1},$$

which makes it possible to obtain the gradient operator

$$\frac{\partial N^I}{\partial S_\beta} = \frac{\partial N^I}{\partial \xi_\alpha} \frac{\partial \xi_\alpha}{\partial S_\beta}.$$

The original reference surface normal gradient is obtained from the nodal normals $\mathbf{T}_3^I$ with

$$B^\circ_{\alpha\beta} = \mathbf{T}_\alpha \cdot \mathbf{T}_3^I \frac{\partial N^I}{\partial S_\beta}.$$

Since the original reference surface normal gradient is obtained by taking derivatives with respect to orthogonal distance measuring coordinates, we will call $B^\circ_{\alpha\beta} = b^\circ_{\alpha\beta}$ the original *curvature* of the reference surface.

Membrane deformation and curvature

It is convenient to define the inverse of the reference surface deformation gradient

$$\overline{h}_{\alpha\beta} = \left[\overline{f}_{\alpha\beta} \right]^{-1}.$$

With this expression we can define the gradient operator in the current state:

$$\frac{\partial}{\partial s_\beta} \stackrel{\text{def}}{=} \overline{h}_{\alpha\beta} \frac{\partial}{\partial S_\alpha}, \quad \text{or inverted} \quad \frac{\partial}{\partial S_\beta} = \overline{f}_{\alpha\beta} \frac{\partial}{\partial s_\alpha}.$$

The gradient operator in the current state can also be defined as the derivative with respect to distance measuring coordinates s_α along the base vectors $\mathbf{t}_\alpha$, since

$$\mathbf{t}_\alpha \cdot \frac{\partial \overline{\mathbf{x}}}{\partial s_\beta} = \mathbf{t}_\alpha \cdot \frac{\partial \overline{\mathbf{x}}}{\partial S_\gamma} \overline{h}_{\gamma\beta} = \overline{f}_{\alpha\gamma} \overline{h}_{\gamma\beta} = \delta_{\alpha\beta}$$

and, hence,

$$\mathbf{t}_\alpha = \frac{\partial \overline{\mathbf{x}}}{\partial s_\alpha}.$$

Hence, it is possible to write for the $\overline{h}_{\alpha\beta}$:

$$\overline{h}_{\alpha\beta} = \mathbf{T}_\alpha \cdot \frac{\partial \overline{\mathbf{X}}}{\partial s_\beta}$$

since

$$\overline{f}_{\alpha\gamma} \overline{h}_{\gamma\beta} = \mathbf{t}_\alpha \cdot \frac{\partial \overline{\mathbf{x}}}{\partial \overline{\mathbf{X}}} \cdot \mathbf{T}_\gamma \mathbf{T}_\gamma \frac{\partial \overline{\mathbf{X}}}{\partial \overline{\mathbf{x}}} \cdot \mathbf{t}_\beta = \mathbf{t}_\alpha \cdot \frac{\partial \overline{\mathbf{x}}}{\partial \overline{\mathbf{X}}} \cdot \frac{\partial \overline{\mathbf{X}}}{\partial \overline{\mathbf{x}}} \cdot \mathbf{t}_\beta = \mathbf{t}_\alpha \cdot \mathbf{t}_\beta = \delta_{\alpha\beta}.$$

In an incremental analysis we can also define the incremental deformation tensor

$$\Delta \overline{f}_{\alpha\beta} = \mathbf{t}_\alpha^{t+\Delta t} \cdot \frac{\partial \overline{\mathbf{x}}^{t+\Delta t}}{\partial s_\beta^t}$$

and its inverse

$$\Delta \overline{h}_{\alpha\beta} = \mathbf{t}_\alpha^t \cdot \frac{\partial \overline{\mathbf{x}}^t}{\partial s_\beta^{t+\Delta t}}.$$

With a local coordinate system defined in the current state, the current gradient of the normal can be transformed into the curvature of the surface:

$$b_{\alpha\beta} \overset{\text{def}}{=} \mathbf{t}_\alpha \cdot \frac{\partial \mathbf{t}_3}{\partial s_\beta} = B_{\alpha\gamma} \overline{h}_{\gamma\beta}.$$

Orientation update

The equations given in the earlier sections are valid for any local coordinate system defined in the current state. The $\mathbf{t}_\alpha$ vectors at the beginning of the analysis are determined following the standard Abaqus conventions. In this section, we outline the way in which the in-plane coordinates are made corotational.

To obtain the updated version of $\mathbf{t}_\alpha$, we follow a two-step approach. First, we construct orthogonal vectors $\hat{\mathbf{t}}_\alpha$ tangential to the surface (following Abaqus conventions). Subsequently, we calculate

$$\hat{f}_{\alpha\beta} = \hat{\mathbf{t}}_\alpha \cdot \frac{\partial \overline{\mathbf{x}}^{t+\Delta t}}{\partial S_\beta}.$$

We then apply an in-plane rotation $\Delta R_{\alpha\beta}$ to the vectors: $\hat{\mathbf{t}}_\alpha$:

$$\Delta R_{11} = \Delta R_{22} = \cos\Delta\psi,$$

$$\Delta R_{21} = -\Delta R_{12} = \sin\Delta\psi,$$

where $\Delta\psi$ is to be determined such that the resulting deformation tensor is symmetric, as

$$\bar{f}_{\alpha\beta} = \Delta R_{\alpha\gamma}\hat{f}_{\gamma\beta} = \hat{f}_{\gamma\alpha}\Delta R_{\beta\gamma} = \bar{f}_{\beta\alpha}.$$

From this follows

$$\tan\Delta\psi = \frac{(\hat{f}_{12} - \hat{f}_{21})}{\hat{f}_{11} + \hat{f}_{22}}.$$

Thus, we can calculate the updated local material directions as

$$\bar{\mathbf{t}}_\alpha^{t+\Delta t} = \Delta R_{\alpha\gamma}\hat{\mathbf{t}}_\gamma.$$

Curvature change

We assume that the nodal spin will be interpolated with the interpolation functions $N^I(\xi_\alpha)$. During an increment the nodal spin is assumed to be constant; consequently, the value of the spin at each material point will be constant. Hence, we can use the same interpolation functions for the incremental finite rotation vector $\Delta\boldsymbol{\phi}$:

$$\Delta\boldsymbol{\phi} = \boldsymbol{\omega}\Delta t = N^I(\xi_\alpha)\boldsymbol{\omega}^I\Delta t = N^I(\xi_\alpha)\Delta\boldsymbol{\phi}^I.$$

The finite rotation vector can be split in a rotation amplitude $\Delta\phi$ and a rotation axis $\mathbf{p}$:

$$\Delta\boldsymbol{\phi} = \Delta\phi\,\mathbf{p}, \qquad \text{with} \quad \Delta\phi \stackrel{\text{def}}{=} |\Delta\boldsymbol{\phi}| \quad \text{and} \quad \mathbf{p} = \Delta\boldsymbol{\phi}/\Delta\phi.$$

To rotate the shell normal, we use quaternion algebra. The incremental nodal rotation is represented by the rotation quaternion $\Delta\boldsymbol{q}$, which is defined by

$$\Delta\boldsymbol{q} \stackrel{\text{def}}{=} \left(\cos\frac{\Delta\phi}{2}\,,\, \sin\frac{\Delta\phi}{2}\,\mathbf{p}\right).$$

An updated shell normal is then obtained according to

$$\tilde{\mathbf{t}}_3^{t+\Delta t} = \Delta\boldsymbol{q}\,\mathbf{t}_3^t\Delta\boldsymbol{q}^\dagger.$$

This updated shell normal does not actually have to be calculated: it is used only for the derivation of the expression for the curvature change. It is not equal to the shell normal used at the start of the next increment $\mathbf{t}_3^{t+\Delta t}$, which will again be chosen perpendicular to the reference surface. The updated normal used here will be approximately orthogonal to the reference surface, depending upon the amount of transverse shear deformation. The gradient of the updated shell normal can be obtained by differentiation:

$$\frac{\partial\tilde{\mathbf{t}}_3^{t+\Delta t}}{\partial S_\beta} = \frac{\partial\Delta\boldsymbol{q}}{\partial S_\beta}\mathbf{t}_3^t\Delta\boldsymbol{q}^\dagger + \Delta\boldsymbol{q}\,\mathbf{t}_3^t\frac{\partial\Delta\boldsymbol{q}^\dagger}{\partial S_\beta} + \Delta\boldsymbol{q}\frac{\partial\mathbf{t}_3^t}{\partial S_\beta}\Delta\boldsymbol{q}^\dagger.$$

The second term on the right-hand side can be written in the form

$$\Delta\boldsymbol{q}\,\mathbf{t}_3^t\frac{\partial\Delta\boldsymbol{q}^\dagger}{\partial S_\beta} = \left[\frac{\partial\Delta\boldsymbol{q}}{\partial S_\beta}\left(\mathbf{t}_3^t\right)^\dagger\Delta\boldsymbol{q}^\dagger\right]^\dagger = -\left[\frac{\partial\Delta\boldsymbol{q}}{\partial S_\beta}\mathbf{t}_3^t\Delta\boldsymbol{q}^\dagger\right]^\dagger.$$

Hence, the scalar parts of the first two terms cancel each other and the vector parts reinforce each other, leading to

$$\frac{\partial\tilde{\mathbf{t}}_3^{t+\Delta t}}{\partial S_\beta} = 2\mathbf{V}\left(\frac{\partial\Delta\boldsymbol{q}}{\partial S_\beta}\mathbf{t}_3^t\Delta\boldsymbol{q}^\dagger\right) + \Delta\boldsymbol{q}\frac{\partial\mathbf{t}_3^t}{\partial S_\beta}\Delta\boldsymbol{q}^\dagger.$$

The inverse of a rotation quaternion such as $\Delta\boldsymbol{q}$ is equal to its conjugate ($\Delta\boldsymbol{q}^{-1} = \Delta\boldsymbol{q}^\dagger$). Hence, we can write

$$\begin{aligned}\frac{\partial\tilde{\mathbf{t}}_3^{t+\Delta t}}{\partial S_\beta} &= 2\mathbf{V}\left(\frac{\partial\Delta\boldsymbol{q}}{\partial S_\beta}\Delta\boldsymbol{q}^\dagger\Delta\boldsymbol{q}\,\mathbf{t}_3^t\Delta\boldsymbol{q}^\dagger\right) + \Delta\boldsymbol{q}\frac{\partial\mathbf{t}_3^t}{\partial S_\beta}\Delta\boldsymbol{q}^\dagger \\ &= \mathbf{V}\left(2\frac{\partial\Delta\boldsymbol{q}}{\partial S_\beta}\Delta\boldsymbol{q}^\dagger\tilde{\mathbf{t}}_3^{t+\Delta t}\right) + \Delta\boldsymbol{q}\frac{\partial\mathbf{t}_3^t}{\partial S_\beta}\Delta\boldsymbol{q}^\dagger \\ &= \Delta\mathbf{R}_\beta\times\tilde{\mathbf{t}}_3^{t+\Delta t} + \Delta\boldsymbol{q}\frac{\partial\mathbf{t}_3^t}{\partial S_\beta}\Delta\boldsymbol{q}^\dagger,\end{aligned}$$

where we have formally defined the incremental gradient update vectors

$$\Delta\mathbf{R}_\beta \stackrel{\text{def}}{=} \mathbf{V}\left(2\frac{\partial\Delta\boldsymbol{q}}{\partial S_\beta}\Delta\boldsymbol{q}^\dagger\right),$$

which must be expressed in terms of the gradient of the incremental rotation. From the definition of the incremental quaternion $\Delta\boldsymbol{q}$ follows

$$\frac{\partial\Delta\boldsymbol{q}}{\partial S_\beta} = \left(\frac{1}{2}\sin\frac{\Delta\phi}{2}\frac{\partial\Delta\phi}{\partial S_\beta},\ \frac{1}{2}\cos\frac{\Delta\phi}{2}\mathbf{p}\frac{\partial\Delta\phi}{\partial S_\beta} + \sin\frac{\Delta\phi}{2}\frac{\partial\mathbf{p}}{\partial S_\beta}\right);$$

thus, for $\Delta\mathbf{R}_\beta$, again with use of the incremental quaternion definition

$$\Delta\mathbf{R}_\beta = \mathbf{p}\frac{\partial\Delta\phi}{\partial S_\beta} + \sin\Delta\phi\,\frac{\partial\mathbf{p}}{\partial S_\beta} + (1-\cos\Delta\phi)\,\mathbf{p}\times\frac{\partial\mathbf{p}}{\partial S_\beta}.$$

From the definition of $\Delta\phi$ and $\mathbf{p}$ follows

$$\frac{\partial\Delta\phi}{\partial S_\beta} = \frac{1}{\Delta\phi}\Delta\boldsymbol{\phi}\cdot\frac{\partial\Delta\boldsymbol{\phi}}{\partial S_\beta},$$
$$\frac{\partial\mathbf{p}}{\partial S_\beta} = \frac{1}{\Delta\phi}\frac{\partial\Delta\boldsymbol{\phi}}{\partial S_\beta} - \frac{1}{\Delta\phi^3}\Delta\boldsymbol{\phi}\,\Delta\boldsymbol{\phi}\cdot\frac{\partial\Delta\boldsymbol{\phi}}{\partial S_\beta} = \frac{1}{\Delta\phi}\left[\frac{\partial\Delta\boldsymbol{\phi}}{\partial S_\beta} - \mathbf{p}\,\mathbf{p}\cdot\frac{\partial\Delta\boldsymbol{\phi}}{\partial S_\beta}\right].$$

After substitution in the expression for $\Delta\mathbf{R}_\beta$ and some algebra one obtains

$$\Delta\mathbf{R}_\beta = \frac{\sin\Delta\phi}{\Delta\phi}\frac{\partial\Delta\boldsymbol{\phi}}{\partial S_\beta} + \frac{1-\cos\Delta\phi}{\Delta\phi}\mathbf{p}\times\frac{\partial\Delta\boldsymbol{\phi}}{\partial S_\beta} + \left(1-\frac{\sin\Delta\phi}{\Delta\phi}\right)\mathbf{p}\,\mathbf{p}\cdot\frac{\partial\Delta\boldsymbol{\phi}}{\partial S_\beta}.$$

Note that $\Delta\mathbf{R}_\beta \to \dfrac{\partial\Delta\boldsymbol{\phi}}{\partial S_\beta}$ when $\Delta\phi\to 0$.

For the gradient $B_{\alpha\beta}$ of the updated shell normal we obtain

$$B^{t+\Delta t}_{\alpha\beta} = \mathbf{t}^{t+\Delta t}_\alpha\cdot\frac{\partial\tilde{\mathbf{t}}^{t+\Delta t}_3}{\partial S_\beta} = \mathbf{t}^{t+\Delta t}_\alpha\left[\Delta\mathbf{R}_\beta\times\tilde{\mathbf{t}}^{t+\Delta t}_3 + \Delta\boldsymbol{q}\,\mathbf{t}^t_\gamma B^t_{\alpha\beta}\Delta\boldsymbol{q}^\dagger\right]$$
$$= \Delta\mathbf{R}_\beta\cdot\left(\tilde{\mathbf{t}}^{t+\Delta t}_3\times\mathbf{t}^{t+\Delta t}_\alpha\right) + \mathbf{t}^{t+\Delta t}_\alpha\cdot\mathbf{t}^{t+\Delta t}_\gamma B^t_{\gamma\beta} = \epsilon^\gamma_\alpha\mathbf{t}^{t+\Delta t}_\gamma\cdot\Delta\mathbf{R}_\beta + B^t_{\alpha\beta},$$

where we have introduced the two-dimensional alternator ϵ^β_α:

$$\epsilon^1_1 = \epsilon^2_2 = 0, \qquad \epsilon^2_1 = -\epsilon^1_2 = 1.$$

Note that the *change* in $B_{\alpha\beta}$ is independent of $B^t_{\alpha\beta}$.

Calculation of $B_{\alpha\beta}$ involves taking the gradient with respect to the reference configuration. It is more convenient to use the reference surface curvature tensor

$$b_{\alpha\beta} \stackrel{\text{def}}{=} B_{\alpha\gamma}\overline{h}_{\gamma\beta} = \mathbf{t}_\alpha\cdot\frac{\partial\mathbf{t}_3}{\partial s_\beta}.$$

We then introduce the incremental curvature update vectors

$$\Delta\mathbf{r}_\beta \stackrel{\text{def}}{=} \Delta\mathbf{R}_\gamma\overline{h}_{\gamma\beta} = \frac{\sin\Delta\phi}{\Delta\phi}\frac{\partial\Delta\boldsymbol{\phi}}{\partial s_\beta} + \frac{1-\cos\Delta\phi}{\Delta\phi}\mathbf{p}\times\frac{\partial\Delta\boldsymbol{\phi}}{\partial s_\beta} + \left(1-\frac{\sin\Delta\phi}{\Delta\phi}\right)\mathbf{p}\,\mathbf{p}\cdot\frac{\partial\Delta\boldsymbol{\phi}}{\partial s_\beta},$$

which makes it possible to write the update equation as

$$b^{t+\Delta t}_{\alpha\beta} = \epsilon^\gamma_\alpha\mathbf{t}^{t+\Delta t}_\gamma\cdot\Delta\mathbf{r}_\beta + b^t_{\alpha\gamma}\Delta\overline{h}_{\gamma\beta}.$$

This expression makes it feasible to calculate the update in the reference surface curvature by taking gradients in the latest updated state only.

Deformation gradient

We already have obtained an expression for the deformation gradient in the reference surface, and we have assumed that the thickness change is constant:

$$\overline{F}_{\alpha\beta} = \overline{f}_{\alpha\beta}, \qquad \overline{F}_{33} = \overline{f}_{33}.$$

At other points in the shell we obtain for the in-plane component

$$F_{\alpha\beta} = \mathbf{t}_\alpha \cdot \frac{\partial \mathbf{x}}{\partial S_\gamma} \left(\mathbf{T}_\beta \cdot \frac{\partial \mathbf{X}}{\partial S_\gamma} \right)^{-1} = (\overline{f}_{\alpha\gamma} + \overline{f}_{33} S_3 B_{\alpha\gamma})(\delta_{\gamma\beta} + S_3 B^\circ_{\gamma\beta})^{-1}.$$

We neglect terms of order $(S_3)^2$, which yields the simplified relation

$$F_{\alpha\beta} = \overline{F}_{\alpha\beta} + S_3(\overline{f}_{33} B_{\alpha\beta} - \overline{f}_{\alpha\gamma} B^\circ_{\gamma\beta}).$$

We can write this as the product of a finite-membrane deformation and a bending perturbation:

$$\begin{aligned} F_{\alpha\beta} &= \left[\delta_{\alpha\gamma} + S_3(\overline{f}_{33} B_{\alpha\delta} \overline{h}_{\delta\gamma} - \overline{f}_{\alpha\delta} B^\circ_{\delta\varepsilon} \overline{h}_{\varepsilon\gamma})\right] \overline{f}_{\gamma\beta} \\ &= \left[\delta_{\alpha\gamma} + S_3(\overline{f}_{33} b_{\alpha\gamma} - \overline{f}_{\alpha\delta} b^\circ_{\delta\varepsilon} \overline{h}_{\varepsilon\gamma})\right] \overline{f}_{\gamma\beta}. \end{aligned}$$

It will be assumed that the deformation (strain and rotation) due to bending is small and, therefore,

$$S_3(\overline{f}_{33} b_{\alpha\gamma} - \overline{f}_{\alpha\delta} b^\circ_{\delta\varepsilon} \overline{h}_{\varepsilon\gamma}) \ll 1.$$

Membrane strain increment

The membrane strain increment follows from the incremental stretch tensor $\Delta\mathbf{V}$, whose components follow from the incremental deformation gradient $\Delta\overline{f}_{\alpha\beta}$ by the polar decomposition $\Delta\overline{f}_{\alpha\beta} = \Delta\overline{V}_{\alpha\gamma}\Delta\overline{R}_{\gamma\beta}$.

Let $\overline{f}^t_{\alpha\beta}$ and $\overline{f}^{t+\Delta t}_{\alpha\beta}$ be the deformation gradient at the beginning and the end of the increment, respectively. By definition $\overline{f}^{t+\Delta t}_{\alpha\beta} = \Delta\overline{f}_{\alpha\delta}\overline{f}^t_{\delta\beta}$. The incremental deformation gradient follows as

$$\Delta\overline{f}_{\alpha\beta} = \overline{f}^{t+\Delta t}_{\alpha\delta} \left(\overline{f}^{t^{-1}} \right)_{\delta\beta}.$$

Since $\Delta\overline{R}_{\gamma\beta}$ are the components of an orthogonal matrix, the square of the incremental stretch tensor can be obtained by

$$\Delta\overline{f}_{\alpha\gamma}\Delta\overline{f}_{\beta\gamma} = \Delta\overline{V}_{\alpha\gamma}\Delta\overline{V}_{\beta\gamma} = \sum_{I=1}^{2} (\Delta\lambda_I)^2 a^I_\alpha a^I_\beta$$

(see "Deformation," Section 1.4.1). The logarithmic strain increment is then

$$\Delta\epsilon_{\alpha\beta} = \sum_{I=1}^{2} \ln(\Delta\lambda_I) a_\alpha^I a_\beta^I$$

and the average material rotation increment is defined from the polar decomposition:

$$\Delta\overline{R}_{\alpha\beta} = \sum_{I=1}^{2} \frac{1}{\Delta\lambda_I} a_\alpha^I a_\gamma^I \Delta\overline{f}_{\gamma\beta}\,.$$

Due to the choice of the element basis directions, it follows that

$$\Delta\overline{R}_{\alpha\beta} \approx \delta_{\alpha\beta}\,.$$

Curvature increment

Following Koiter-Sanders shell theory, and compensating for the rotation of the base vectors relative to the material, we define the physical curvature increment $\Delta\kappa_{\alpha\beta}$ as

$$\Delta\kappa_{\alpha\beta} = \mathrm{sym}\left[b_{\alpha\beta}^{t+\Delta t} - b_{\alpha\gamma}^t \Delta\overline{R}_{\beta\gamma} + b_{\alpha\gamma}^t \Delta\overline{R}_{\delta\gamma}\Delta\overline{\varepsilon}_{\delta\beta}\right] = \mathrm{sym}\left[b_{\alpha\beta}^{t+\Delta t} - b_{\alpha\gamma}^t \Delta\overline{R}_{\delta\gamma}(\delta_{\delta\beta} - \Delta\overline{\varepsilon}_{\delta\beta})\right].$$

Neglecting terms of the order $(\Delta\overline{\varepsilon}_{\alpha\beta})^2$ relative to $\Delta\overline{\varepsilon}_{\alpha\beta}$, this expression can be rewritten as

$$\Delta\kappa_{\alpha\beta} = \mathrm{sym}\left[b_{\alpha\beta}^{t+\Delta t} - b_{\alpha\gamma}^t \Delta\overline{R}_{\delta\gamma}\Delta\overline{V}^{-1}_{\delta\beta}\right] = \mathrm{sym}\left[b_{\alpha\beta}^{t+\Delta t} - b_{\alpha\gamma}^t \Delta\overline{h}_{\gamma\beta}\right] = \mathrm{sym}\left[\epsilon_\alpha^\gamma \mathbf{t}_\gamma^{t+\Delta t} \cdot \Delta\mathbf{r}_\beta\right],$$

where use was made of the curvature update formula. Observe that the curvature at the beginning of the increment, $b_{\alpha\beta}^t$, does not appear in this equation. Hence, there is no need to calculate the initial curvature $b_{\alpha\beta}^\circ$, and we can assume $b_{\alpha\beta}^\circ = 0$. The deformation gradient can, hence, also be simplified to

$$F_{\alpha\beta} = \overline{f}_{\alpha\beta} + S_3 \overline{f}_{33} b_{\alpha\gamma} \overline{f}_{\gamma\beta}.$$

For the material strain increment at a point through the shell thickness Koiter-Sanders theory thus yields

$$\Delta\varepsilon_{\alpha\beta} = \Delta\overline{\varepsilon}_{\alpha\beta} + \overline{f}_{33}^{t+\Delta t} S_3 \Delta\kappa_{\alpha\beta}.$$

Virtual work

The virtual work contribution of the stresses is

$$\delta\Pi = \int_V \sigma_{\alpha\beta}\delta\varepsilon_{\alpha\beta} dV.$$

We assume that the variations in the strain can be expressed in terms of variation in membrane strain and curvature with the same relations as apply to the increment in strain:

$$\delta\varepsilon_{\alpha\beta} = \delta\overline{\varepsilon}_{\alpha\beta} + \overline{f}_{33}^{t+\Delta t} S_3 \delta\kappa_{\alpha\beta},$$

which transforms the virtual work equation into

$$\delta\Pi = \int_V \sigma_{\alpha\beta}(\delta\overline{\varepsilon}_{\alpha\beta} + \overline{f}_{33} S_3 \delta\kappa_{\alpha\beta}) dV.$$

We introduce the membrane forces $N_{\alpha\beta}$ and the bending moments $M_{\alpha\beta}$:

$$N_{\alpha\beta} \stackrel{\text{def}}{=} \int_h \sigma_{\alpha\beta}\overline{f}_{33} dS_3, \qquad M_{\alpha\beta} \stackrel{\text{def}}{=} \int_h \sigma_{\alpha\beta}\overline{f}_{33}^2 S_3 \mathrm{d}S_3,$$

which allows us to write

$$\delta\Pi = \int_A \left(N_{\alpha\beta}\delta\overline{\varepsilon}_{\alpha\beta} + M_{\alpha\beta}\delta\kappa_{\alpha\beta}\right) dA.$$

The membrane strain variation follows with the usual expression

$$\delta\overline{\varepsilon}_{\alpha\beta} = \mathrm{sym}(\delta\overline{f}_{\alpha\gamma}\overline{h}_{\gamma\beta}) = \mathrm{sym}(\delta\mathbf{t}_\alpha \cdot \mathbf{t}_\beta + \mathbf{t}_\alpha \cdot \frac{\partial\delta\overline{\mathbf{x}}}{\partial s_\beta}) = \mathrm{sym}(\mathbf{t}_\alpha \cdot \frac{\partial\delta\overline{\mathbf{x}}}{\partial s_\beta}),$$

where we have used the identity $\mathrm{sym}(\delta\mathbf{t}_\alpha \cdot \mathbf{t}_\beta) = \delta(\mathbf{t}_\alpha \cdot \mathbf{t}_\beta) = 0$.

The variation in the curvature is obtained by taking variations in the incremental curvature, which yields

$$\begin{aligned}\delta\kappa_{\alpha\beta} &= \mathrm{sym}(\epsilon_\alpha^\gamma \mathbf{t}_\gamma \cdot \delta\Delta\mathbf{r}_\beta + \epsilon_\alpha^\gamma \delta\mathbf{t}_\gamma \cdot \Delta\mathbf{r}_\beta) \\ &= \mathrm{sym}(\epsilon_\alpha^\gamma \mathbf{t}_\gamma \cdot \delta\Delta\mathbf{R}_\delta \overline{h}_{\delta\beta} + \epsilon_\alpha^\gamma \mathbf{t}_\gamma \cdot \Delta\mathbf{R}_\delta \delta\overline{h}_{\delta\beta} + \epsilon_\alpha^\gamma \delta\mathbf{t}_\gamma \cdot \Delta\mathbf{R}_\delta \overline{h}_{\delta\beta}).\end{aligned}$$

We neglect the terms of order $\Delta\kappa_{\alpha\beta}$ and also terms of order $\mathbf{t}_3 \cdot \Delta\mathbf{r}_\delta$, which yields

$$\delta\kappa_{\alpha\beta} = \mathrm{sym}(\epsilon_\alpha^\gamma \mathbf{t}_\gamma \cdot \delta\Delta\mathbf{r}_\delta \overline{h}_{\delta\beta}).$$

We evaluate $\delta\Delta\mathbf{r}_\delta$ with respect to the current state (at the end of the increment). Hence for the evaluation we can assume $\Delta\boldsymbol{\phi} = 0$. Moreover, we neglect terms of the order $\mathbf{t}_\alpha \cdot \dfrac{\partial\delta\Delta\boldsymbol{\phi}}{\partial S_\beta}$ since they are proportional to $\Delta\kappa_{\alpha\beta}$. Hence, we obtain

$$\mathbf{t}_\gamma \delta\Delta\mathbf{r}_\delta \approx \mathbf{t}_\gamma \cdot \frac{\partial\delta\boldsymbol{\phi}}{\partial S_\delta},$$

which substituted in the expression for $\delta\kappa_{\alpha\beta}$ yields

$$\delta\kappa_{\alpha\beta} = \mathrm{sym}\left(\epsilon_\alpha^\gamma \mathbf{t}_\gamma \cdot \frac{\partial\delta\boldsymbol{\phi}}{\partial S_\delta}\overline{h}_{\delta\beta}\right) = \mathrm{sym}\left(\epsilon_\alpha^\gamma \mathbf{t}_\gamma \cdot \frac{\partial\delta\boldsymbol{\phi}}{\partial s_\beta}\right).$$

The rate of virtual work

To obtain an expression for the rate of virtual work, we first write the virtual work equation in terms of the reference volume

$$\delta\Pi = \int_{V^\circ} \tau_{\alpha\beta}\delta\varepsilon_{\alpha\beta} dV^\circ = \int_{A^\circ}\int_h \tau_{\alpha\beta}\delta\varepsilon_{\alpha\beta} dS_3 dA^\circ,$$

where $\tau_{\alpha\beta}$ is the Kirchhoff stress tensor, related to the Cauchy or true stress tensor via

$$\tau_{\alpha\beta} = J\sigma_{\alpha\beta}.$$

The rate of change then becomes

$$\mathrm{d}\delta\Pi = \int_{A^\circ}\int_h (\mathrm{d}^\nabla\tau_{\alpha\beta}\delta\varepsilon_{\alpha\beta} + \tau_{\alpha\beta}\mathrm{d}^\nabla\delta\varepsilon_{\alpha\beta}) dS_3 dA^\circ.$$

Here d^∇ indicates that the rates are taken in a material, corotational coordinate system. The terms involving stress rates are related to the material behavior. We assume constitutive equations of the form

$$\mathrm{d}^\nabla\tau_{\alpha\beta} = JC_{\alpha\beta\gamma\delta}\mathrm{d}\varepsilon_{\delta\gamma}.$$

Substituted in the expression $d\delta\Pi$ and transformed back to the current configuration, this yields

$$\mathrm{d}\delta\Pi = \int_A\int_h (\delta\varepsilon_{\alpha\beta}C_{\alpha\beta\gamma\delta}\mathrm{d}\varepsilon_{\gamma\delta} + \sigma_{\alpha\beta}\mathrm{d}^\nabla\delta\varepsilon_{\alpha\beta})\overline{f}_{33} dS_3 dA.$$

Consistent with the derivation of the virtual work equation itself, we neglect terms of the order $\mathrm{d}\overline{f}_{33}S_3\delta\kappa_{\alpha\beta}$. Hence, the rate of virtual work can be written as

$$\begin{aligned}\mathrm{d}\delta\Pi = \int_A \Big[\int_h (\delta\overline{\varepsilon}_{\alpha\beta} + \overline{f}_{33}S_3\delta\kappa_{\alpha\beta})C_{\alpha\beta\gamma\delta}(\mathrm{d}\overline{\varepsilon}_{\gamma\delta} + \overline{f}_{33}S_3\mathrm{d}\kappa_{\gamma\delta})\overline{f}_{33} dS_3 + \\ N_{\alpha\beta}\mathrm{d}^\nabla\delta\overline{\varepsilon}_{\alpha\beta} + M_{\alpha\beta}\mathrm{d}^\nabla\delta\kappa_{\alpha\beta}\Big] dA.\end{aligned}$$

Second variation of the membrane strain

It remains to determine $\mathrm{d}^\nabla\delta\overline{\varepsilon}_{\alpha\beta}$ and $\mathrm{d}^\nabla\delta\kappa_{\alpha\beta}$. From the first variation $\delta\overline{\varepsilon}_{\alpha\beta}$ follows

$$\mathrm{d}^\nabla\delta\overline{\varepsilon}_{\alpha\beta} = \mathrm{sym}(\mathrm{d}^\nabla\delta\overline{f}_{\alpha\gamma}\overline{h}_{\gamma\beta} + \delta\overline{f}_{\alpha\gamma}\mathrm{d}^\nabla\overline{h}_{\gamma\beta}).$$

Since $\overline{h}_{\alpha\beta}$ is the inverse of $\overline{f}_{\alpha\beta}$, it follows that

$$\mathrm{d}^\nabla\overline{h}_{\alpha\beta} = -\overline{h}_{\alpha\gamma}\mathrm{d}^\nabla\overline{f}_{\gamma\delta}\overline{h}_{\delta\beta}.$$

Substitution in the expression for the second variation yields

$$\begin{aligned}\mathrm{d}^{\nabla}\delta\overline{\varepsilon}_{\alpha\beta} &= \mathrm{sym}\left(\mathrm{d}^{\nabla}\mathbf{t}_{\alpha}\cdot\frac{\partial\delta\overline{\mathbf{x}}}{\partial s_{\beta}} - \mathbf{t}_{\alpha}\cdot\frac{\partial\delta\overline{\mathbf{x}}}{\partial s_{\gamma}}\mathrm{d}^{\nabla}\mathbf{t}_{\gamma}\cdot\frac{\partial\overline{\mathbf{x}}}{\partial s_{\beta}} - \mathbf{t}_{\alpha}\cdot\frac{\partial\delta\overline{\mathbf{x}}}{\partial s_{\gamma}}\mathbf{t}_{\gamma}\cdot\frac{\partial\mathrm{d}^{\nabla}\overline{\mathbf{x}}}{\partial s_{\beta}}\right)\\ &= \mathrm{sym}\left(\mathrm{d}^{\nabla}\mathbf{t}_{\alpha}\cdot\frac{\partial\delta\overline{\mathbf{x}}}{\partial s_{\beta}} - \mathrm{d}^{\nabla}\mathbf{t}_{\gamma}\cdot\mathbf{t}_{\beta}\mathbf{t}_{\alpha}\cdot\frac{\partial\delta\overline{\mathbf{x}}}{\partial s_{\gamma}} - \frac{\partial\mathrm{d}^{\nabla}\overline{\mathbf{x}}}{\partial s_{\beta}}\cdot\mathbf{t}_{\gamma}\mathbf{t}_{\alpha}\cdot\frac{\partial\delta\overline{\mathbf{x}}}{\partial s_{\gamma}}\right).\end{aligned}$$

The corotational rate of the base vectors follows from

$$\mathrm{d}^{\nabla}\mathbf{t}_{\beta} = \frac{\partial\mathrm{d}\overline{\mathbf{x}}}{\partial s_{\beta}} + \frac{\partial\overline{\mathbf{x}}}{\partial S_{\gamma}}\mathrm{d}^{\nabla}\overline{h}_{\gamma\beta} = \frac{\partial\mathrm{d}\overline{\mathbf{x}}}{\partial s_{\beta}} - \mathrm{d}^{\nabla}\mathbf{t}_{\gamma}\cdot\mathbf{t}_{\beta}\mathbf{t}_{\gamma} - \frac{\partial\mathrm{d}\overline{\mathbf{x}}}{\partial s_{\beta}}\cdot\mathbf{t}_{\gamma}\mathbf{t}_{\gamma}.$$

Substituted in the first term of the previous expression yields

$$\begin{aligned}\mathrm{d}^{\nabla}\delta\overline{\varepsilon}_{\alpha\beta} =& \mathrm{sym}\left[\frac{\partial\mathrm{d}\overline{\mathbf{x}}}{\partial s_{\alpha}}\cdot\frac{\partial\delta\overline{\mathbf{x}}}{\partial s_{\beta}} - \mathrm{d}^{\nabla}\mathbf{t}_{\gamma}\cdot\mathbf{t}_{\alpha}\mathbf{t}_{\gamma}\cdot\frac{\partial\delta\overline{\mathbf{x}}}{\partial s_{\beta}} - \mathrm{d}^{\nabla}\mathbf{t}_{\gamma}\cdot\mathbf{t}_{\beta}\mathbf{t}_{\alpha}\cdot\frac{\partial\delta\overline{\mathbf{x}}}{\partial s_{\gamma}} - \right.\\ & \left.\frac{\partial\mathrm{d}\overline{\mathbf{x}}}{\partial s_{\alpha}}\cdot\mathbf{t}_{\gamma}\mathbf{t}_{\gamma}\cdot\frac{\partial\delta\overline{\mathbf{x}}}{\partial s_{\beta}} - \frac{\partial\mathrm{d}\overline{\mathbf{x}}}{\partial s_{\beta}}\cdot\mathbf{t}_{\gamma}\mathbf{t}_{\alpha}\cdot\frac{\partial\delta\overline{\mathbf{x}}}{\partial s_{\gamma}}\right]\\ =& \mathrm{sym}\left[\frac{\partial\mathrm{d}\overline{\mathbf{x}}}{\partial s_{\alpha}}\cdot\frac{\partial\delta\overline{\mathbf{x}}}{\partial s_{\beta}} - 2\left(\mathrm{d}^{\nabla}\mathbf{t}_{\gamma}\cdot\mathbf{t}_{\alpha} + \frac{\partial\mathrm{d}\overline{\mathbf{x}}}{\partial s_{\alpha}}\cdot\mathbf{t}_{\gamma}\right)\delta\overline{\varepsilon}_{\gamma\beta}\right].\end{aligned}$$

The in-plane components of the corotational rate of the base vectors can also be expressed in terms of the in-plane material spin in the reference surface:

$$\mathrm{d}^{\nabla}\mathbf{t}_{\gamma} = \mathrm{d}\Omega_{\gamma\beta}\mathbf{t}_{\beta} = \frac{1}{2}\left(\mathbf{t}_{\beta}\cdot\frac{\partial\mathrm{d}\overline{\mathbf{x}}}{\partial s_{\gamma}} - \mathbf{t}_{\gamma}\cdot\frac{\partial\mathrm{d}\overline{\mathbf{x}}}{\partial s_{\beta}}\right)\mathbf{t}_{\beta}.$$

Substitution in the last obtained expression for $\mathrm{d}^{\nabla}\delta\overline{\varepsilon}_{\alpha\beta}$ yields

$$\mathrm{d}^{\nabla}\delta\overline{\varepsilon}_{\alpha\beta} = \mathrm{sym}\left(\frac{\partial\mathrm{d}\overline{\mathbf{x}}}{\partial s_{\alpha}}\cdot\frac{\partial\delta\overline{\mathbf{x}}}{\partial s_{\beta}} - 2\mathrm{d}\overline{\varepsilon}_{\alpha\gamma}\delta\overline{\varepsilon}_{\gamma\beta}\right).$$

This expression is identical to the one obtained with "standard" continuum elements.

Second variation of the curvature

We need to calculate the second variation of the curvature to calculate the initial stress contribution from the curvature:

$$m^{\alpha\beta}\mathrm{d}^{\nabla}\delta\kappa_{\alpha\beta}\,.$$

To simplify the computation, we rely on the intrinsic definition of curvature and express the curvature in derivatives with respect to the isoparametric coordinates. Accordingly,

$$m^{\alpha\beta}\mathrm{d}^{\nabla}\delta\kappa_{\alpha\beta} = m^{ab}\mathrm{d}^{\nabla}\delta\kappa_{ab} = m^{ab}\mathrm{d}^{\nabla}\left(\frac{\partial\delta\overline{\mathbf{x}}}{\partial\xi^a}\cdot\frac{\partial\mathbf{t}_3}{\partial\xi^b} + \frac{\partial\overline{\mathbf{x}}}{\partial\xi^a}\cdot\frac{\partial\delta\mathbf{t}_3}{\partial\xi^b}\right),$$

where the bending resultant components m^{ab} are the components expressed in the orthonormal coordinate system ($m^{\alpha\beta}$) transformed by $\partial\xi^a/\partial s_\alpha$.

Denoting derivatives with respect to the isoparametric coordinates as $\partial/\partial\xi^a \equiv {}_{,a}$, the second variation of the curvature is

$$\mathrm{d}^{\nabla}\delta\kappa_{ab} = \delta\overline{\mathbf{x}}_{,a}\cdot\mathrm{d}^{\nabla}\mathbf{t}_{3,b} + \mathrm{d}\overline{\mathbf{x}}_{,a}\cdot\delta\mathbf{t}_{3,b} + \overline{\mathbf{x}}_{,a}\cdot(\mathrm{d}^{\nabla}\delta\mathbf{t}_3)_{,b}\,.$$

Using the fact that $\delta\mathbf{t}_3 = \delta\boldsymbol{\phi}\times\mathbf{t}_3$ and $\mathrm{d}^{\nabla}\mathbf{t}_3 = \dot{\boldsymbol{\phi}}\times\mathbf{t}_3$, we find that

$$\begin{aligned}\mathrm{d}^{\nabla}\delta\kappa_{ab} = {}& \delta\overline{\mathbf{x}}_{,a}\cdot[-\hat{\mathbf{t}}_3]\cdot\dot{\boldsymbol{\phi}}_{,b} + \delta\overline{\mathbf{x}}_{,a}\cdot[-\hat{\mathbf{t}}_{3,b}]\cdot\dot{\boldsymbol{\phi}} + \delta\boldsymbol{\phi}_{,b}\cdot[\hat{\mathbf{t}}_3]\cdot\mathrm{d}\overline{\mathbf{x}}_{,a} + \delta\boldsymbol{\phi}\cdot[\hat{\mathbf{t}}_{3,b}]\cdot\mathrm{d}\overline{\mathbf{x}}_{,a}\\ &+ \delta\boldsymbol{\phi}_{,b}\cdot(\mathbf{t}_3\overline{\mathbf{x}}_{,a})\cdot\dot{\boldsymbol{\phi}} + \delta\boldsymbol{\phi}\cdot(\mathbf{t}_{3,b}\overline{\mathbf{x}}_{,a})\cdot\dot{\boldsymbol{\phi}} + \delta\boldsymbol{\phi}\cdot(\mathbf{t}_3\overline{\mathbf{x}}_{,a})\cdot\dot{\boldsymbol{\phi}}_{,b}\\ &- \delta\boldsymbol{\phi}_{,b}\cdot(\mathbf{t}_3\cdot\overline{\mathbf{x}}_{,a}\mathbf{I})\cdot\dot{\boldsymbol{\phi}} - \delta\boldsymbol{\phi}\cdot(\mathbf{t}_3\cdot\overline{\mathbf{x}}_{,a}\mathbf{I})\cdot\dot{\boldsymbol{\phi}}_{,b} - \delta\boldsymbol{\phi}\cdot(\mathbf{t}_{3,b}\cdot\overline{\mathbf{x}}_{,a}\mathbf{I})\cdot\dot{\boldsymbol{\phi}}\,.\end{aligned}$$

Here $[\hat{\mathbf{t}}_3]$ indicates the skew-symmetric tensor with axial vector $\mathbf{t}_3$.

Transverse shear treatment

Several interpolation schemes have been proposed to avoid shear-locking, which typically arises as the thickness of a plate or shell goes to zero. Here we employ an assumed strain method based on the Hu-Washizu principle. This scheme derives from that by MacNeal (1978), subsequently extended and reformulated in Hughes and Tezduyar (1981) and MacNeal (1982) and revisited in Bathe and Dvorkin (1984). Computational aspects of the nonlinear theory are investigated in Simo, Fox, and Rifai (1989) for fully integrated quadrilateral shell elements. For reduced integration quadrilateral and triangular shell elements that can be used for both implicit and explicit integration, this assumed strain method needs to be modified. We summarize below the assumed strain method used with fully integrated elements, followed by the modifications required for the one-point integration plus stabilization used in Abaqus.

Construction of the assumed strain field

Consider a typical isoparametric finite element, as depicted in Figure 3.6.5–1, and denote by A, B, C, D the set of midpoints of the element boundaries.

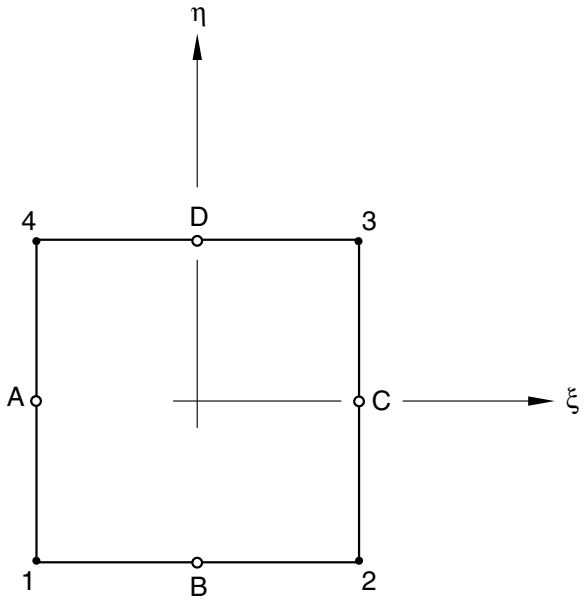


Figure 3.6.5–1 Notation for the assumed strain field on the standard isoparametric element.

The following assumed transverse shear strain field is used:

$$\overline{\gamma}_1 = \frac{1}{2}\left[(1-\eta)\gamma_1^B + (1+\eta)\gamma_1^D\right]$$
$$\overline{\gamma}_2 = \frac{1}{2}\left[(1-\xi)\gamma_2^A + (1+\xi)\gamma_2^C\right],$$

where

$$\gamma_2^A = \mathbf{t}^A \cdot \overline{\mathbf{x}}_{,2}^A - \mathbf{T}^A \cdot \overline{\mathbf{X}}_{,2}^A, \qquad \gamma_1^B = \mathbf{t}^B \cdot \overline{\mathbf{x}}_{,1}^B - \mathbf{T}^B \cdot \overline{\mathbf{X}}_{,1}^B,$$
$$\gamma_2^C = \mathbf{t}^C \cdot \overline{\mathbf{x}}_{,2}^C - \mathbf{T}^C \cdot \overline{\mathbf{X}}_{,2}^C, \qquad \gamma_1^D = \mathbf{t}^D \cdot \overline{\mathbf{x}}_{,1}^D - \mathbf{T}^D \cdot \overline{\mathbf{X}}_{,1}^D$$

are the covariant transverse shear strains evaluated at the midpoints of the element boundaries. In the above transverse shear strain definitions, the use of uppercase letters indicates quantities in the reference configuration and the use of lowercase letters indicates the deformed configuration. For readability we have omitted the subscript 3 from the director field. Making use of the bilinear element interpolation, it follows that

$$\overline{\mathbf{x}}_{,2}^A = \tfrac{1}{2}(\overline{\mathbf{x}}_4 - \overline{\mathbf{x}}_1), \qquad \overline{\mathbf{x}}_{,1}^B = \tfrac{1}{2}(\overline{\mathbf{x}}_2 - \overline{\mathbf{x}}_1),$$
$$\overline{\mathbf{x}}_{,2}^C = \tfrac{1}{2}(\overline{\mathbf{x}}_3 - \overline{\mathbf{x}}_2), \qquad \overline{\mathbf{x}}_{,1}^D = \tfrac{1}{2}(\overline{\mathbf{x}}_3 - \overline{\mathbf{x}}_4),$$

where $\overline{\mathbf{x}}_I$, for $I = 1, 2, 3, 4$, are the reference surface position vectors of the element nodes.

By making use of the assumed strain field along with the update formulae for the director field, the assumed covariant transverse shear field can be written concisely in matrix notation. Recall the director field update equation and the corresponding linearized director field:

$$\mathbf{t}_{t+\Delta t} = \exp[\hat{\Delta\boldsymbol{\phi}}] \cdot \mathbf{t}_t \qquad \Rightarrow \qquad \delta\mathbf{t} = \delta\boldsymbol{\phi} \times \mathbf{t}\,.$$

It follows from the element interpolation that

$$\delta\mathbf{t}^A = \tfrac{1}{2}(\delta\boldsymbol{\phi}_4 + \delta\boldsymbol{\phi}_1) \times \mathbf{t}^A\,, \qquad \delta\mathbf{t}^B = \tfrac{1}{2}(\delta\boldsymbol{\phi}_2 + \delta\boldsymbol{\phi}_1) \times \mathbf{t}^B\,,$$

$$\delta\mathbf{t}^C = \tfrac{1}{2}(\delta\boldsymbol{\phi}_3 + \delta\boldsymbol{\phi}_2) \times \mathbf{t}^C\,, \qquad \delta\mathbf{t}^D = \tfrac{1}{2}(\delta\boldsymbol{\phi}_3 + \delta\boldsymbol{\phi}_4) \times \mathbf{t}^D\,.$$

Define the following vectors:

$$\overline{\boldsymbol{\gamma}} = \left\{ \begin{array}{c} \overline{\gamma}_1 \\ \overline{\gamma}_2 \end{array} \right\}, \qquad \delta\overline{\mathbf{x}} = \left\{ \begin{array}{c} \delta\overline{\mathbf{x}}_1 \\ \delta\overline{\mathbf{x}}_2 \\ \delta\overline{\mathbf{x}}_3 \\ \delta\overline{\mathbf{x}}_4 \end{array} \right\}, \qquad \delta\boldsymbol{\phi} = \left\{ \begin{array}{c} \delta\boldsymbol{\phi}_1 \\ \delta\boldsymbol{\phi}_2 \\ \delta\boldsymbol{\phi}_3 \\ \delta\boldsymbol{\phi}_4 \end{array} \right\}.$$

Then, the linearized transverse shear strain is

$$\delta\overline{\boldsymbol{\gamma}} \equiv \left\{ \begin{array}{c} \delta\overline{\gamma}_1 \\ \delta\overline{\gamma}_2 \end{array} \right\} = \bar{B}_{sm}\delta\overline{\mathbf{x}} + \bar{B}_{sb}\delta\boldsymbol{\phi}\,,$$

where

$$\bar{B}_{sm} = \frac{1}{4}\begin{bmatrix} -(1-\eta)\mathbf{t}^{B\,T} & (1-\eta)\mathbf{t}^{B\,T} & (1+\eta)\mathbf{t}^{D\,T} & -(1+\eta)\mathbf{t}^{D\,T} \\ -(1-\xi)\mathbf{t}^{A\,T} & -(1+\xi)\mathbf{t}^{C\,T} & (1+\xi)\mathbf{t}^{C\,T} & (1-\xi)\mathbf{t}^{A\,T} \end{bmatrix}.$$

Define the four vectors:

$$\boldsymbol{\eta}_2^A = \mathbf{t}^A \times \overline{\mathbf{x}}_{,2}^A\,, \qquad \boldsymbol{\eta}_1^B = \mathbf{t}^B \times \overline{\mathbf{x}}_{,1}^B\,,$$

$$\boldsymbol{\eta}_2^C = \mathbf{t}^C \times \overline{\mathbf{x}}_{,2}^C\,, \qquad \boldsymbol{\eta}_1^D = \mathbf{t}^D \times \overline{\mathbf{x}}_{,1}^D\,.$$

Then the rotation or bending part of the strain/displacement operator is written

$$\bar{B}_{sb} = \frac{1}{4}\begin{bmatrix} (1-\eta)\boldsymbol{\eta}_1^{B\,T} & (1-\eta)\boldsymbol{\eta}_1^{B\,T} & (1+\eta)\boldsymbol{\eta}_1^{D\,T} & (1+\eta)\boldsymbol{\eta}_1^{D\,T} \\ (1-\xi)\boldsymbol{\eta}_2^{A\,T} & (1+\xi)\boldsymbol{\eta}_2^{C\,T} & (1+\xi)\boldsymbol{\eta}_2^{C\,T} & (1-\xi)\boldsymbol{\eta}_2^{A\,T} \end{bmatrix}.$$

Constitutive relations

A St. Venant-Kirchhoff constitutive model for the Kirchhoff curvilinear components of the resultant transverse shear force is written in terms of the transverse shear strains as

$$\left\{ \begin{array}{c} Q^1 \\ Q^2 \end{array} \right\} = \mathbf{C}_s \left\{ \begin{array}{c} \overline{\gamma}_1 \\ \overline{\gamma}_2 \end{array} \right\} ,$$

where $\mathbf{C}_s$ is the transverse shear stiffness in curvilinear coordinates. For a single isotropic layer,

$$\mathbf{C}_s = \tfrac{5}{6} G_s h \left[\begin{array}{cc} A^{11} & A^{12} \\ A^{21} & A^{22} \end{array} \right] .$$

The matrix $[A^{\alpha\beta}]$ is the inverse of the metric $[A_{\alpha\beta}]$, where metric components in the reference configuration $A_{\alpha\beta}$ are defined by the inner product

$$A_{\alpha\beta} = \overline{\mathbf{X}}_{,\alpha} \cdot \overline{\mathbf{X}}_{,\beta} .$$

The Cauchy or true transverse shear force components in the shell orthonormal coordinate system $\{q^1, q^2\}^T$ are calculated with the coordinate transformation $f^a_\alpha = \partial s^a / \partial \xi^\alpha$ as

$$\mathbf{q} = \left\{ \begin{array}{c} q^1 \\ q^2 \end{array} \right\} = \frac{A}{a} \left[\begin{array}{cc} f^1_1 & f^1_2 \\ f^2_1 & f^2_2 \end{array} \right] \left\{ \begin{array}{c} Q^1 \\ Q^2 \end{array} \right\} ,$$

where $\boldsymbol{A}$ is the element's reference area and $\boldsymbol{a}$ is the current area.

Initial stress stiffness

The calculation of the initial stress stiffness matrix requires the second variation of the assumed transverse strain field. This calculation can be summarized in matrix notation as follows. Define vectors of variations of the nodal displacement quantities:

$$\delta\mathbf{u} = \left\{ \begin{array}{c} \delta\overline{\mathbf{x}}_1 \\ \delta\boldsymbol{\phi}_1 \\ \delta\overline{\mathbf{x}}_2 \\ \delta\boldsymbol{\phi}_2 \\ \delta\overline{\mathbf{x}}_3 \\ \delta\boldsymbol{\phi}_3 \\ \delta\overline{\mathbf{x}}_4 \\ \delta\boldsymbol{\phi}_4 \end{array} \right\} , \qquad \mathrm{d}\mathbf{u} = \left\{ \begin{array}{c} \mathrm{d}\overline{\mathbf{x}}_1 \\ \mathrm{d}\boldsymbol{\phi}_1 \\ \mathrm{d}\overline{\mathbf{x}}_2 \\ \mathrm{d}\boldsymbol{\phi}_2 \\ \mathrm{d}\overline{\mathbf{x}}_3 \\ \mathrm{d}\boldsymbol{\phi}_3 \\ \mathrm{d}\overline{\mathbf{x}}_4 \\ \mathrm{d}\boldsymbol{\phi}_4 \end{array} \right\} .$$

Then the initial stress contribution is written

$$\mathbf{q} \cdot \mathrm{d}\delta\bar{\gamma} = \int \delta\mathbf{u} \cdot \mathbf{K}_s \cdot \Delta\mathbf{u} \, da ,$$

where da is the area measure in the current configuration and $\mathbf{K}_s$ is the (symmetric) transverse shear contribution to the initial stress, defined as follows. Let $\mathbf{I}$ be the 3×3 identity matrix; then define the symmetric matrices

$$\mathbf{A} = Q^2(1-\xi)\left[\text{sym}\{\mathbf{t}^A(\overline{\mathbf{x}}^A_{,2})^T\} - \gamma_2^A\mathbf{I}\right] , \quad \mathbf{B} = Q^1(1-\eta)\left[\text{sym}\{\mathbf{t}^B(\overline{\mathbf{x}}^B_{,1})^T\} - \gamma_1^B\mathbf{I}\right] ,$$

$$\mathbf{C} = Q^2(1+\xi)\left[\text{sym}\{\mathbf{t}^C(\overline{\mathbf{x}}^C_{,2})^T\} - \gamma_2^C\mathbf{I}\right] , \quad \mathbf{D} = Q^1(1+\eta)\left[\text{sym}\{\mathbf{t}^D(\overline{\mathbf{x}}^D_{,1})^T\} - \gamma_1^D\mathbf{I}\right] .$$

Also define the skew-symmetric matrices

$$\hat{\mathbf{a}} = Q^2(1-\xi)\hat{\mathbf{t}}^A , \qquad \hat{\mathbf{b}} = Q^1(1-\eta)\hat{\mathbf{t}}^B ,$$

$$\hat{\mathbf{c}} = Q^2(1+\xi)\hat{\mathbf{t}}^C , \qquad \hat{\mathbf{d}} = Q^1(1+\eta)\hat{\mathbf{t}}^D .$$

Also, let $\mathbf{0}$ be the 3×3 zero matrix. Then $\mathbf{K}_s$ is written

$$\mathbf{K}_s = \frac{1}{8}\begin{bmatrix}
\mathbf{0} & \hat{\mathbf{a}}+\hat{\mathbf{b}} & \mathbf{0} & \hat{\mathbf{b}} & \mathbf{0} & \mathbf{0} & & \hat{\mathbf{a}} \\
-\hat{\mathbf{a}}-\hat{\mathbf{b}} & \mathbf{A}+\mathbf{B} & \hat{\mathbf{b}} & \mathbf{B} & \mathbf{0} & \mathbf{0} & \hat{\mathbf{a}} & \mathbf{A} \\
\mathbf{0} & -\hat{\mathbf{b}} & \mathbf{0} & -\hat{\mathbf{b}}+\hat{\mathbf{c}} & \mathbf{0} & \hat{\mathbf{c}} & \mathbf{0} & \mathbf{0} \\
-\hat{\mathbf{b}} & \mathbf{B} & \hat{\mathbf{b}}-\hat{\mathbf{c}} & \mathbf{B}+\mathbf{C} & \hat{\mathbf{c}} & \mathbf{C} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & -\hat{\mathbf{c}} & \mathbf{0} & -\hat{\mathbf{c}}-\hat{\mathbf{d}} & \mathbf{0} & -\hat{\mathbf{d}} \\
\mathbf{0} & \mathbf{0} & -\hat{\mathbf{c}} & \mathbf{C} & \hat{\mathbf{c}}+\hat{\mathbf{d}} & \mathbf{C}+\mathbf{D} & -\hat{\mathbf{d}} & \mathbf{D} \\
\mathbf{0} & -\hat{\mathbf{a}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \hat{\mathbf{d}} & \mathbf{0} & \hat{\mathbf{d}}-\hat{\mathbf{a}} \\
-\hat{\mathbf{a}} & \mathbf{A} & \mathbf{0} & \mathbf{0} & \hat{\mathbf{d}} & \mathbf{D} & -\hat{\mathbf{d}}+\hat{\mathbf{a}} & \mathbf{D}+\mathbf{A}
\end{bmatrix} .$$

One point integration plus stabilization

For reduced-integration elements the transverse shear force components need to be evaluated at the center of the elements. Consider π_s the transverse shear contribution to the internal energy:

$$\pi_s = \frac{1}{2}\int \overline{\gamma}\cdot\mathbf{C}_s\cdot\overline{\gamma}\, dA .$$

The reference area measure dA is written in terms of the isoparametric coordinates as $dA = Jd\xi d\eta$, where $J = \sqrt{A_{11}A_{22}-(A_{12})^2}$ and $A_{\alpha\beta}$ are the components of the reference surface metric in the undeformed configuration.

This transverse shear energy can be approximated in many ways to produce a one point integration at the center of the element plus hourglass stabilization. It is important that this treatment yield accurate representation of transverse shear deformation in thick shell problems and provide robust performance for skewed elements. The treatment should collapse smoothly to a triangle, which should be insensitive to the node numbering during collapse; that is, the triangle's response should not depend on the nodal connectivity. For an entire mesh of triangular elements, the treatment should give convergent results (that is, the element should not lock). Furthermore, the high frequency response of the transverse shear treatment should be controlled so that transverse shear response does not dominate the stable time increment for explicit dynamic analysis (including for skewed triangular or quadrilateral geometries). All of these requirements are embodied in the following transverse shear treatment.

Define the transverse shear strain at the center of the element (the homogeneous part) and the "hourglass" transverse shear strain vectors as

$$\overline{\gamma}_0 = \frac{1}{2}\left\{ \begin{array}{c} \gamma_1^B + \gamma_1^D \\ \gamma_2^A + \gamma_2^C \end{array} \right\} + \gamma_{cc} \left\{ \begin{array}{c} c_\xi \\ c_\eta \end{array} \right\} \quad \text{and} \quad \boldsymbol{\gamma}_{hg} = \left\{ \begin{array}{c} \gamma_{bf} \\ \gamma_{cc} \end{array} \right\} .$$

The element distortion coefficients c_ξ and c_η are constants determined by the element reference geometry. For geometries with constant Jacobian transformation, $c_\xi = c_\eta = 0$. The components of the hourglass strain vector $\boldsymbol{\gamma}_{hg}$ are defined in terms of the edge strains as

$$\gamma_{bf} = -\Gamma_A \gamma_2^A + \Gamma_B \gamma_1^B + \Gamma_C \gamma_2^C - \Gamma_D \gamma_1^D \quad \text{and} \quad \gamma_{cc} = -\gamma_2^A + \gamma_1^B + \gamma_2^C - \gamma_1^D .$$

The coefficients Γ_A, Γ_B, Γ_C, and Γ_D are constants determined from the reference geometry of the element. For rectangular elements $\Gamma_A = \frac{1}{2}$, $\Gamma_B = -\frac{1}{2}$, $\Gamma_C = \frac{1}{2}$, $\Gamma_D = -\frac{1}{2}$, and γ_{bf} can be identified as the strain associated with the rotational "butterfly" deformation pattern. We call γ_{cc} the "crop circle" mode strain since it corresponds to a deformation pattern that resembles the sweeping over the element normals in a circular pattern.

The inclusion of the crop circle strain γ_{cc} in the homogeneous part of the transverse shear strain $\bar{\gamma}_0$ has two important consequences. First, it makes the transverse shear response insensitive to the nodal connectivity for a triangular element. That is, when a side of a quadrilateral element is collapsed to form a triangle, the element's response is independent of the choice of node numbering on the element. Second, for explicit dynamic analyses the coefficients c_ξ and c_η are chosen to minimize the highest frequencies associated with the homogeneous part of the transverse shear response.

To illustrate the crop circle and butterfly transverse shear patterns, consider a square, initially flat element. Furthermore, consider plate theory kinematics; that is, two rotations and a vertical deflection at the nodes. The crop circle pattern has zero vertical deflection at the nodes and a nodal rotation vector pattern as illustrated in Figure 3.6.5–2.

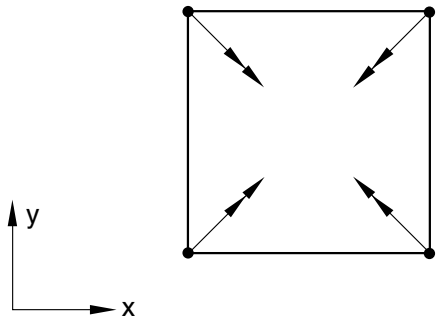


Figure 3.6.5–2 Crop circle pattern: zero deflection and circularly symmetric rotations.

The butterfly pattern has vertical deflections that correspond to cross-diagonal bending; that is, two equal deflections at two nodes across a diagonal, with equal and opposite deflections at the remaining two nodes. The nodal rotations develop in a way that opposes the bending motion of the reference surface; that is, the rotations are opposite the rotations that would develop for this displacement pattern to produce

pure bending. The butterfly mode's nodal vertical deflection and rotation vector pattern are illustrated in Figure 3.6.5–3.

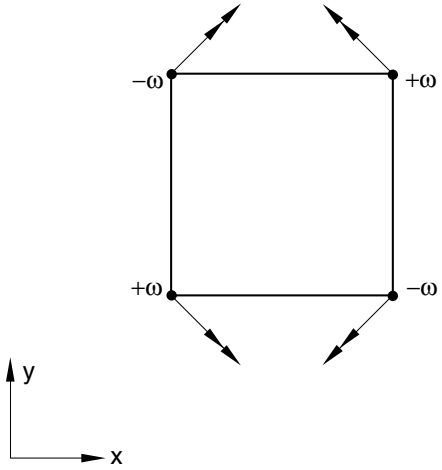


Figure 3.6.5–3 Butterfly pattern: vertical deflection and rotation vectors.

Let the reference element area be $A_0 = 4J_0$. The transverse shear energy can be approximated as a center point value plus a stabilization term:

$$\pi_s = \frac{1}{2}A_0\overline{\boldsymbol{\gamma}}_0 \cdot \mathbf{C}_{s0} \cdot \overline{\boldsymbol{\gamma}}_0 + \frac{1}{2}A_0\boldsymbol{\gamma}_{hg} \cdot \mathbf{H} \cdot \boldsymbol{\gamma}_{hg}\,,$$

where $\mathbf{C}_{s0}$ is the transverse shear stiffness evaluated at the center of the element and the hourglass stiffness $\mathbf{H}$ is the diagonal matrix

$$\mathbf{H} = \frac{C_{s0}^{eff}}{12}\begin{bmatrix} 1 & 0 \\ 0 & 0.001 \end{bmatrix}.$$

The effective stiffness C_{s0}^{eff} is the average direct component of the transverse shear stiffness, $C_{s0}^{eff} = (C_{s0}^{11} + C_{s0}^{22})/2$.

The formulation of the homogeneous part of the transverse shear has two contributions: the average edge strain across the element, plus the element distortion term. The average strain treatment is essentially the same as that for the assumed strain formulation of MacNeal and others presented earlier, with expressions evaluated at the center of the element ($\xi = 0$ and $\eta = 0$). The details of this part are omitted; only the element distortion term is presented in detail. The variation of the homogeneous transverse shear strain can be written

$$\delta\bar{\boldsymbol{\gamma}}_0 = \bar{B}_{sm0}\delta\overline{\mathbf{x}} + \bar{B}_{sb0}\delta\boldsymbol{\phi} + \bar{B}_{ccd}\delta\overline{\mathbf{x}} + \bar{B}_{ccr}\delta\boldsymbol{\phi}\,,$$

where $\bar{B}_{sm0}$ and $\bar{B}_{sb0}$ are $\bar{B}_{sm}$ and $\bar{B}_{sb}$ evaluated at the center of the element,

$$\bar{B}_{ccd} = \frac{1}{2}\begin{bmatrix} c_\xi(\mathbf{t}^A - \mathbf{t}^B)^T & c_\xi(\mathbf{t}^B - \mathbf{t}^C)^T & c_\xi(\mathbf{t}^C - \mathbf{t}^D)^T & c_\xi(\mathbf{t}^D - \mathbf{t}^A)^T \\ c_\eta(\mathbf{t}^A - \mathbf{t}^B)^T & c_\eta(\mathbf{t}^B - \mathbf{t}^C)^T & c_\eta(\mathbf{t}^C - \mathbf{t}^D)^T & c_\eta(\mathbf{t}^D - \mathbf{t}^A)^T \end{bmatrix}$$

and

$$\bar{B}_{ccr} = \frac{1}{2}\begin{bmatrix} c_\xi(\boldsymbol{\eta}_1^B - \boldsymbol{\eta}_2^A)^T & c_\xi(\boldsymbol{\eta}_1^B + \boldsymbol{\eta}_2^C)^T & c_\xi(\boldsymbol{\eta}_2^C - \boldsymbol{\eta}_1^D)^T & c_\xi(-\boldsymbol{\eta}_2^A - \boldsymbol{\eta}_1^D)^T \\ c_\eta(\boldsymbol{\eta}_1^B - \boldsymbol{\eta}_2^A)^T & c_\eta(\boldsymbol{\eta}_1^B + \boldsymbol{\eta}_2^C)^T & c_\eta(\boldsymbol{\eta}_2^C - \boldsymbol{\eta}_1^D)^T & c_\eta(-\boldsymbol{\eta}_2^A - \boldsymbol{\eta}_1^D)^T \end{bmatrix} .$$

The stabilization term has a similar formulation. The variation of the hourglass strain is

$$\delta\boldsymbol{\gamma}_{hg} = \bar{B}_d \delta\overline{\mathbf{x}} + \bar{B}_r \delta\boldsymbol{\phi} \,,$$

where

$$\bar{B}_d = \frac{1}{2}\begin{bmatrix} (\Gamma_A\mathbf{t}^A - \Gamma_B\mathbf{t}^B)^T & (\Gamma_B\mathbf{t}^B - \Gamma_C\mathbf{t}^C)^T & (\Gamma_C\mathbf{t}^C - \Gamma_D\mathbf{t}^D)^T & (\Gamma_D\mathbf{t}^D - \Gamma_A\mathbf{t}^A)^T \\ (\mathbf{t}^A - \mathbf{t}^B)^T & (\mathbf{t}^B - \mathbf{t}^C)^T & (\mathbf{t}^C - \mathbf{t}^D)^T & (\mathbf{t}^D - \mathbf{t}^A)^T \end{bmatrix}$$

and

$$\bar{B}_r = \frac{1}{2}\begin{bmatrix} (\Gamma_B\boldsymbol{\eta}_1^B - \Gamma_A\boldsymbol{\eta}_2^A)^T & (\Gamma_B\boldsymbol{\eta}_1^B + \Gamma_C\boldsymbol{\eta}_2^C)^T & (\Gamma_C\boldsymbol{\eta}_2^C - \Gamma_D\boldsymbol{\eta}_1^D)^T & (-\Gamma_A\boldsymbol{\eta}_2^A - \Gamma_D\boldsymbol{\eta}_1^D)^T \\ (\boldsymbol{\eta}_1^B - \boldsymbol{\eta}_2^A)^T & (\boldsymbol{\eta}_1^B + \boldsymbol{\eta}_2^C)^T & (\boldsymbol{\eta}_2^C - \boldsymbol{\eta}_1^D)^T & (-\boldsymbol{\eta}_2^A - \boldsymbol{\eta}_1^D)^T \end{bmatrix} .$$

The hourglass force components h^1 and h^2 are given by the constitutive relations

$$h^1 = \frac{C_{s0}^{eff}}{12}\gamma_{bf} \quad \text{and} \quad h^2 = 0.001\frac{C_{s0}^{eff}}{12}\gamma_{cc} \,.$$

Comments on stabilization

(1) The butterfly mode γ_{bf} is applied with a "large" or physical hourglass stiffness. For a reference geometry with constant Jacobian, the butterfly stabilization term can be derived from an exact integration of the assumed strain formulation of the transverse shear energy. It is important to apply this constraint with a high stiffness to prevent overly flexible response for quadrilateral elements. The crop circle mode is applied with a "small" or weak stiffness. Although this mode can propagate, it is rarely problematic and is often prevented with boundary conditions.

(2) As the quadrilateral element is degenerated to a triangle, the two hourglass constraints converge into a single constraint: the crop circle constraint. However, as is well-known, for a constant strain triangle the element will lock for certain meshes with three transverse shear constraints per element. Therefore, in the case of a triangular element, the (strong) butterfly mode stabilization is not applied. Only the (weak) crop circle mode stabilization is applied. Thus, in addition to the two homogeneous transverse shear strains, the triangle has a weak constraint to prevent spurious zero energy modes, yet avoids locking in most situations.

The initial stress contribution from the stabilization terms takes the following form:

$$A_0(h^1 \mathrm{d}\delta\gamma_{bf} + h^2 \mathrm{d}\delta\gamma_{cc}) = A_0 \delta\mathbf{u} \cdot \mathbf{K}_h \cdot \mathrm{d}\mathbf{u}\,,$$

where $\mathbf{K}_h$ is the (symmetric) transverse shear stabilization contribution to the initial stress. Define the symmetric matrices

$$\mathbf{H}^A = (h^1\Gamma_A + h^2)\left[\mathrm{sym}\{\mathbf{t}^A(\overline{\mathbf{x}}^A_{,2})^T\} - \gamma^A_2\mathbf{I}\right]\,, \quad \mathbf{H}^B = (h^1\Gamma_B + h^2)\left[\mathrm{sym}\{\mathbf{t}^B(\overline{\mathbf{x}}^B_{,1})^T\} - \gamma^B_1\mathbf{I}\right]\,,$$

$$\mathbf{H}^C = (h^1\Gamma_C + h^2)\left[\mathrm{sym}\{\mathbf{t}^C(\overline{\mathbf{x}}^C_{,2})^T\} - \gamma^C_2\mathbf{I}\right]\,, \quad \mathbf{H}^D = (h^1\Gamma_D + h^2)\left[\mathrm{sym}\{\mathbf{t}^D(\overline{\mathbf{x}}^D_{,1})^T\} - \gamma^D_1\mathbf{I}\right]\,.$$

Also define the skew-symmetric matrices

$$\hat{\mathbf{a}}_h = (h^1\Gamma_A + h^2)\hat{\mathbf{t}}^A\,, \qquad \hat{\mathbf{b}}_h = (h^1\Gamma_B + h^2)\hat{\mathbf{t}}^B\,,$$

$$\hat{\mathbf{c}}_h = (h^1\Gamma_C + h^2)\hat{\mathbf{t}}^C\,, \qquad \hat{\mathbf{d}}_h = (h^1\Gamma_D + h^2)\hat{\mathbf{t}}^D\,.$$

Then $\mathbf{K}_h$ is written

$$\mathbf{K}_h = \frac{1}{4}\begin{bmatrix}
\mathbf{0} & -\hat{\mathbf{a}}_h + \hat{\mathbf{b}}_h & \mathbf{0} & \hat{\mathbf{b}}_h & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\hat{\mathbf{a}}_h \\
\hat{\mathbf{a}}_h - \hat{\mathbf{b}}_h & -\mathbf{H}^A + \mathbf{H}^B & \hat{\mathbf{b}}_h & \mathbf{H}^B & \mathbf{0} & \mathbf{0} & -\hat{\mathbf{a}}_h & -\mathbf{H}^A \\
\mathbf{0} & -\hat{\mathbf{b}}_h & \mathbf{0} & -\hat{\mathbf{b}}_h + \hat{\mathbf{c}}_h & \mathbf{0} & \hat{\mathbf{c}}_h & \mathbf{0} & \mathbf{0} \\
-\hat{\mathbf{b}}_h & \mathbf{H}^B & \hat{\mathbf{b}}_h - \hat{\mathbf{c}}_h & \mathbf{H}^B + \mathbf{H}^C & \hat{\mathbf{c}}_h & \mathbf{H}^C & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & -\hat{\mathbf{c}}_h & \mathbf{0} & -\hat{\mathbf{c}}_h + \hat{\mathbf{d}}_h & \mathbf{0} & \hat{\mathbf{d}}_h \\
\mathbf{0} & \mathbf{0} & -\hat{\mathbf{c}}_h & \mathbf{H}^C & \hat{\mathbf{c}}_h - \hat{\mathbf{d}}_h & \mathbf{H}^C - \mathbf{H}^D & \hat{\mathbf{d}}_h & -\mathbf{H}^D \\
\mathbf{0} & \hat{\mathbf{a}}_h & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\hat{\mathbf{d}}_h & \mathbf{0} & -\hat{\mathbf{d}}_h + \hat{\mathbf{a}}_h \\
\hat{\mathbf{a}}_h & -\mathbf{H}^A & \mathbf{0} & \mathbf{0} & -\hat{\mathbf{d}}_h & -\mathbf{H}^D & \hat{\mathbf{d}}_h - \hat{\mathbf{a}}_h & -\mathbf{H}^D - \mathbf{H}^A
\end{bmatrix}.$$

Note that once the matrix entries in $\mathbf{K}_h$ are defined, $\mathbf{K}_h$ is filled just as $\mathbf{K}_s$.

The initial stress contribution from the homogeneous part consists of two terms, one from the assumed strain formulation (evaluated at $\xi = \eta = 0$) as detailed earlier, and the other from the crop circle mode addition. These two terms can be written

$$A_0\mathbf{q}|_0 \cdot \mathrm{d}\delta\bar{\gamma}_0 = A_0\delta\mathbf{u} \cdot \mathbf{K}_s|_0 \cdot \mathrm{d}\mathbf{u} + A_0(q^1 c_\xi + q^2 c_\eta)\mathrm{d}\delta\gamma_{cc}\,,$$

where $\mathbf{q}|_0$ and $\mathbf{K}_s|_0$ are the shear force and matrix $\mathbf{K}_s$ evaluated at the element center. The matrix expression for $(q^1 c_\xi + q^2 c_\eta)\mathrm{d}\delta\gamma_{cc}$ is analogous to $\mathbf{K}_h$ from the stabilization terms.

In-plane displacement hourglass control

The in-plane displacement hourglass control is applied in the same way as in the Abaqus membrane elements. The hourglass strains are defined by

$$z_\alpha = \frac{\partial\overline{\mathbf{x}}}{\partial S_\alpha} \cdot \overline{\mathbf{x}}^I\gamma^I - \frac{\partial\overline{\mathbf{X}}}{\partial S_\alpha} \cdot \overline{\mathbf{X}}^I\gamma^I = \frac{\partial\overline{\mathbf{x}}}{\partial S_\alpha} \cdot \overline{\mathbf{x}}^I\gamma^I - \mathbf{T}_\alpha \cdot \overline{\mathbf{X}}^I\gamma^I,$$

where γ^I is the hourglass mode. This mode is obtained by making the "regular" hourglass mode $\Gamma^I = \{1, -1, 1, -1\}$ orthogonal to the homogeneous deformation mode in the undeformed shape of the element. This last condition can be written as

$$\gamma^I = \Gamma^I - \Gamma^J \mathbf{X}^J \cdot \mathbf{T}_\beta \frac{\partial N^I}{\partial S_\beta}.$$

Observe that

$$\mathbf{T}_\alpha \cdot \overline{\mathbf{X}}^I \gamma^I = \mathbf{T}_\alpha \cdot \overline{\mathbf{X}}^I \Gamma^I - \Gamma^J \overline{\mathbf{X}}^J \cdot \mathbf{T}_\beta \frac{\partial \overline{\mathbf{X}}}{\partial S_\beta} \cdot \mathbf{T}_\alpha = \mathbf{T}_\alpha \cdot \overline{\mathbf{X}}^I \Gamma^I - \Gamma^J \overline{\mathbf{X}}^J \cdot \mathbf{T}_\alpha = 0,$$

and consequently

$$z_\alpha = \frac{\partial \mathbf{x}}{\partial S_\alpha} \cdot \mathbf{x}^I \gamma^I.$$

This expression can be worked out further. We define the projected nodal coordinates

$$S^I_\alpha \stackrel{\text{def}}{=} \mathbf{T}_\alpha \cdot \overline{\mathbf{X}}^I$$

and the projected element area

$$A = \frac{1}{2}\left[(S^3_1 - S^1_1)(S^4_2 - S^2_2) + (S^2_1 - S^4_1)(S^3_2 - S^1_2)\right].$$

The hourglass mode can then be written in the form

$$\begin{aligned}
\gamma^1 &= \frac{1}{A}\left[S^2_1(S^3_2 - S^4_2) + S^3_1(S^4_2 - S^2_2) + S^4_1(S^2_2 - S^3_2)\right], \\
\gamma^2 &= \frac{1}{A}\left[S^3_1(S^1_2 - S^4_2) + S^4_1(S^3_2 - S^1_2) + S^1_1(S^4_2 - S^3_2)\right], \\
\gamma^3 &= \frac{1}{A}\left[S^4_1(S^1_2 - S^2_2) + S^1_1(S^2_2 - S^4_2) + S^2_1(S^4_2 - S^1_2)\right], \\
\gamma^4 &= \frac{1}{A}\left[S^1_1(S^3_2 - S^2_2) + S^2_1(S^1_2 - S^3_2) + S^3_1(S^2_2 - S^1_2)\right].
\end{aligned}$$

The hourglass stiffness is chosen equal to

$$K_h = (r_F G)\frac{\partial N^I}{\partial S_\beta}\frac{\partial N^I}{\partial S_\beta} hA,$$

where G is the shear modulus and r_F is a small number chosen to be 0.005 in Abaqus/Standard and 0.05 in Abaqus/Explicit. When the hourglass control is based on assumed enhanced strain, the artificial stiffness factor is replaced by coefficients derived from a three-field variational principle. The hourglass force Z conjugate to z is then equal to

$$Z_\alpha = K_h z_\alpha .$$

For virtual work we need the first variation of the hourglass strain. From the expression for the strain follows immediately

$$\delta z_\alpha = \frac{\partial \overline{\mathbf{x}}}{\partial S_\alpha} \cdot \delta \overline{\mathbf{x}}^I \gamma^I + \frac{\partial \delta \overline{\mathbf{x}}}{\partial S_\alpha} \cdot \overline{\mathbf{x}}^I \gamma^I .$$

Note that the second term vanishes in the initial configuration since $\overline{\mathbf{X}}^I \gamma^I = 0$. The second variation is needed for the Jacobian. From the first variation follows right away

$$\mathrm{d}\delta z_\alpha = \frac{\partial \mathrm{d}\overline{\mathbf{x}}}{\partial S_\alpha} \cdot \delta \overline{\mathbf{x}}^I \gamma^I + \frac{\partial \delta \overline{\mathbf{x}}}{\partial S_\alpha} \cdot \mathrm{d}\overline{\mathbf{x}}^I \gamma^I .$$

The second variation does not contribute in the initial configuration since initially $Z_\alpha = 0$.

Rotational hourglass control

The expressions for the curvature change, the transverse shear constraints, and the drilling mode constraints still leave three nonhomogeneous rotational modes unconstrained. These modes correspond to zero rotation at the midedges and zero gradient at the centroid. Hence, they correspond to the familiar $\Gamma^I = \{1, -1, 1, -1\}$ hourglass pattern. To pass curvature patch tests exactly, it is necessary to use orthogonalized hourglass patterns as derived for in-plane hourglass control.

This last aspect implies that the rotational hourglass mode corresponds to the mixed derivative of the rotation at the centroid:

$$\gamma^I \Delta \boldsymbol{\phi}^I = 4 \times \frac{\partial^2 \Delta \boldsymbol{\phi}}{\partial \xi_1 \partial \xi_2}.$$

We cannot use the above formulation directly in a formulation suitable for multiple finite rotation increments. Hence, we use the same approach as for the calculation of the curvature change. For the purpose of the calculation we define the updated shell direction vectors

$$\tilde{\mathbf{t}}_i^{t+\Delta t} = \Delta \boldsymbol{q}\, \mathbf{t}_i^t \Delta \boldsymbol{q}^\dagger .$$

The updated shell direction vectors do not actually have to be stored: they are used only for the derivation of the expression for the hourglass strain. We now formally define the hourglass strain tensor as

$$\vartheta_{ij} = \mathbf{t}_i \cdot \frac{\partial^2 \mathbf{t}_j}{\partial \xi_1 \partial \xi_2}.$$

Observe that

$$\frac{\partial^2 (\mathbf{t}_i \cdot \mathbf{t}_j)}{\partial \xi_1 \partial \xi_2} = \mathbf{t}_i \cdot \frac{\partial^2 \mathbf{t}_j}{\partial \xi_1 \partial \xi_2} + \frac{\partial^2 \mathbf{t}_i}{\partial \xi_1 \partial \xi_2} \cdot \mathbf{t}_j + \frac{\partial \mathbf{t}_i}{\partial \xi_1} \cdot \frac{\partial \mathbf{t}_j}{\partial \xi_2} + \frac{\partial \mathbf{t}_i}{\partial \xi_2} \cdot \frac{\partial \mathbf{t}_j}{\partial \xi_1} = 0.$$

For the purpose of hourglass strain calculation we assume that all products of first-order derivatives with respect to ξ_1 and ξ_2 can be neglected. Consequently,

$$\frac{\partial^2(\mathbf{t}_i \cdot \mathbf{t}_j)}{\partial\xi_1\partial\xi_2} = \mathbf{t}_i \cdot \frac{\partial^2 \mathbf{t}_j}{\partial\xi_1\partial\xi_2} + \frac{\partial^2 \mathbf{t}_i}{\partial\xi_1\partial\xi_2} \cdot \mathbf{t}_j = 0;$$

and, hence,

$$\vartheta_{ij} = -\vartheta_{ji}$$

is skew-symmetric. Observe that the mixed derivative of $\mathbf{t}_i$ can be expressed in terms of the hourglass strain tensor with

$$\mathbf{t}_i\vartheta_{ij} = \mathbf{t}_i\mathbf{t}_i \cdot \frac{\partial^2 \mathbf{t}_j}{\partial\xi_1\partial\xi_2} = \frac{\partial^2 \mathbf{t}_j}{\partial\xi_1\partial\xi_2}.$$

In the undeformed configuration, we assume that $\vartheta_{ij} = 0$. Subsequent values of ϑ_{ij} are obtained incrementally. From the expression for $\tilde{\mathbf{t}}_i^{t+\Delta t}$ we obtain

$$\begin{aligned}\frac{\partial^2\tilde{\mathbf{t}}_i^{t+\Delta t}}{\partial\xi_1\partial\xi_2} =& \frac{\partial^2\Delta\boldsymbol{q}}{\partial\xi_1\partial\xi_2}\mathbf{t}_i^t\Delta\boldsymbol{q}^\dagger + \Delta\boldsymbol{q}\,\mathbf{t}_i^t\frac{\partial\Delta\boldsymbol{q}^\dagger}{\partial\xi_1\partial\xi_2} + \Delta\boldsymbol{q}\frac{\partial\mathbf{t}_i^t}{\partial\xi_1\partial\xi_2}\Delta\boldsymbol{q}^\dagger + \\ & \frac{\partial\Delta\boldsymbol{q}}{\partial\xi_1}\mathbf{t}_i^t\frac{\partial\Delta\boldsymbol{q}^\dagger}{\partial\xi_2} + \frac{\partial\Delta\boldsymbol{q}}{\partial\xi_2}\mathbf{t}_i^t\frac{\partial\Delta\boldsymbol{q}^\dagger}{\partial\xi_1} + \frac{\partial\Delta\boldsymbol{q}}{\partial\xi_1}\frac{\partial\mathbf{t}_i^t}{\partial\xi_2}\Delta\boldsymbol{q}^\dagger + \\ & \frac{\partial\Delta\boldsymbol{q}}{\partial\xi_2}\frac{\partial\mathbf{t}_i^t}{\partial\xi_1}\Delta\boldsymbol{q}^\dagger + \Delta\boldsymbol{q}\frac{\partial\mathbf{t}_i^t}{\partial\xi_1}\frac{\partial\Delta\boldsymbol{q}^\dagger}{\partial\xi_2} + \Delta\boldsymbol{q}\frac{\partial\mathbf{t}_i^t}{\partial\xi_2}\frac{\partial\Delta\boldsymbol{q}^\dagger}{\partial\xi_1}.\end{aligned}$$

In this expression we also ignore all terms with products of derivatives with respect to ξ_1 and ξ_2. Hence, the above expression simplifies to

$$\frac{\partial\tilde{\mathbf{t}}_i^{t+\Delta t}}{\partial\xi_1\partial\xi_2} = \frac{\partial^2\Delta\boldsymbol{q}}{\partial\xi_1\partial\xi_2}\mathbf{t}_i^t\Delta\boldsymbol{q}^\dagger + \Delta\boldsymbol{q}\,\mathbf{t}_i^t\frac{\partial\Delta\boldsymbol{q}^\dagger}{\partial\xi_1\partial\xi_2} + \Delta\boldsymbol{q}\frac{\partial\mathbf{t}_i^t}{\partial\xi_1\partial\xi_2}\Delta\boldsymbol{q}^\dagger.$$

The second term on the right-hand side can be written in the form

$$\Delta\boldsymbol{q}\,\mathbf{t}_i^t\frac{\partial\Delta\boldsymbol{q}^\dagger}{\partial\xi_1\partial\xi_2} = \left[\frac{\partial\Delta\boldsymbol{q}}{\partial\xi_1\partial\xi_2}\left(\mathbf{t}_i^t\right)^\dagger\Delta\boldsymbol{q}^\dagger\right]^\dagger = -\left[\frac{\partial\Delta\boldsymbol{q}}{\partial\xi_1\partial\xi_2}\mathbf{t}_i^t\Delta\boldsymbol{q}^\dagger\right]^\dagger.$$

Hence, the scalar parts of the first two terms cancel each other and the vector parts reinforce each other, leading to

$$\frac{\partial\tilde{\mathbf{t}}_i^{t+\Delta t}}{\partial\xi_1\partial\xi_2} = 2\mathbf{V}\left(\frac{\partial\Delta\boldsymbol{q}}{\partial\xi_1\partial\xi_2}\mathbf{t}_i^t\Delta\boldsymbol{q}^\dagger\right) + \Delta\boldsymbol{q}\frac{\partial\mathbf{t}_i^t}{\partial\xi_1\partial\xi_2}\Delta\boldsymbol{q}^\dagger.$$

The inverse of a rotation quaternion such as $\Delta\boldsymbol{q}$ is equal to its conjugate ($\Delta\boldsymbol{q}^{-1} = \Delta\boldsymbol{q}^\dagger$); hence, we can write

$$\begin{aligned}\frac{\partial\tilde{\mathbf{t}}_i^{t+\Delta t}}{\partial\xi_1\partial\xi_2} &= 2\mathbf{V}\left(\frac{\partial\Delta\boldsymbol{q}}{\partial\xi_1\partial\xi_2}\Delta\boldsymbol{q}^\dagger\Delta\boldsymbol{q}\,\mathbf{t}_i^t\Delta\boldsymbol{q}^\dagger\right)+\Delta\boldsymbol{q}\frac{\partial\mathbf{t}_i^t}{\partial\xi_1\partial\xi_2}\Delta\boldsymbol{q}^\dagger\\ &= \mathbf{V}\left(2\frac{\partial\Delta\boldsymbol{q}}{\partial\xi_1\partial\xi_2}\Delta\boldsymbol{q}^\dagger\tilde{\mathbf{t}}_i^{t+\Delta t}\right)+\Delta\boldsymbol{q}\frac{\partial\mathbf{t}_i^t}{\partial\xi_1\partial\xi_2}\Delta\boldsymbol{q}^\dagger\\ &= \Delta\mathbf{R}_\Gamma\times\tilde{\mathbf{t}}_i^{t+\Delta t}+\Delta\boldsymbol{q}\frac{\partial\mathbf{t}_i^t}{\partial\xi_1\partial\xi_2}\Delta\boldsymbol{q}^\dagger,\end{aligned}$$

where we have formally defined the incremental hourglass update vector

$$\Delta\mathbf{R}_\Gamma\overset{\text{def}}{=}\mathbf{V}\left(2\frac{\partial\Delta\boldsymbol{q}}{\partial\xi_1\partial\xi_2}\Delta\boldsymbol{q}^\dagger\right),$$

which must be expressed in terms of the incremental rotation hourglass mode. From the definition of the incremental quaternion $\Delta\boldsymbol{q}$ follows, while neglecting the products of ξ_1 and ξ_2 derivatives:

$$\frac{\partial\Delta\boldsymbol{q}}{\partial\xi_1\partial\xi_2}=\left(\frac{1}{2}\sin\frac{\Delta\phi}{2}\frac{\partial\Delta\phi}{\partial\xi_1\partial\xi_2},\,\frac{1}{2}\cos\frac{\Delta\phi}{2}\mathbf{p}\frac{\partial\Delta\phi}{\partial\xi_1\partial\xi_2}+\sin\frac{\Delta\phi}{2}\frac{\partial\mathbf{p}}{\partial\xi_1\partial\xi_2}\right);$$

thus, for $\Delta\mathbf{R}_\Gamma$ again with use of the incremental quaternion definition

$$\Delta\mathbf{R}_\Gamma=\mathbf{p}\frac{\partial\Delta\phi}{\partial\xi_1\partial\xi_2}+\sin\Delta\phi\,\frac{\partial\mathbf{p}}{\partial\xi_1\partial\xi_2}+(1-\cos\Delta\phi)\,\mathbf{p}\times\frac{\partial\mathbf{p}}{\partial\xi_1\partial\xi_2}.$$

From the definition of $\Delta\phi$ and $\mathbf{p}$ follows, again neglecting the products of first derivatives

$$\begin{aligned}\frac{\partial\Delta\phi}{\partial\xi_1\partial\xi_2}&=\frac{1}{\Delta\phi}\Delta\boldsymbol{\phi}\cdot\frac{\partial\Delta\boldsymbol{\phi}}{\partial\xi_1\partial\xi_2},\\ \frac{\partial\mathbf{p}}{\partial\xi_1\partial\xi_2}&=\frac{1}{\Delta\phi}\frac{\partial\Delta\boldsymbol{\phi}}{\partial\xi_1\partial\xi_2}-\frac{1}{\Delta\phi^3}\Delta\boldsymbol{\phi}\,\Delta\boldsymbol{\phi}\cdot\frac{\partial\Delta\boldsymbol{\phi}}{\partial\xi_1\partial\xi_2}=\frac{1}{\Delta\phi}\left[\frac{\partial\Delta\boldsymbol{\phi}}{\partial\xi_1\partial\xi_2}-\mathbf{p}\,\mathbf{p}\cdot\frac{\partial\Delta\boldsymbol{\phi}}{\partial\xi_1\partial\xi_2}\right].\end{aligned}$$

After substitution in the expression for $\Delta\mathbf{R}_\Gamma$ and some algebra one obtains

$$\Delta\mathbf{R}_\Gamma=\frac{\sin\Delta\phi}{\Delta\phi}\frac{\partial\Delta\boldsymbol{\phi}}{\partial\xi_1\partial\xi_2}+\frac{1-\cos\Delta\phi}{\Delta\phi}\mathbf{p}\times\frac{\partial\Delta\boldsymbol{\phi}}{\partial\xi_1\partial\xi_2}+\left(1-\frac{\sin\Delta\phi}{\Delta\phi}\right)\mathbf{p}\,\mathbf{p}\cdot\frac{\partial\Delta\boldsymbol{\phi}}{\partial\xi_1\partial\xi_2}.$$

Note that $\Delta\mathbf{R}_\Gamma\to\dfrac{\partial\Delta\boldsymbol{\phi}}{\partial\xi_1\partial\xi_2}$ when $\Delta\phi\to 0$. For the updated hourglass tensor one readily obtains

$$\begin{aligned}\vartheta_{ij}^{t+\Delta t}&=\tilde{\mathbf{t}}_i^{t+\Delta t}\cdot\frac{\partial^2\tilde{\mathbf{t}}_j^{t+\Delta t}}{\partial\xi_1\partial\xi_2}=\tilde{\mathbf{t}}_i^{t+\Delta t}\cdot\left[\Delta\mathbf{R}_\Gamma\times\tilde{\mathbf{t}}_j^{t+\Delta t}+\Delta\boldsymbol{q}\frac{\partial^2\mathbf{t}_j^t}{\partial\xi_1\partial\xi_2}\Delta\boldsymbol{q}^\dagger\right]\\ &=-\Delta\mathbf{R}_\Gamma\cdot(\tilde{\mathbf{t}}_i^{t+\Delta t}\times\tilde{\mathbf{t}}_j^{t+\Delta t})+\tilde{\mathbf{t}}_i^{t+\Delta t}\cdot\left(\Delta\boldsymbol{q}\,\mathbf{t}_k^t\vartheta_{kj}^t\Delta\boldsymbol{q}^\dagger\right)\\ &=-\Delta\mathbf{R}_\Gamma\cdot(\tilde{\mathbf{t}}_i^{t+\Delta t}\times\tilde{\mathbf{t}}_j^{t+\Delta t})+\tilde{\mathbf{t}}_i^{t+\Delta t}\cdot\tilde{\mathbf{t}}_k^{t+\Delta t}\vartheta_{kj}^t=-\Delta\mathbf{R}_\Gamma\cdot(\tilde{\mathbf{t}}_i^{t+\Delta t}\times\tilde{\mathbf{t}}_j^{t+\Delta t})+\vartheta_{ij}^t.\end{aligned}$$

This expression simplifies further with the introduction of the hourglass vector

$$\vartheta_i = -\frac{1}{2} e_{ijk} \vartheta_{jk},$$

which yields the update formula

$$\vartheta_i^{t+\Delta t} = \Delta \mathbf{R}_\Gamma \cdot \tilde{\mathbf{t}}_i^{t+\Delta t} + \vartheta_i^t.$$

The first and second variation are obtained in entirely the same way as the first and second variation of the curvature change. For the first variation we neglect terms of order $\Delta \mathbf{R}_\Gamma$ and obtain

$$\delta\vartheta_i = \delta\Delta\mathbf{R}_\Gamma \cdot \mathbf{t}_i + \Delta\mathbf{R}_\Gamma \cdot \delta\mathbf{t}_i = \delta\Delta\mathbf{R}_\Gamma \cdot \mathbf{t}_i = \mathbf{t}_i \cdot \frac{\partial^2 \delta\boldsymbol{\phi}}{\partial\xi_1 \partial\xi_2}.$$

For the second variation we ignore in addition the terms of order $\mathrm{d}\mathbf{R}_\Gamma$ and $\delta\mathbf{R}_\Gamma$ with as final result

$$\mathrm{d}\delta\vartheta_i = 0.$$

Degenerate elements

In general meshes it will be desirable to collapse at least some of the quadrilateral elements to triangles or to use the triangular element S3 or S3R, which is in fact an internally collapsed S4R element. For this case the calculation of the membrane strains and the curvature changes proceeds along the same lines as before. The transverse shears will now be zero at the degenerate edges. Finally, calculation of all hourglass constraints will be omitted.

Rotary inertia scaling for explicit dynamics

For numerical efficiency in explicit dynamic analysis, it is desirable to have the stable time increment determined by the membrane response of the structure. For this reason scaling of the rotary inertia based on the element's reference geometry is included in Abaqus/Explicit.

In explicit dynamic analyses the stable time increment is proportional to the inverse of the highest frequency of the element. Therefore, we must ensure that the highest frequency associated with the transverse shear response does not exceed the highest frequency associated with the membrane response. For thick elements (that is, for elements whose thickness is order unity relative to a characteristic length in the element), the membrane frequencies are dominant. The primary consideration in choosing appropriate scalings is that in the limit as the element's thickness goes to zero, the transverse shear frequencies remain below the membrane ones. Recall that for a one-dimensional spring-mass oscillator, the natural frequency ω can be written in terms of the stiffness $\boldsymbol{K}$ and the mass $\boldsymbol{M}$ as

$$\omega = \sqrt{\frac{K}{M}}\,.$$

For the transverse shear response the rotary inertia, which is proportional to the cube of the thickness, plays the role of the mass of the system. All other quantities—the membrane stiffness, the mass

associated with membrane deformation, and the transverse shear stiffness—are proportional to the thickness. Hence as the thickness of the element goes to zero, the frequencies associated with transverse shear go to infinity proportional to the inverse of the thickness, while the membrane frequencies remain constant. Without scaling, the stable time increment would go to zero as the thickness becomes small.

Rotary inertia scaling

For thin elements the rotary inertia is small (negligible) relative to the rotational inertia of the mass at the nodes rotating about an axis through the element center. Therefore, we choose a scaling on the rotary inertia such that it never becomes smaller than a fixed (small) percentage of the rotational inertia of the mass at the nodes rotating about an axis through the center of the element.

Let $\boldsymbol{R}$ be the nondimensional rotary inertia scaling, where $R \geq 1$. When $R = 1$, the true rotary inertia is used. Consider a lumped mass matrix for a 4-node element, and let the element be flat and square. For rotations about an axis in the plane of the element, parallel to an element edge, passing through the center, the contribution to the rotational inertia of the mass at the nodes is

$$\frac{1}{4}\rho A h L^2 \,,$$

where $\boldsymbol{A}$ is the area of the element, $\boldsymbol{L}$ is the edge length, and ρ is the mass density. The sum of the rotary inertia at the four nodes is

$$R\rho A\frac{h^3}{12} \,.$$

The ratio of the in-plane contribution to the rotary contribution is

$$\frac{3L^2}{Rh^2} \,.$$

For the rotary inertia to remain a fixed fraction—say ϵ—of the mass contribution as the thickness goes to zero, asymptotically $\boldsymbol{R}$ must be proportional to L^2/h^2; that is,

$$R \to \epsilon\frac{3L^2}{h^2} \,.$$

For planar geometries with element directors along the normal direction, closed-form expressions are possible for the highest membrane and transverse shear frequencies. In such cases the length parameter $\boldsymbol{L}$ is interpreted as a characteristic element length that depends on the element distortion. To handle arbitrarily shaped curved elements, exact calculation of the element frequencies becomes difficult. However, we can safely bound the frequencies by an appropriate choice of $\boldsymbol{L}$ in the following scaling:

$$R = \left[16 + \epsilon^4\left(\frac{3L^2}{h^2}\right)^4\right]^{\frac{1}{4}} \,,$$

where $\epsilon = 1/4$. For triangular elements the characteristic length, L, is the minimum element edge length; and for quadrilateral elements,

$$L = \frac{2\|\mathbf{x}_{,\xi} \times \mathbf{x}_{,\eta}\|}{\max(\|\mathbf{x}_{,\xi}\|, \|\mathbf{x}_{,\eta}\|)} .$$

The factor 16 in the definition of R is used to protect against bending frequencies determining the stable time increment in very fine meshes subjected to loads that cause an increase in thickness of the shell.

Rotational bulk viscosity for explicit dynamics

For the displacement degrees of freedom, bulk viscosity introduces damping associated with volumetric straining. Linear bulk viscosity or truncation frequency damping is used to damp the high frequency ringing that leads to unwanted noise in the solution or spurious overshoot in the response amplitude. For the same reason, in shells we need to damp the high frequency ringing in the rotational degrees of freedom with linear bulk viscosity acting on the mean curvature strain rate. This damping generates a bulk viscosity "pressure moment," m, which is linear in the mean curvature strain rate:

$$m = b\frac{h_0^2}{12}\rho c_d L\frac{\Delta\kappa}{\Delta t} ,$$

where b is a damping coefficient (default = 0.06), h_0 is the original thickness, ρ is the mass density, c_d is the current dilatational wave speed, L is the characteristic length used for rotary inertia scaling, and $\Delta\kappa = \Delta\kappa_{11} + \Delta\kappa_{22}$ is twice the increment in mean curvature. The dilatational wave speed is given in terms of the effective Lamé's constants as

$$c_d = \sqrt{\frac{(\lambda + 2\mu)}{\rho}} .$$

The resultant pressure moment mh, where h is the current thickness, is added to the direct components of the moment resultant.

Fully integrated finite-membrane-strain shell formulation

Element S4 is a fully integrated finite-membrane-strain shell element. Since the element's stiffness is fully integrated, no spurious membrane or bending zero energy modes exist and no membrane or bending mode hourglass stabilization is used. Drill rotation control, however, is required. Element S4 uses the same drill stiffness formulation as used for element S4R. Similarly, element S4 assumes that the transverse shear strain (and force, since the transverse shear treatment is elastic based on the initial elastic modulus of the material) is constant over the element. Therefore, all four stiffness integration locations will have the same transverse shear strain, transverse shear section force, and transverse shear stress distribution. The transverse shear treatment for S4 is identical to that for S4R.

It is well known that a standard displacement formulation will exhibit shear locking for applications dominated by in-plane bending deformation. However, a standard displacement formulation for the out-of-plane bending stiffness is not subject to similar locking response. Hence, S4 uses a standard

displacement formulation for the element's bending stiffness, and the theory presented above for the rotation kinematics and bending strain measures applies to S4. The primary difference between the element formulations for S4 and S4R is the treatment of the membrane strain field. This formulation is the topic of the following discussion.

The membrane formulation used for S4 does not rely on the fact that S4 is a shell element. Hence, the discussion below details the formulation from the point of view that the membrane response is governed by the equilibrium for a three-dimensional body in a state of plane stress.

Consider an *enhancement* $\tilde{\mathbf{D}}$ to the rate of deformation tensor $\mathbf{D}$. We introduce the enhanced rate of deformation tensor, $\bar{\mathbf{D}}$, as

$$\bar{\mathbf{D}} \stackrel{\text{def}}{=} \mathbf{D} + \tilde{\mathbf{D}}\,,$$

where $\tilde{\mathbf{D}}$ is defined subsequently.

Admissible variations in the rate of deformation are also introduced as

$$\bar{\delta}\mathbf{D} \stackrel{\text{def}}{=} \delta\mathbf{D} + \delta\tilde{\mathbf{D}}\,,$$

where

$$\delta\mathbf{D} \stackrel{\text{def}}{=} \text{sym}\left(\frac{\partial \delta\mathbf{x}}{\partial \mathbf{x}}\right).$$

We now introduce constraints on the enhancements $\mathbf{D}$ and $\tilde{\mathbf{D}}$:

$$\int_V \left(\bar{\mathbf{D}} - \mathbf{D}\right) : \overset{*}{\boldsymbol{\sigma}}\, dV = \int_V \left(\delta\bar{\mathbf{D}} - \delta\mathbf{D}\right) : \overset{*}{\boldsymbol{\sigma}}\, dV = 0$$

so that the modified virtual work statement can be written in the form

$$\int_V \boldsymbol{\sigma}(\bar{\mathbf{D}}) : \delta\bar{\mathbf{D}}\, dV + \int_V \mathbf{b}\cdot\delta\mathbf{x}\, dV - \int_{S_T} \mathbf{t}\cdot\delta\mathbf{x}\, dS_T = 0,$$

where $\mathbf{t}$ is the specified traction on S_T and $\delta\mathbf{x} = \mathbf{0}$ on S_u. $\overset{*}{\boldsymbol{\sigma}}$ is an arbitrary stress field, and the constitutive equation $\boldsymbol{\sigma} = \boldsymbol{\sigma}\,(\bar{\mathbf{D}})$ is enforced pointwise.

In the modified virtual work statement all kinematic quantities and corresponding variations ($\bar{\mathbf{D}}$, $\mathbf{D}$, $\delta\bar{\mathbf{D}}$, and $\delta\mathbf{D}$) are known functions of $\mathbf{x}$, $\delta\mathbf{x}$, and the reference configuration. A fundamental requirement for the validity of the formulation is that the modified virtual work statement leads to the proper equilibrium equations. If $\overset{*}{\boldsymbol{\sigma}}$ is arbitrary, the constraint equations can be rewritten as

$$\bar{\mathbf{D}} = \mathbf{D} \quad \text{and} \quad \delta\bar{\mathbf{D}} = \delta\mathbf{D}\,.$$

Substituting these two relations in the modified virtual work equation yields

$$\int_V \boldsymbol{\sigma} : \delta\mathbf{D}\, dV + \int_V \mathbf{b}\cdot\delta\mathbf{x}\, dV - \int_{S_T} \mathbf{t}\cdot\delta\mathbf{x}\, dS_T = 0$$

where we have used the constitutive equation $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\mathbf{D})$. We recognize this variational statement as the usual virtual work equation, and a straightforward application of the divergence theorem leads to the standard equilibrium equations.

In the actual implementation we choose to satisfy the constraints only for piecewise constant stress fields $\overset{*}{\boldsymbol{\sigma}}$. Hence, over the element domain V_e we require

$$\int_{V^e} \tilde{\mathbf{D}}\, dV = \int_{V^e} \delta\tilde{\mathbf{D}}\, dV = 0.$$

The enhancements $\tilde{\mathbf{D}}$ and $\delta\tilde{\mathbf{D}}$ are chosen such that they eliminate the shear locking for in-plane bending. In addition, the direct strain field is enhanced to approximate the strains caused by Poisson's effect in bending.

Patch test

To pass the patch test, the choice of enhancements $\tilde{\mathbf{D}}$ cannot be arbitrary. A sufficient condition for the satisfaction of the patch test is that for homogeneous deformations we have $\bar{\mathbf{D}} = \mathbf{D}$ or $\tilde{\mathbf{D}} = \mathbf{0}$ pointwise. In that case $\sigma(\bar{\mathbf{D}}) = \sigma(\mathbf{D})$ and the stress is homogeneous. Since the stress is homogeneous, it can be moved outside the volume integral in the modified virtual work statement. The volume integral condition on $\delta\tilde{\mathbf{D}}$ implies that the expression is independent of the enhancement and leads to the standard displacement formulation, which is known to satisfy the patch test.

References

- "Shell elements: overview," Section 29.6.1 of the Abaqus Analysis User's Guide
- "Shell section behavior," Section 29.6.4 of the Abaqus Analysis User's Guide

3.6.6 SMALL-STRAIN SHELL ELEMENTS IN Abaqus/Explicit

Product: Abaqus/Explicit

The small-strain shell elements in Abaqus/Explicit use a Mindlin-Reissner type of flexural theory that includes transverse shear and are based on a corotational velocity-strain formulation described by Belytschko et al. (1984, 1992). A corotational finite element formulation reduces the complexities of nonlinear mechanics by embedding a local coordinate system in each element at the sampling point of that element. By expressing the element kinematics in a local coordinate frame, the number of computations is reduced substantially. Therefore, the corotational velocity-strain formulation provides significant speed advantages in explicit time integration software, where element computations can dominate during the overall solution process.

Corotational coordinate system

The geometry of the shell is defined by its reference surface, which is determined by the nodal coordinates of the element. The embedded element corotational coordinate system, $\hat{\mathbf{x}}$, is tangent to the reference surface and rotates with the element. This embedded corotational coordinate system serves as a local coordinate system and is constructed as follows:

1. For the quadrilateral element the local coordinate $\hat{x}_1$ is coincident with the line connecting the midpoints of sides, $\mathbf{r}_{ac}$, as shown in Figure 3.6.6–1. The $\hat{x}_1$–$\hat{x}_2$ plane is defined to pass through this line normal to the cross product $\mathbf{r}_{ac} \times \mathbf{r}_{bd}$.
2. For the triangular element the local coordinate $\hat{x}_1$ is coincident with the side connecting nodes 1 and 2 as shown in Figure 3.6.6–1. The $\hat{x}_1$–$\hat{x}_2$ plane coincides with the plane of the element.

For notational purposes the corotational coordinate system is defined by a triad $\hat{e}_i$, and any vector or tensor whose components are expressed in this system will bear a superposed "hat."

Although the corotational coordinate system described here is used in the actual element computations, this system is transparent to the user. All reported stresses, strains, and other tensorial quantities for these shell elements are defined with respect to the coordinate system described in "Finite-strain shell element formulation," Section 3.6.5.

Velocity strain formulation

The velocity of any point in the shell reference surface is given in terms of the discrete nodal velocity with the bilinear isoparametric shape functions $N^I(\xi_\alpha)$ as

$$\mathbf{v}^m = N^I(\xi_\alpha)\mathbf{v}^I,$$

$$\theta = N^I(\xi_\alpha)\theta^I,$$

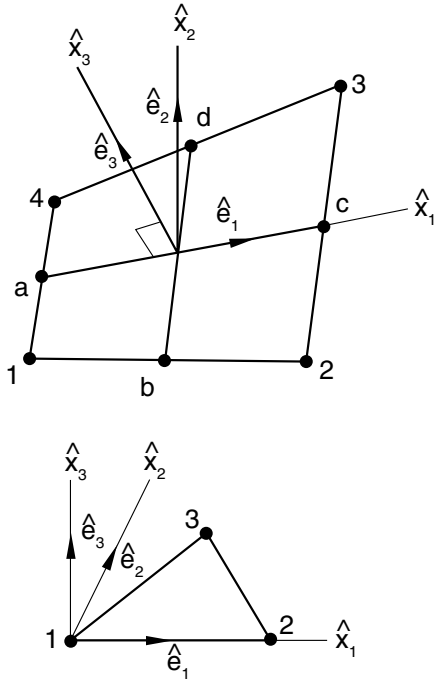


Figure 3.6.6–1 Local coordinate system for small-strain quadrilateral and triangular shell elements.

where $\mathbf{v}^I$ and θ^I are the nodal translation and rotation velocity, respectively. The functions $N^I(\xi_\alpha)$ are $C_\circ$ continuous, and ξ_α are nonorthogonal, nondistance measuring parametric coordinates. Here Greek subscripts range from 1 to 2, and uppercase Roman superscripts denote the nodes of an element. A standard summation convention is used for repeated superscripts and subscripts except where noted otherwise.

In the Mindlin-Reissner theory of plates and shells, the velocity of any point in the shell is defined by the velocity of the reference surface, $\hat{\mathbf{v}}^m$, and the angular velocity vector, $\hat{\theta}$, as

$$\hat{\mathbf{v}} = \hat{\mathbf{v}}^m - \hat{x}_3 \hat{e}_3 \times \hat{\theta},$$

where $\times$ denotes the vector cross product and $\hat{x}_3$ is the distance in the normal direction through the thickness of the shell element. The corotational components of the velocity strain (rate of deformation) are given by

$$\hat{d}_{ij} = \frac{1}{2}\left(\frac{\partial \hat{v}_i}{\partial \hat{x}_1 j} + \frac{\partial \hat{v}_j}{\partial \hat{x}_1 i}\right),$$

which allows us to write each velocity strain component in terms of the nodal translational and rotational velocities:

$$\hat{d}_1 = \frac{\partial \hat{v}_1^m}{\partial \hat{x}_1} + \hat{x}_3 \frac{\partial \hat{\theta}_1}{\partial \hat{x}_1},$$

$$\hat{d}_2 = \frac{\partial \hat{v}_2^m}{\partial \hat{x}_2} - \hat{x}_3 \frac{\partial \hat{\theta}_2}{\partial \hat{x}_2},$$

$$\hat{d}_{12} = \frac{\partial \hat{v}_1^m}{\partial \hat{x}_2} + \frac{\partial \hat{v}_2^m}{\partial \hat{x}_1} + \hat{x}_3 \left(\frac{\partial \hat{\theta}_2}{\partial \hat{x}_2} - \frac{\partial \hat{\theta}_1}{\partial \hat{x}_1} \right),$$

$$\hat{d}_{23} = \frac{\partial \hat{v}_3^m}{\partial \hat{x}_2} - \hat{\theta}_1,$$

$$\hat{d}_{13} = \frac{\partial \hat{v}_3^m}{\partial \hat{x}_1} + \hat{\theta}_2.$$

Small-strain element S4RS

The S4RS element is based on Belytschko et al. (1984). By using one-point quadrature at the center of the element—i.e., at ξ_α=0—we obtain the gradient operator

$$\begin{Bmatrix} B_1^I \\ B_2^I \end{Bmatrix} = \frac{1}{2A} \begin{bmatrix} \hat{y}_2 - \hat{y}_4 & \hat{y}_3 - \hat{y}_1 & \hat{y}_4 - \hat{y}_2 & \hat{y}_1 - \hat{y}_3 \\ \hat{x}_4 - \hat{x}_2 & \hat{x}_1 - \hat{x}_3 & \hat{x}_2 - \hat{x}_4 & \hat{x}_3 - \hat{x}_1 \end{bmatrix}.$$

The velocity strain can then be expressed as

$$\hat{d}_1 = B_1^I \hat{v}_1^I + \hat{x}_3 B_1^I \hat{\theta}_2^I,$$

$$\hat{d}_2 = B_2^I \hat{v}_2^I - \hat{x}_3 B_2^I \hat{\theta}_1^I,$$

$$2\hat{d}_{12} = B_2^I \hat{v}_1^I + B_1^I \hat{v}_2^I + \hat{x}_3 (B_2^I \hat{\theta}_2^I - B_1^I \hat{\theta}_1^I),$$

$$2\hat{d}_{13} = B_1^I \hat{v}_3^I + \frac{1}{4} S^I \hat{\theta}_2^I,$$

$$2\hat{d}_{23} = B_2^I \hat{v}_3^I - \frac{1}{4} S^I \hat{\theta}_1^I,$$

where

$$S^I = [1, 1, 1, 1].$$

The local nodal forces and moments are computed in terms of the section force and moment resultants by

$$\hat{f}_1^I = A\left(B_1^I \hat{f}_1 + B_2^I \hat{f}_{12}\right),$$

$$\hat{f}_2^I = A\left(B_2^I \hat{f}_2 + B_1^I \hat{f}_{12}\right),$$

$$\hat{f}_3^I = \kappa A\left(B_1^I \hat{f}_{13} + B_2^I \hat{f}_{23}\right),$$

$$\hat{m}_1^I = A\left(B_2^I \hat{m}_2 + B_1^I \hat{m}_{12} - \frac{\kappa}{4} S^I \hat{f}_{23}\right),$$

$$\hat{m}_2^I = -A\left(B_1^I \hat{m}_1 + B_2^I \hat{m}_{12} - \frac{\kappa}{4} S^I \hat{f}_{13}\right),$$

$$\hat{m}_3^I = 0.$$

The section force and moment resultants are given by

$$\hat{f}_{\alpha\beta} = \int_h \hat{\sigma}_{\alpha\beta} d\hat{z},$$

$$\hat{m}_{\alpha\beta} = -\int_h \hat{z}\hat{\sigma}_{\alpha\beta} d\hat{z},$$

where A is the area of the element, h is the thickness, and $\hat{\sigma}_{\alpha\beta}$ are the Cauchy stresses computed in the corotational system from the velocity strain and the applicable constitutive model. Although κ is the shear factor in classical Mindlin-Reissner plate theory, it is used here as a penalty parameter to enforce the Kirchhoff normality condition as the shell becomes thin.

Small-strain element S4RSW

The major objective in the development of the S4RS element was to obtain a convergent, stable element with the minimum number of computations. Because of the emphasis on speed, a few simplifications were made in formulating the equations for the S4RS element. Although the S4RS element performs very well in most practical applications, it has two known shortcomings:

1. It can perform poorly when warped, and in particular, it does not solve the twisted beam problem correctly.
2. It does not pass the bending patch test in the thin plate limit.

In the S4RSW element additional terms are added to the strain-displacement equations to eliminate the first shortcoming, and a shear projection is used in the calculation of the transverse shear to address the second shortcoming. The components of velocity strain in the S4RSW element are given in Belytschko et al. (1992) as

$$\hat{d}_1 = B_1^I \hat{v}_1^I + \hat{x}_3 (B_1^{cI} \hat{v}_1^I + B_1^I \dot{p}_1^I),$$

$$\hat{d}_2 = B_2^I \hat{v}_2^I + \hat{x}_3 (B_2^{cI} \hat{v}_2^I + B_2^I \dot{p}_2^I),$$

$$2\hat{d}_{12} = B_2^I \hat{v}_1^I + B_1^I \hat{v}_2^I + \hat{x}_3 (B_1^{cI} \hat{v}_2^I + B_2^{cI} \hat{v}_1^I + B_1^I \dot{p}_2^I + B_2^I \dot{p}_1^I),$$

where p_α^I is the pseudonormal at node $\boldsymbol{I}$ and B_α^{cI} is given by

$$\begin{Bmatrix} B_1^{cI} \\ B_2^{cI} \end{Bmatrix} = \frac{2 z_\gamma}{A^2} \begin{bmatrix} \hat{x}_1 - \hat{x}_3 & \hat{x}_4 - \hat{x}_2 & \hat{x}_3 - \hat{x}_1 & \hat{x}_2 - \hat{x}_4 \\ \hat{y}_1 - \hat{y}_3 & \hat{y}_4 - \hat{y}_2 & \hat{y}_3 - \hat{y}_1 & \hat{y}_2 - \hat{y}_4 \end{bmatrix},$$

where

$$z_\gamma = \gamma^I \hat{x}_3^I,$$

$$\gamma^I = \frac{1}{4} [h^I - B_\alpha^I \hat{x}_\alpha^J h^J],$$

$$h^I = [+1, -1, +1, -1].$$

The pseudonormal p_α^I represents a nodal normal local to a particular element found by taking the vector cross product of the adjacent element sides.

The components of the transverse shear velocity strain are given by

$$\hat{\gamma}_{13} = N^I(\xi_\alpha) \bar{\theta}_2^I,$$

$$\hat{\gamma}_{23} = -N^I(\xi_\alpha) \bar{\theta}_1^I,$$

where nodal rotational components $\bar{\theta}_1^I$ and $\bar{\theta}_2^I$ are based on a projection and a transformation. Consider three adjacent local element nodes $\boldsymbol{K}$, $\boldsymbol{I}$, and $\boldsymbol{J}$ as shown in Figure 3.6.6–2. Outward facing vectors e_n^K and e_n^I are constructed perpendicular to element sides KI and IJ, respectively. In addition, they are tangent to the reference surface at the midsides.

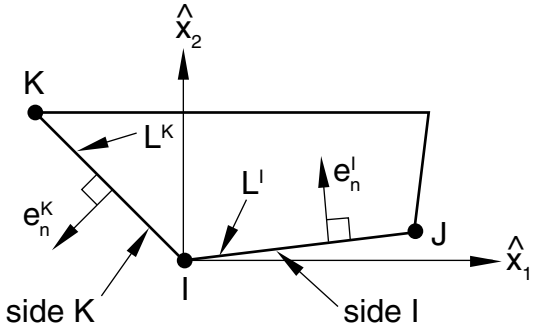


Figure 3.6.6–2 Vector and edge definition for shear projection in the element S4RSW.

The angular velocity θ_n^I about outward facing vector e_n^I is then given by a nodal projection

$$\theta_n^I = \frac{1}{2}(\theta_{nI}^I + \theta_{nI}^J) + \frac{1}{L^I}(\hat{v}_3^J - \hat{v}_3^I)\,, \text{no sum on } I,$$

where θ_{nI}^I is the rotational velocity at node ***I*** about e_n^I, θ_{nI}^J is the rotational velocity at node ***J*** about e_n^I, and L^I is the length of side ***I***. Finally, the nodal rotational components $\bar{\theta}_1^I$ and $\bar{\theta}_2^I$ required for the transverse shear velocity strain are given by the transformation

$$\bar{\theta}_1^I = (e_n^I \cdot \hat{e}_1)\theta_n^I + (e_n^K \cdot \hat{e}_1)\theta_n^K,$$

$$\bar{\theta}_2^I = (e_n^I \cdot \hat{e}_2)\theta_n^I + (e_n^K \cdot \hat{e}_2)\theta_n^K.$$

Evaluating the resulting forms for the transverse shear at the centroidal quadrature point gives

$$2\hat{d}_{13} = B_{11}^{sI}\hat{v}_3^I + B_{12}^{sI}\hat{\theta}_1^I + B_{13}^{sI}\hat{\theta}_2^I,$$

$$2\hat{d}_{13} = B_{21}^{sI}\hat{v}_3^I + B_{22}^{sI}\hat{\theta}_1^I + B_{23}^{sI}\hat{\theta}_2^I,$$

where

$$\left\{ \begin{array}{c} B_{1i}^{sI} \\ B_{2i}^{sI} \end{array} \right\} = \frac{1}{4} \left[\begin{array}{ccc} 2(\bar{x}_1^{JI} - \bar{x}_1^{IK}) & \hat{x}_1^{JI}\bar{x}_2^{JI} + \hat{x}_1^{IK}\bar{x}_2^{IK} & -(\hat{x}_1^{JI}\bar{x}_1^{JI} + \hat{x}_1^{IK}\bar{x}_1^{IK}) \\ 2(\bar{x}_2^{JI} - \bar{x}_2^{IK}) & \hat{x}_2^{JI}\bar{x}_2^{JI} + \hat{x}_2^{IK}\bar{x}_2^{IK} & -(\hat{x}_1^{JI}\bar{x}_2^{JI} + \hat{x}_1^{IK}\bar{x}_2^{IK}) \end{array} \right] \text{ no sum on } I,\ J, \text{ and } K,$$

$$\hat{x}_\alpha^{JI} = \hat{x}_\alpha^J - \hat{x}_\alpha^I$$

and

$$\bar{x}_\alpha^{JI} = \frac{\hat{x}_\alpha^{JI}}{\left(L^I\right)^2}\,, \text{ no sum on } I.$$

The local nodal forces and moments are then given in terms of the section resultant forces and moments by

$$\hat{f}_1^I = A\left(B_1^I \hat{f}_1 + B_2^I \hat{f}_{12} + B_1^{cI} \hat{m}_1 + B_2^{cI} \hat{m}_{12}\right),$$

$$\hat{f}_2^I = A\left(B_2^I \hat{f}_2 + B_1^I \hat{f}_{12} + B_2^{cI} \hat{m}_2 + B_1^{cI} \hat{m}_{12}\right),$$

$$\hat{f}_3^I = \kappa A\left(B_{11}^{sI} \hat{f}_{13} + B_{21}^{sI} \hat{f}_{23}\right),$$

$$\hat{m}_1^I = A\left(B_2^I \hat{m}_2 + B_1^I \hat{m}_{12} + \kappa(B_{12}^{sI} \hat{f}_{13} + B_{22}^{sI} \hat{f}_{23})\right),$$

$$\hat{m}_2^I = -A\left(B_1^I \hat{m}_1 + B_2^I \hat{m}_{12} + \kappa(B_{13}^{sI} \hat{f}_{13} + B_{23}^{sI} \hat{f}_{23})\right),$$

$$\hat{m}_3^I = 0.$$

Small-strain element S3RS

The triangular shell element formulation is similar to that of the S4RS element and is based on Kennedy et al. (1986). This element is not subject to zero energy modes inherent in quadrilateral element formulations.

The velocity strain is computed as in the S4RS element except that the gradient operator is given by

$$\begin{Bmatrix} B_1^I \\ B_2^I \end{Bmatrix} = \frac{1}{2A}\begin{bmatrix} -\hat{y}_3 & \hat{y}_3 & 0 \\ \hat{x}_3 - \hat{x}_2 & -\hat{x}_3 & \hat{x}_2 \end{bmatrix}.$$

The local nodal forces and moments for the triangular shell can be expressed in terms of section resultant forces and moments as

$$\hat{f}_1^I = A\left(B_1^I \hat{f}_1 + B_2^I \hat{f}_{12}\right),$$

$$\hat{f}_2^I = A\left(B_2^I \hat{f}_2 + B_1^I \hat{f}_{12}\right),$$

$$\hat{m}_1^I = A\left(B_2^I \hat{m}_2 + B_1^I \hat{m}_{12} - \frac{\kappa}{3}\hat{f}_{23}\right),$$

$$\hat{m}_2^I = -A\left(B_1^I \hat{m}_1 + B_2^I \hat{m}_{12} - \frac{\kappa}{3}\hat{f}_{13}\right),$$

$$\hat{m}_3^I = 0.$$

The $\hat{x}_3$-components of the nodal forces are obtained by successively solving the following equations:

$$\hat{m}_1^1 + \hat{m}_1^2 + \hat{m}_1^3 + \hat{x}_3\hat{f}_3^3 = 0,$$

$$\hat{m}_2^1 + \hat{m}_2^2 + \hat{m}_2^3 - \hat{x}_3\hat{f}_3^3 - \hat{x}_2\hat{f}_3^2 = 0,$$

$$\hat{f}_3^1 + \hat{f}_3^2 + \hat{f}_3^3 = 0,$$

which represent the equations of moment equilibrium about the $\hat{x}_1$-axis, moment equilibrium about the $\hat{x}_2$-axis, and force equilibrium in the $\hat{x}_3$-direction.

Hourglass control

Since the one-point quadrature is used, several spurious modes, often known as hourglass modes, are possible for the quadrilateral elements. To suppress the hourglass modes, a consistent spurious mode control as described by Belytschko et al. (1984) is used. The hourglass shape vector γ^I is defined as

$$\gamma^I = h^I - B_\alpha^I \hat{x}_\alpha^J h^J.$$

The hourglass strain rates are obtained by

$$\dot{q}_\alpha^B = \gamma^I \hat{\theta}_\alpha^I,$$

$$\dot{q}_3^B = \gamma^I \hat{v}_3^I,$$

$$\dot{q}_\alpha^M = \gamma^I \hat{v}_\alpha^I,$$

where the superscripts $\boldsymbol{B}$ and $\boldsymbol{M}$ denote hourglass modes associated with bending and in-plane (membrane) response, respectively. The corresponding generalized hourglass stresses for the element S4RS are given by

$$\dot{Q}_\alpha^B = \gamma_r s^r \frac{h^3 EA}{192} B_\beta^I B_\beta^I \dot{q}_\alpha^B,$$

$$\dot{Q}_3^B = \gamma_w s^w \frac{\kappa h^3 GA}{12} B_\beta^I B_\beta^I \dot{q}_3^B,$$

$$\dot{Q}_\alpha^M = \gamma_s s^s \frac{hEA}{8} B_\beta^I B_\beta^I \dot{q}_\alpha^M,$$

where h is the thickness of the shell and E and G are Young's modulus and shear modulus, respectively. The default hourglass control parameters are γ_s=γ_r=0.050 and γ_w=0.005. The scaling factors s^r, s^r, and s^w (by default s^r=s^r=s^w=1) are used to change the corresponding default hourglass control parameters by the user. For the S4RSW element the generalized hourglass stresses $\dot{Q}_3^B$ and $\dot{Q}_\alpha^M$ are the same as those in the element S4RS, but the generalized hourglass stress $\dot{Q}_\alpha^B$ is modified to

$$\dot{Q}_\alpha^B = \gamma_r s^r \frac{h^3 EA}{192} \left(1 + \frac{2\kappa A}{3h^2}\right) B_\beta^I B_\beta^I \dot{q}_\alpha^M.$$

The nodal hourglass forces and moments corresponding to the generalized hourglass stresses are

$$\hat{m}_\alpha^{hI} = \gamma^I Q_\alpha^B,$$

$$\hat{f}_3^{hI} = \gamma^I Q_3^B,$$

$$\hat{f}_\alpha^{hI} = \gamma^I Q_\alpha^M.$$

These hourglass forces and moments are added directly to the local nodal forces and moments described previously.

Reference

- "Choosing a shell element," Section 29.6.2 of the Abaqus Analysis User's Guide

3.6.7 AXISYMMETRIC SHELL ELEMENT ALLOWING ASYMMETRIC LOADING

Product: Abaqus/Standard

The Abaqus/Standard element library includes a family of nonlinear thin shell elements with axisymmetric reference geometry that allow asymmetric loading and deformation (SAXA1N and SAXA2N). This section provides their theoretical formulation. These elements encompass a broad range of practical applications from the bending/ovalization of variable diameter pipes to the bending of circular plates. The theoretical formulation of these elements is similar to the general finite-strain shell element described in "Finite-strain shell element formulation," Section 3.6.5. Furthermore, this formulation is the shell counterpart to the continuum axisymmetric bending elements described in "Axisymmetric elements allowing nonlinear bending," Section 3.2.9.

As with the continuum axisymmetric bending formulation, the restriction is made that a plane of symmetry exists in the r–z plane at $\theta = 0$. Hence in-plane bending of the model is permitted, while deformations such as torsion about the axis of symmetry are precluded. The symmetries of the undeformed configuration and of the deformation are exploited through the assumption of particular displacement and rotation interpolations around the circumference of the shell. Specifically, Fourier series expansions are used in the θ or circumferential direction that preserve the plane of symmetry.

Geometric description

Let (S, θ) be coordinate functions parametrizing the reference surface of the shell and let $S_3 \in [-h/2, h/2]$ be the coordinate function in the thickness direction, where h is the shell's initial thickness. (For a detailed account of the geometric description of the finite-strain shell formulation, see "Finite-strain shell element formulation," Section 3.6.5.) Then points in the reference or undeformed configuration are identified by the normal coordinates mapping

$$\mathbf{X}(S, \theta, S_3) = \overline{\mathbf{X}}(S, \theta) + S_3 \mathbf{T}_3(S, \theta)\,,$$

where $\mathbf{X}$ is the three-dimensional position of a material point, $\overline{\mathbf{X}}$ is the shell reference surface mapping, and $\mathbf{T}_3$ is the unit normal to the shell reference surface. The fact that $\mathbf{T}_3$ is a unit vector assumes that the reference configuration is (locally) of constant thickness. Owing to the axisymmetric reference configuration, $\overline{\mathbf{X}}$ can be given relative to a global Cartesian coordinate system as

$$\overline{\mathbf{X}}(S, \theta) = \left\{ \begin{array}{c} X^1(S, \theta) \\ X^2(S, \theta) \\ X^3(S, \theta) \end{array} \right\} = \left\{ \begin{array}{c} r(S) \cos\theta \\ r(S) \sin\theta \\ z(S) \end{array} \right\},$$

where $r(S)$ is the radius, $z(S)$ is the axial position, and (r, z, θ) are the cylindrical coordinates. (Note that the usual convention for cylindrical coordinates (r, θ, z) has been changed, which is consistent with the axisymmetric shell elements and the axisymmetric elements allowing nonlinear bending.) By definition the normal field to the shell reference surface is $\mathbf{T}_3 = \overline{\mathbf{X}}_{,S} \times \overline{\mathbf{X}}_{,\theta} / \| \overline{\mathbf{X}}_{,S} \times \overline{\mathbf{X}}_{,\theta} \|$, which by direct computation yields

$$\mathbf{T}_3 = \frac{1}{[(r')^2 + (z')^2]^{\frac{1}{2}}} \begin{Bmatrix} -z' \cos\theta \\ -z' \sin\theta \\ r' \end{Bmatrix} \quad \text{with} \quad \|\overline{\mathbf{X}}_{,S} \times \overline{\mathbf{X}}_{,\theta}\| = r[(r')^2 + (z')^2]^{\frac{1}{2}} ,$$

where $r' = \partial r/\partial S$ and $z' = \partial z/\partial S$. Relative to the cylindrical coordinate system, $\mathbf{T}_3 = [(r')^2 + (z')^2]^{-\frac{1}{2}}(-z'\mathbf{e}_r + r'\mathbf{e}_z)$.

The basic kinematic assumption is that for any deformed configuration, the position of a point in the body can be identified by

$$\mathbf{x}(S, \theta, S_3) = \overline{\mathbf{x}}(S, \theta) + S_3 \overline{f}_{33}(S, \theta)\mathbf{t}_3(S, \theta) , \tag{3.6.7–1}$$

where $\mathbf{x}$ is the deformed position of the material point, $\overline{\mathbf{x}}$ is the deformed shell reference surface mapping, $\mathbf{t}_3$ is the deformed unit director field, and $\overline{f}_{33}$ is the thickness change parameter. Of critical importance for any shell formulation is the treatment of the rotation field; that is, the treatment of the director field $\mathbf{t}_3$. The geometric description and the incremental update procedure for the director field are given in detail below.

Under the kinematic assumption above, the deformed configuration of the shell is completely determined by the reference surface mapping $\overline{\mathbf{x}}$, the deformed director field $\mathbf{t}_3$ and the thickness parameter $\overline{f}_{33}$.

We define the following displacement quantities. Since $\overline{\mathbf{x}}$ is an element of a (linear) vector space, we can define the reference surface displacement vector $\mathbf{u}$ by the difference between the deformed reference surface and the undeformed reference surface; i.e.,

$$\mathbf{u}(S, \theta) = \overline{\mathbf{x}}(S, \theta) - \overline{\mathbf{X}}(S, \theta) .$$

The director field, however, is a unit vector field that is not a member of a linear vector space. The orientation of the director field is defined in terms of a rotation vector $\boldsymbol{\phi}$ as

$$\mathbf{t}_3(S, \theta) = \exp[\hat{\boldsymbol{\phi}}(S, \theta)] \cdot \mathbf{T}_3(S, \theta) .$$

Here $\hat{\boldsymbol{\phi}}$ is the skew-symmetric matrix with axial vector $\boldsymbol{\phi}$, defined by the properties

$$\hat{\boldsymbol{\phi}} \cdot \boldsymbol{\phi} = \mathbf{0} \quad \text{and} \quad \hat{\boldsymbol{\phi}} \cdot \mathbf{v} = \boldsymbol{\phi} \times \mathbf{v} \text{ for all vectors } \mathbf{v}$$

and $\exp[\hat{\boldsymbol{\phi}}]$ is an orthogonal transformation given by the closed-form expression

$$\exp[\hat{\boldsymbol{\phi}}] = \cos\|\boldsymbol{\phi}\|\mathbf{I} + \frac{\sin\|\boldsymbol{\phi}\|}{\|\boldsymbol{\phi}\|}\hat{\boldsymbol{\phi}} + \frac{(1 - \cos\|\boldsymbol{\phi}\|)}{\|\boldsymbol{\phi}\|^2}\boldsymbol{\phi}\boldsymbol{\phi} .$$

Alternatively, quaternion algebra can be used to specify the orientation of the deformed director field $\mathbf{t}_3$. In this case the orthogonal matrix $\exp[\hat{\boldsymbol{\phi}}]$ is replaced by the quaternion parameter $\boldsymbol{q} = (\boldsymbol{q}_\mathbf{o}, \mathbf{q})$, where

$$q_0 = \cos\|\boldsymbol{\phi}/2\| \quad \text{and} \quad \mathbf{q} = \frac{\sin\|\boldsymbol{\phi}/2\|}{\|\boldsymbol{\phi}\|}\boldsymbol{\phi}\,.$$

The orientation of the unit director field then follows as

$$\mathbf{t}_3 = (2q_0^2 - 1)\mathbf{T}_3 + 2q_0\mathbf{q}\times\mathbf{T}_3 + 2(\mathbf{q}\cdot\mathbf{T}_3)\mathbf{q}\,.$$

Similarly, the orthogonal transformation $\exp[\hat{\boldsymbol{\phi}}]$ can be extracted from the quaternion parameter as

$$\exp[\hat{\boldsymbol{\phi}}] = (2q_0^2 - 1)\mathbf{I} + 2q_0\hat{\mathbf{q}} + 2\mathbf{q}\mathbf{q}\,.$$

Interpolations

Displacement and rotation components are given relative to the cylindrical coordinate system (r, z, θ) with orthonormal basis vectors $\{\mathbf{e}_r, \mathbf{e}_z, \mathbf{e}_\theta\}$ that are fixed in the reference or undeformed configuration. A general interpolation scheme for $\mathbf{u} = u_r\mathbf{e}_r + u_z\mathbf{e}_z + u_\theta\mathbf{e}_\theta$ and $\boldsymbol{\phi} = \phi_r\mathbf{e}_r + \phi_z\mathbf{e}_z + \phi_\theta\mathbf{e}_\theta$ using a Fourier expansion in the θ variable is

$$\left.\begin{aligned} u_i(S,\theta) &= \sum_{m=1}^{M} H^m(S)\left(u_i^{m0} + \sum_{p=1}^{P}(\cos(p\theta)u_{ic}^{mp} + \sin(p\theta)u_{is}^{mp})\right)\\ \phi_i(S,\theta) &= \sum_{m=1}^{M} H^m(S)\left(\phi_i^{m0} + \sum_{p=1}^{P}(\cos(p\theta)\phi_{ic}^{mp} + \sin(p\theta)\phi_{is}^{mp})\right)\end{aligned}\right\} \quad (i = r, z, \theta)\,.$$

Here $H^m(S)$ are the polynomial interpolation functions along the generator lines of the axisymmetric reference configuration; u_i^{m0}, u_{ic}^{mp}, u_{is}^{mp}, ϕ_i^{m0}, ϕ_{ic}^{mp}, ϕ_{is}^{mp} are the solution amplitude values (Fourier coefficients); M is the number of terms used in the interpolation along the generator lines; and P is the number of Fourier interpolation terms used around the circumference of the reference shell. Note that an axisymmetric deformation is obtained for the choice $P = 0$.

The symmetry requirement in the r–z plane at $\theta = 2n\pi$, $n = 0, 1, \ldots$, eliminates many of the above Fourier coefficients. For the displacement vector the only admissible terms are

$$\begin{Bmatrix} u_r(S,\theta)\\ u_z(S,\theta)\\ u_\theta(S,\theta)\end{Bmatrix} = \sum_{m=1}^{M} H^m(S)\left(\begin{Bmatrix} u_r^{m0}\\ u_z^{m0}\\ 0\end{Bmatrix} + \sum_{p=1}^{P}\cos(p\theta)\begin{Bmatrix} u_{rc}^{mp}\\ u_{zc}^{mp}\\ 0\end{Bmatrix} + \sum_{p=1}^{P}\sin(p\theta)\begin{Bmatrix} 0\\ 0\\ u_{\theta s}^{mp}\end{Bmatrix}\right)\,.$$

For the rotation components, symmetry requirements switch the role of the r and z components with the θ components:

$$\begin{Bmatrix} \phi_r(S,\theta)\\ \phi_z(S,\theta)\\ \phi_\theta(S,\theta)\end{Bmatrix} = \sum_{m=1}^{M} H^m(S)\left(\begin{Bmatrix} 0\\ 0\\ \phi_\theta^{m0}\end{Bmatrix} + \sum_{p=1}^{P}\cos(p\theta)\begin{Bmatrix} 0\\ 0\\ \phi_{\theta c}^{mp}\end{Bmatrix} + \sum_{p=1}^{P}\sin(p\theta)\begin{Bmatrix} \phi_{rs}^{mp}\\ \phi_{zs}^{mp}\\ 0\end{Bmatrix}\right)\,.$$

For practical reasons the values of u_r, u_z, and ϕ_θ are often required at specific locations around the circumference of the shell. Therefore, displacement and rotation components u_r^{mp}, u_z^{mp}, and ϕ_θ^{mp} are used instead of the Fourier coefficients u_{rc}^{mp}, u_{zc}^{mp}, and $\phi_{\theta c}^{mp}$. Furthermore, a negative sign is introduced in the interpolation for ϕ_θ for the following reason: The Abaqus convention for axisymmetric shell elements is that the axial tangent direction is drawn between nodes in ascending node number (the shell local 1-direction). The normal to the shell is then obtained by a 90° counter-clockwise rotation of the tangent (the shell local 3-direction). However, a positive rotation of the normal field (about the shell local 2-direction) is counterclockwise. This convention implies a left-handed shell local coordinate system. For the axisymmetric shell bending elements, a right-handed shell local coordinate system is required at the integration points; thus, the direction of positive rotation is reversed there.

Rearranging the Fourier series expansions and making the substitution $u_{\theta s}^{mp} \mapsto u_\theta^{mp}$, the interpolations for the displacement components are

$$\left\{ \begin{array}{c} u_r(S,\theta) \\ u_z(S,\theta) \\ u_\theta(S,\theta) \end{array} \right\} = \sum_{m=1}^{M} H^m(S) \left(\sum_{p=1}^{P+1} R^p(\theta) \left\{ \begin{array}{c} u_r^{mp} \\ u_z^{mp} \\ 0 \end{array} \right\} + \sum_{p=1}^{P} \sin(p\theta) \left\{ \begin{array}{c} 0 \\ 0 \\ u_\theta^{mp} \end{array} \right\} \right) . \qquad (3.6.7\text{–}2)$$

Similarly, replacing ϕ_{rs}^{mp} and ϕ_{zs}^{mp} with ϕ_r^{mp} and ϕ_z^{mp} respectively, the interpolation for the rotation components becomes

$$\left\{ \begin{array}{c} \phi_r(S,\theta) \\ \phi_z(S,\theta) \\ \phi_\theta(S,\theta) \end{array} \right\} = \sum_{m=1}^{M} H^m(S) \left(\sum_{p=1}^{P+1} R^p(\theta) \left\{ \begin{array}{c} 0 \\ 0 \\ -\phi_\theta^{mp} \end{array} \right\} + \sum_{p=1}^{P} \sin(p\theta) \left\{ \begin{array}{c} \phi_r^{mp} \\ \phi_z^{mp} \\ 0 \end{array} \right\} \right) . \qquad (3.6.7\text{–}3)$$

In the above interpolations, u_r^{mp}, u_z^{mp}, and ϕ_θ^{mp} are physical displacement and rotation components at $\theta^p = \pi(p-1)/P$ and $R^p(\theta)$ are trigonometric interpolation functions with the property that $R^p(\theta^q) = \left\{ \begin{array}{l} 1\,,\ p = q \\ 0\,,\ p \neq q \end{array} \right.$ defined by:

$P = 1$: $\theta^1 = 0$, $\theta^2 = \pi$,

$$R^1(\theta) = \frac{1}{2}(1 + \cos\theta)$$
$$R^2(\theta) = \frac{1}{2}(1 - \cos\theta)$$

$P = 2$: $\theta^1 = 0$, $\theta^2 = \pi/2$, $\theta^3 = \pi$,

$$R^1(\theta) = \frac{1}{4}(1 + 2\cos\theta + \cos 2\theta)$$
$$R^2(\theta) = \frac{1}{2}(1 - \cos 2\theta)$$
$$R^3(\theta) = \frac{1}{4}(1 - 2\cos\theta + \cos 2\theta)$$

$P = 3$: $\theta^1 = 0$, $\theta^2 = \pi/3$, $\theta^3 = 2\pi/3$, $\theta^4 = \pi$,

$$R^1(\theta) = \frac{1}{6}(1 + 2\cos\theta + 2\cos 2\theta + \cos 3\theta)$$
$$R^2(\theta) = \frac{1}{3}(1 + \cos\theta - \cos 2\theta - \cos 3\theta)$$
$$R^3(\theta) = \frac{1}{3}(1 - \cos\theta - \cos 2\theta + \cos 3\theta)$$
$$R^4(\theta) = \frac{1}{6}(1 - 2\cos\theta + 2\cos 2\theta - \cos 3\theta)$$

$P = 4$: $\theta^1 = 0$, $\theta^2 = \pi/4$, $\theta^3 = \pi/2$, $\theta^4 = 3\pi/4$, $\theta^5 = \pi$,

$$R^1(\theta) = \frac{1}{8}(1 + 2\cos\theta + 2\cos 2\theta + 2\cos 3\theta + \cos 4\theta)$$
$$R^2(\theta) = \frac{1}{4}(1 + \sqrt{2}\cos\theta - \sqrt{2}\cos 3\theta - \cos 4\theta)$$
$$R^3(\theta) = \frac{1}{4}(1 - 2\cos 2\theta + \cos 4\theta)$$
$$R^4(\theta) = \frac{1}{4}(1 - \sqrt{2}\cos\theta + \sqrt{2}\cos 3\theta - \cos 4\theta)$$
$$R^5(\theta) = \frac{1}{8}(1 - 2\cos\theta + 2\cos 2\theta - 2\cos 3\theta + \cos 4\theta)$$

As with the continuum axisymmetric bending element, $P = 4$ is the highest-order interpolation offered with respect to θ. The element becomes significantly more expensive as higher-order interpolations are used, and it is assumed that the general-purpose finite-strain shell is less expensive than using this element with $P > 4$.

Virtual work

The virtual work expression from the three-dimensional theory is

$$\delta\Pi = \int_V \sigma^{ij}\delta E_{ij}\, dV\,,$$

where V is the current volume of the deformed body, σ^{ij} are the curvilinear components of the Cauchy stress tensor, E_{ij} are the components of the Lagrange strain tensor, and δE_{ij} are the variational or linearized strain measure components. By definition the Lagrange strain tensor components are given by

$$E_{ij} \stackrel{\text{def}}{=} \tfrac{1}{2}\left(\frac{\partial \mathbf{x}}{\partial \xi^i}\cdot\frac{\partial \mathbf{x}}{\partial \xi^j} - \frac{\partial \mathbf{X}}{\partial \xi^i}\cdot\frac{\partial \mathbf{X}}{\partial \xi^j}\right).$$

Note that in the statement of virtual work, no choice has thus far been made regarding the curvilinear coordinate functions ξ^i $(i = 1, 2, 3)$. Furthermore, the current volume measure dV is given by the parametric relationship

$$dV = j d\xi^1 d\xi^2 d\xi^3\,, \quad \text{where} \quad j \stackrel{\text{def}}{=} \det[\nabla \mathbf{x}] = \frac{\partial \mathbf{x}}{\partial \xi^1} \times \frac{\partial \mathbf{x}}{\partial \xi^2} \cdot \frac{\partial \mathbf{x}}{\partial \xi^3}\,.$$

We now introduce the kinematic assumption Equation 3.6.7–1 into the definition of E_{ij} to find

$$\begin{aligned}
E_{\alpha\beta} &= \tfrac{1}{2}(\overline{\mathbf{x}}_{,\alpha} \cdot \overline{\mathbf{x}}_{,\beta} - \overline{\mathbf{X}}_{,\alpha} \cdot \overline{\mathbf{X}}_{,\beta}) + \tfrac{1}{2} S_3[\overline{\mathbf{x}}_{,\alpha} \cdot (\overline{f}_{33}\mathbf{t}_3)_{,\beta} + \overline{\mathbf{x}}_{,\beta} \cdot (\overline{f}_{33}\mathbf{t}_3)_{,\alpha} \\
&\quad - \overline{\mathbf{X}}_{,\alpha} \cdot \mathbf{T}_{3,\beta} - \overline{\mathbf{X}}_{,\beta} \cdot \mathbf{T}_{3,\alpha}] + \tfrac{1}{2}(S_3)^2[(\overline{f}_{33}\mathbf{t}_3)_{,\alpha} \cdot (\overline{f}_{33}\mathbf{t}_3)_{,\beta} - \mathbf{T}_{3,\alpha} \cdot \mathbf{T}_{3,\beta}]\,, \\
E_{3\alpha} &= \tfrac{1}{2}(\overline{f}_{33}\overline{\mathbf{x}}_{,\alpha} \cdot \mathbf{t}_3 + S_3 \overline{f}_{33}\overline{f}_{33,\alpha})\,, \\
E_{33} &= \tfrac{1}{2}[(\overline{f}_{33})^2 - 1]\,,
\end{aligned}$$

where differentiation is now with respect to the parametric coordinates, so that $\alpha, \beta \in \{S, \theta\}$ and $3 = S_3$. Define the following shell strain measure components and kinematic relationships:

$$\begin{aligned}
\epsilon_{\alpha\beta} &\stackrel{\text{def}}{=} \tfrac{1}{2}(\overline{\mathbf{x}}_{,\alpha} \cdot \overline{\mathbf{x}}_{,\beta} - \overline{\mathbf{X}}_{,\alpha} \cdot \overline{\mathbf{X}}_{,\beta}) && \text{membrane strain,} \\
\kappa_{\alpha\beta} &\stackrel{\text{def}}{=} \tfrac{1}{2}(\overline{\mathbf{x}}_{,\alpha} \cdot \mathbf{t}_{3,\beta} + \overline{\mathbf{x}}_{,\beta} \cdot \mathbf{t}_{3,\alpha}) - b^0_{\alpha\beta} && \text{bending strain,} \\
\gamma_\alpha &\stackrel{\text{def}}{=} \overline{\mathbf{x}}_{,\alpha} \cdot \mathbf{t}_3 && \text{transverse shear,} \\
\mu &\stackrel{\text{def}}{=} \ln(\overline{f}_{33}) && \text{logarithmic stretch.}
\end{aligned}$$

In the above, $b^0_{\alpha\beta} \stackrel{\text{def}}{=} \overline{\mathbf{X}}_{,\alpha} \cdot \mathbf{T}_{3,\beta} = b^0_{\beta\alpha}$ are the components of the second fundamental form of the undeformed reference surface.

Substituting the above definitions into the virtual work expression, we find (after some manipulation) that the volume integral reduces to the following integral over the deformed reference surface

$$\delta\Pi = \int_A \left[N^{\alpha\beta}\delta\epsilon_{\alpha\beta} + M^{\alpha\beta}\delta\kappa_{\alpha\beta} + Q^\alpha \delta\gamma_\alpha + L^3\delta\mu + M^{3\alpha}(\delta\mu)_{,\alpha}\right] \overline{j}\, dS d\theta\,,$$

where $\delta\mu = \delta\overline{f}_{33}/\overline{f}_{33}$ and the current reference surface Jacobian determinant is $\overline{j} = \|\overline{\mathbf{x}}_{,1} \times \overline{\mathbf{x}}_{,2}\|$. In the above virtual work expression, the $(S_3)^2$ term in $E_{\alpha\beta}$ has been neglected. This term is $O(h^2/R^2)$—where $\boldsymbol{h}$ is the thickness and $\boldsymbol{R}$ is some characteristic radius of curvature—and is negligible in light of the kinematic assumption Equation 3.6.7–1. The shell stress resultant components are defined by the following integrals through the thickness of the shell:

$$
\begin{aligned}
N^{\alpha\beta} &= \frac{1}{\overline{\overline{j}}} \int_h \sigma^{\alpha\beta}\, j dS_3\,, \\
M^{\alpha\beta} &= \overline{f}_{33} \frac{1}{\overline{\overline{j}}} \int_h S_3 \sigma^{\alpha\beta}\, j dS_3\,, \\
Q^{\alpha} &= \overline{f}_{33} \frac{1}{\overline{\overline{j}}} \int_h \left(\sigma^{3\alpha} + S_3 \sigma^{\alpha\beta}\mu_{,\beta}\right) j dS_3\,, \\
L^{3} &= \overline{f}_{33} \frac{1}{\overline{\overline{j}}} \int_h \left[\sigma^{33}\overline{f}_{33} + \sigma^{3\alpha}\gamma_\alpha + 2S_3\sigma^{3\alpha}\overline{f}_{33,\alpha} + S_3\sigma^{\alpha\beta}(\kappa_{\alpha\beta} + \gamma_\alpha\mu_{,\beta})\right] j dS_3\,, \\
M^{3\alpha} &= \overline{f}_{33} \frac{1}{\overline{\overline{j}}} \int_h S_3\left(\sigma^{3\alpha}\overline{f}_{33} + \sigma^{\alpha\beta}\gamma_\beta\right) j dS_3\,.
\end{aligned}
$$

For thin shells the Kirchhoff-Love approximation, which states that the deformed director field $\mathbf{t}_3$ is (approximately) the normal field to the deformed reference surface, is introduced along with the plane stress assumption $\sigma^{33} = 0$. Consistent with these approximations, we neglect all $O(h/R)$ terms and terms proportional to the gradient of the thickness parameter. Accordingly, we set

$$
\gamma_\alpha = 0\,, \quad \frac{j}{\overline{\overline{j}}} = \overline{f}_{33} \quad \text{and} \quad \overline{f}_{33,\alpha} = \mu_{,\alpha} = 0\,.
$$

We can now summarize the virtual work expression for thin (Kirchhoff-Love) shells:

$$
\delta\Pi = \int_A \left[N^{\alpha\beta}\delta\epsilon_{\alpha\beta} + M^{\alpha\beta}\delta\kappa_{\alpha\beta} + Q^{\alpha}\delta\gamma_\alpha\right] \overline{j} dS d\theta\,, \tag{3.6.7–4}
$$

where the shell resultant components are defined in terms of the Cauchy stress tensor components by the integrals

$$
N^{\alpha\beta} = \overline{f}_{33} \int_h \sigma^{\alpha\beta}\, dS_3 \quad \text{and} \quad M^{\alpha\beta} = (\overline{f}_{33})^2 \int_h S_3 \sigma^{\alpha\beta}\, dS_3\,.
$$

In the expression for $\delta\Pi$, $Q^\alpha = (\overline{f}_{33})^2 \int_h \sigma^{3\alpha}\, dS_3$ is interpreted as a constraint stress that enforces that the director field remain normal to the reference surface. Two other contributions to the virtual work expression $M^{\alpha\beta}\kappa_{\alpha\beta}\delta\mu$ and $M^{3\alpha}(\delta\mu)_{,\alpha}$, where $M^{3\alpha} = (\overline{f}_{33})^2 \int_h S_3\sigma^{3\alpha}\, dS_3$ are $O(h/R)$ and, thus, neglected.

Orthonormal surface coordinate system and coordinate transformation

It is desirable to define stress resultant quantities relative to an orthonormal basis in the deformed configuration. To do this, we define a normal coordinate system $\{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3\}$, where $\mathbf{n}_1$ and $\mathbf{n}_2$ are tangent to the deformed reference surface and $\mathbf{n}_3$ is the unit normal field.

Define the following notation. Let $\overline{f}^a_\alpha$ $(a = 1, 2)$ be the components of $\overline{\mathbf{x}}_{,\alpha}$ relative to the basis $\{\mathbf{n}_1, \mathbf{n}_2\}$; that is

$$\frac{\partial \overline{\mathbf{x}}}{\partial \xi^\alpha} = \overline{f}^a_\alpha \mathbf{n}_a \,. \tag{3.6.7–5}$$

Furthermore, let the inverse of the matrix of components $[\overline{f}^a_\alpha]$ be given by $[\overline{h}^\alpha_a]$, such that

$$\overline{f}^a_\alpha \overline{h}^\alpha_b = \delta^a_b \quad \text{and} \quad \overline{h}^\alpha_a \overline{f}^a_\beta = \delta^\alpha_\beta \,.$$

Note that the basis vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ induce distance measuring coordinates s^1 and s^2 such that

$$\overline{f}^a_\alpha = \frac{\partial s^a}{\partial \xi^\alpha} \quad \text{and} \quad \overline{h}^\alpha_a = \frac{\partial \xi^\alpha}{\partial s^a} \,.$$

It follows from Equation 3.6.7–5 that the orthonormal tangent vectors are given by

$$\mathbf{n}_a = \overline{h}^\alpha_a \overline{\mathbf{x}}_{,\alpha} = \frac{\partial \overline{\mathbf{x}}}{\partial s^a} \,.$$

For the material calculations, it is important to express both the strain and stress quantities relative to the local orthonormal frame $\{\mathbf{n}_1, \mathbf{n}_2\}$. Accordingly, let $N^{ab} = N^{ba}$ and $M^{ab} = M^{ba}$ be the membrane and bending stress resultant components relative to this local orthonormal basis. Thus, we can write

$$N^{\alpha\beta} = N^{ab} \overline{h}^\alpha_a \overline{h}^\beta_b \quad \text{and} \quad M^{\alpha\beta} = M^{ab} \overline{h}^\alpha_a \overline{h}^\beta_b \,,$$

where we recall that $\overline{h}^\alpha_a \overset{\text{def}}{=} \partial \xi^\alpha / \partial s^a$. Then the stress resultant contributions to the virtual work expression can be transformed as follows. First, the membrane contribution:

$$\begin{aligned} N^{\alpha\beta} \delta \epsilon_{\alpha\beta} &= N^{\alpha\beta} \overline{\mathbf{x}}_{,\alpha} \cdot (\delta \overline{\mathbf{x}})_{,\beta} \\ &= N^{ab} \overline{h}^\alpha_a \overline{h}^\beta_b \overline{\mathbf{x}}_{,\alpha} \cdot (\delta \overline{\mathbf{x}})_{,\beta} \\ &= N^{ab} \frac{\partial \overline{\mathbf{x}}}{\partial s^a} \cdot \frac{\partial (\delta \overline{\mathbf{x}})}{\partial s^b} \,. \end{aligned}$$

Recall, however, that by the definition of the coordinates s^a, $\partial \overline{\mathbf{x}} / \partial s^a = \mathbf{n}_a$. Thus,

$$N^{\alpha\beta} \delta \epsilon_{\alpha\beta} = N^{ab} \text{sym} \big[\mathbf{n}_a \cdot \frac{\partial (\delta \overline{\mathbf{x}})}{\partial s^b} \big] \,.$$

Similarly, the bending contribution is:

$$\begin{aligned} M^{\alpha\beta} \delta \kappa_{\alpha\beta} &= M^{\alpha\beta} \big[\overline{\mathbf{x}}_{,\alpha} \cdot (\delta \mathbf{t}_3)_{,\beta} + \mathbf{t}_3 \cdot (\delta \overline{\mathbf{x}})_{,\alpha} \big] \\ &= M^{ab} \big[\mathbf{n}_a \cdot \frac{\partial (\delta \boldsymbol{\phi} \times \mathbf{t}_3)}{\partial s^b} + \frac{\partial \mathbf{t}_3}{\partial s^b} \cdot \frac{\partial (\delta \overline{\mathbf{x}})}{\partial s^a} \big] \,. \end{aligned}$$

Let e^c_a be the two-dimensional alternator, such that $e^1_1 = e^2_2 = 0$ and $e^2_1 = -e^1_2 = 1$. Then $\mathbf{t}_3 \times \mathbf{n}_a = e^c_a \mathbf{n}_c$ and

$$M^{\alpha\beta}\delta\kappa_{\alpha\beta} = M^{ab}\left[e_a^c\mathbf{n}_c\cdot\frac{\partial(\delta\boldsymbol{\phi})}{\partial s^b}+\frac{\partial\mathbf{t}_3}{\partial s^b}\cdot\left(\frac{\partial(\delta\overline{\mathbf{x}})}{\partial s^a}-\delta\boldsymbol{\phi}\times\mathbf{n}_a\right)\right].$$

Since the second term in the brackets is proportional to the bending curvature, we neglect this term relative to the first, yielding

$$M^{\alpha\beta}\delta\kappa_{\alpha\beta} = M^{ab}\mathrm{sym}\left[e_a^c\mathbf{n}_c\cdot\frac{\partial(\delta\boldsymbol{\phi})}{\partial s^b}\right].$$

Strain displacement operators

We now write the virtual work expression Equation 3.6.7–4 in matrix operator notation. Define the following stress resultant component vectors:

$$\mathbf{N}\overset{\text{def}}{=}\left\{\begin{array}{c}N^{11}\\N^{22}\\N^{12}\end{array}\right\}\quad\text{and}\quad\mathbf{M}\overset{\text{def}}{=}\left\{\begin{array}{c}M^{11}\\M^{22}\\M^{12}\end{array}\right\}.$$

Define the matrix strain displacement operators as follows:

$$\mathbf{B}_N=\left[\begin{array}{c|c}\mathbf{n}_1^T\frac{\partial}{\partial s^1}&\mathbf{0}_{1\times3}\\\mathbf{n}_2^T\frac{\partial}{\partial s^2}&\mathbf{0}_{1\times3}\\\mathbf{n}_1^T\frac{\partial}{\partial s^2}+\mathbf{n}_2^T\frac{\partial}{\partial s^1}&\mathbf{0}_{1\times3}\end{array}\right]\quad\text{and}\quad\mathbf{B}_M=\left[\begin{array}{c|c}\mathbf{0}_{1\times3}&\mathbf{n}_2^T\frac{\partial}{\partial s^1}\\\mathbf{0}_{1\times3}&-\mathbf{n}_1^T\frac{\partial}{\partial s^2}\\\mathbf{0}_{1\times3}&\mathbf{n}_2^T\frac{\partial}{\partial s^2}-\mathbf{n}_1^T\frac{\partial}{\partial s^1}\end{array}\right],$$

where $\mathbf{0}_{1\times3}$ is the column of zeros $\{\,0\quad0\quad0\,\}^T$. Then the virtual work expression Equation 3.6.7–4 is equivalently stated

$$\delta\Pi=\int_A\left[\mathbf{N}\cdot\mathbf{B}_N\left\{\begin{array}{c}\delta\overline{\mathbf{x}}\\\delta\boldsymbol{\phi}\end{array}\right\}+\mathbf{M}\cdot\mathbf{B}_M\left\{\begin{array}{c}\delta\overline{\mathbf{x}}\\\delta\boldsymbol{\phi}\end{array}\right\}\right]\overline{j}dSd\theta\,.$$

Corotational coordinate system

Thus far $\mathbf{n}_1$ and $\mathbf{n}_2$ are any two orthonormal vectors in the tangent plane to the reference surface. We can uniquely choose these two vectors by requiring the matrix components of the incremental reference surface deformation gradient $[\Delta\overline{f}{}^a_b]$, where

$$\Delta\overline{f}{}^a_b\overset{\text{def}}{=}\frac{\partial s^{(t+\Delta t)a}}{\partial s^{(t)b}}=\overline{f}{}^{(t+\Delta t)a}_{\beta}\overline{h}{}^{(t)\beta}_b\,,$$

to be symmetric; i.e., $\Delta\overline{f}{}^1_2=\Delta\overline{f}{}^2_1$. Note that by definition $\overline{f}{}^{(t+\Delta t)a}_{\alpha}=\Delta\overline{f}{}^a_b\overline{f}{}^{(t)b}_{\alpha}$. This symmetry condition defines a corotational orthonormal basis in the deformed configuration. This orthonormal basis is calculated as follows.

Obtain the Abaqus convention pair of orthonormal surface vectors $\hat{\mathbf{t}}_a$. We then rotate these vectors in the tangent plane to the reference surface about the normal vector $\mathbf{n}_3^{t+\Delta t}$ by the angle $\Delta\psi$, where $\Delta\psi$ is determined by the symmetry condition $\Delta\overline{f}_2^1 = \Delta\overline{f}_1^2$. Thus, we define

$$\begin{aligned} \mathbf{n}_1^{t+\Delta t} &= \quad \cos(\Delta\psi)\hat{\mathbf{t}}_1 + \sin(\Delta\psi)\hat{\mathbf{t}}_2\,, \\ \mathbf{n}_2^{t+\Delta t} &= -\sin(\Delta\psi)\hat{\mathbf{t}}_1 + \cos(\Delta\psi)\hat{\mathbf{t}}_2\,. \end{aligned} \tag{3.6.7–6}$$

It follows by definition that

$$\Delta\overline{f}_b^a = \mathbf{n}_a^{(t+\Delta t)} \cdot \frac{\partial\overline{\mathbf{x}}^{(t+\Delta t)}}{\partial s^{(t)b}}\,.$$

Therefore, define the quantity

$$\Delta\hat{f}_{ab} \stackrel{\text{def}}{=} \hat{\mathbf{t}}_a \cdot \frac{\partial\overline{\mathbf{x}}^{(t+\Delta t)}}{\partial s^{(t)b}}\,.$$

The symmetry condition requires that

$$(\cos(\Delta\psi)\hat{\mathbf{t}}_1 + \sin(\Delta\psi)\hat{\mathbf{t}}_2) \cdot \frac{\partial\overline{\mathbf{x}}^{t+\Delta t}}{\partial s^{(t)2}} = (-\sin(\Delta\psi)\hat{\mathbf{t}}_1 + \cos(\Delta\psi)\hat{\mathbf{t}}_2) \cdot \frac{\partial\overline{\mathbf{x}}^{t+\Delta t}}{\partial s^{(t)1}}\,.$$

From this it can be determined that

$$\Delta\psi = \tan^{-1}\left[\frac{\Delta\hat{f}_{21} - \Delta\hat{f}_{12}}{\Delta\hat{f}_{11} + \Delta\hat{f}_{22}}\right]\,,$$

and $\{\mathbf{n}_1^{t+\Delta t}, \mathbf{n}_2^{t+\Delta t}\}$ are then determined by Equation 3.6.7–6.

Having determined the updated vectors $\mathbf{n}_a^{t+\Delta t}$, we can calculate the quantities required for coordinate transformation. First the incremental deformation gradient components and the Jacobian matrix components:

$$\Delta\overline{f}_b^a \equiv \Delta\overline{f}_{ab} = \mathbf{n}_a^{(t+\Delta t)} \cdot \frac{\partial\overline{\mathbf{x}}^{(t+\Delta t)}}{\partial s^{(t)b}} \quad \text{and} \quad \overline{f}_\alpha^{(t+\Delta t)a} = \Delta\overline{f}_b^a \overline{f}_\alpha^{(t)b}\,.$$

Note that due to the symmetry of the incremental deformation gradient components, there is no ambiguity in writing $\Delta\overline{f}_b^a = \Delta\overline{f}_{ab}$. We can now calculate the inverse components

$$\overline{h}_a^{(t+\Delta t)\alpha} = \left[\overline{f}_\alpha^{(t+\Delta t)a}\right]^{-1} = \frac{\partial\xi^\alpha}{\partial s^{(t+\Delta t)a}}\,.$$

Consistent linearization

The iterative solution procedure requires the calculation of the consistent tangent stiffness. This stiffness has two parts, one resulting from the material model and the other resulting from the changing geometry.

We denote the second variational quantities with d. Returning to Equation 3.6.7–4, the virtual work expression can be written as

$$\delta\Pi = \int_V J\sigma^{ab}\delta E_{ab}\, dV^0 = \int_A \int_h J\sigma^{ab}\delta E_{ab}\, j^0 dS_3 dS d\theta\,,$$

where $J = j/j^0$, j^0 is the Jacobian determinant of the reference configuration parametrization given by $j^0 = \det[\nabla \mathbf{X}]$, and components are relative to the corotated frame with $a, b \in \{1, 2\}$. We assume constitutive behavior such that

$$\mathrm{d}(J\sigma^{ab}) = JC^{abcd}\mathrm{d}E_{cd}\,.$$

Thus, the variation of the virtual work expression yields

$$\begin{aligned}\mathrm{d}(\delta\Pi) = \int_A \Big[\int_h \frac{j}{\overline{j}} \big(\delta\epsilon_{ab} + \overline{f}_{33} S_3 \delta\kappa_{ab}\big) C^{abcd} \big(\mathrm{d}\epsilon_{cd} + \overline{f}_{33} S_3 \mathrm{d}\kappa_{cd}\big)\, dS_3 \\ + N^{ab}\mathrm{d}(\delta\epsilon_{ab}) + M^{ab}\mathrm{d}(\delta\kappa_{ab})\Big]\, \overline{j} dS d\theta\,.\end{aligned}$$

The first term in the integrand forms the material stiffness, while the second two terms form the geometric stiffness. The second variation of the strain measure components are calculated in "Finite-strain shell element formulation," Section 3.6.5. The second variation of the membrane strain is

$$\mathrm{d}(\delta\epsilon_{ab}) = \mathrm{sym}\Big[\frac{\partial(\delta\overline{\mathbf{x}})}{\partial s^a} \cdot \frac{\partial(\mathrm{d}\overline{\mathbf{x}})}{\partial s^b} - 2\delta\epsilon_{ac}\mathrm{d}\epsilon_{cb}\Big]\,.$$

The second variation of the bending strain is zero; i.e.,

$$\mathrm{d}(\delta\kappa_{ab}) = 0\,.$$

Incremental degrees of freedom: interpolations and configuration updates

At the beginning of an increment we have the configuration at iteration k, denoted $(\overline{\mathbf{x}}^k, \mathbf{t}_3^k)$. In the incremental solution procedure we solve for the incremental displacement field and the incremental rotation field in components relative to the reference cylindrical coordinate system:

$$(\Delta u_r^{mp}, \Delta u_z^{mp}, \Delta u_\theta^{mp}, \Delta\phi_r^{mp}, \Delta\phi_z^{mp}, \Delta\phi_\theta^{mp})\,.$$

We use these incremental fields to update the configuration to iteration $k + 1$.

1. *Reference surface update*: The displacement increments are interpolated using the same interpolation scheme as the total displacement vector in Equation 3.6.7–2:

$$\Delta\mathbf{u}(S,\theta) = \left\{\begin{array}{c}\Delta u_r(S,\theta)\\ \Delta u_z(S,\theta)\\ \Delta u_\theta(S,\theta)\end{array}\right\} = \sum_{m=1}^{M} H^m(S) \left(\sum_{p=1}^{P+1} R^p(\theta) \left\{\begin{array}{c}\Delta u_r^{mp}\\ \Delta u_z^{mp}\\ 0\end{array}\right\} + \sum_{p=1}^{P} \sin(p\theta) \left\{\begin{array}{c}0\\ 0\\ \Delta u_\theta^{mp}\end{array}\right\} \right)\,.$$

The reference surface position map is updated from the displacement increment by

$$\overline{\mathbf{x}}^{k+1}(S,\theta) = \overline{\mathbf{x}}^{k}(S,\theta) + \Delta\mathbf{u}(S,\theta)\,.$$

2. *Rotation field update*: The incremental rotation field is updated with the same interpolation scheme as the total rotation field in Equation 3.6.7–3:

$$\Delta\boldsymbol{\phi}(S,\theta) = \left\{\begin{array}{c}\Delta\phi_r(S,\theta)\\ \Delta\phi_z(S,\theta)\\ \Delta\phi_\theta(S,\theta)\end{array}\right\} = \sum_{m=1}^{M} H^m(S)\left(\sum_{p=1}^{P+1} R^p(\theta)\left\{\begin{array}{c}0\\0\\-\Delta\phi_\theta^{mp}\end{array}\right\} + \sum_{p=1}^{P}\sin(p\theta)\left\{\begin{array}{c}\Delta\phi_r^{mp}\\ \Delta\phi_z^{mp}\\ 0\end{array}\right\}\right)\,.$$

This incremental rotation vector corresponds to a finite rotation, characterized by quaternion parameters as

$$\Delta q_0 = \cos\|\Delta\boldsymbol{\phi}/2\| \quad \text{and} \quad \Delta\mathbf{q} = \frac{\sin\|\Delta\boldsymbol{\phi}/2\|}{\|\Delta\boldsymbol{\phi}\|}\Delta\boldsymbol{\phi}\,.$$

The total rotation quaternion parameter can be updated by the update formula

$$\boldsymbol{q}^{k+1} = (\boldsymbol{q}_o^{k+1}, \mathbf{q}^{k+1}) = (\boldsymbol{\Delta q}_o\boldsymbol{q}_o^k - \boldsymbol{\Delta q}\cdot\mathbf{q}^k, \boldsymbol{\Delta q}_o\mathbf{q}^k + \boldsymbol{q}_o^k\boldsymbol{\Delta q} + \boldsymbol{\Delta q}\times\mathbf{q}^k)\,.$$

Similarly, the deformed unit director field can be updated from the incremental rotation field as

$$\mathbf{t}_3^{k+1} = (2(\Delta q_0)^2 - 1)\mathbf{t}_3^k + 2\Delta q_0\Delta\mathbf{q}\times\mathbf{t}_3^k + 2(\Delta\mathbf{q}\cdot\mathbf{t}_3^k)\Delta\mathbf{q} \stackrel{\text{def}}{=} (\Delta q_0, \Delta\mathbf{q})\circ\mathbf{t}_3^k\,.$$

Here we have used the notation $(q_0,\mathbf{q})\circ\mathbf{v}$ to denote the rotation of the vector $\mathbf{v}$ by the quaternion $(q_0,\mathbf{q})$. The curvatures are calculated from the gradient of the director field, which is updated by

$$\mathbf{t}_{3,\alpha}^{k+1} = (\Delta q_0, \Delta\mathbf{q})\circ\mathbf{t}_{3,\alpha}^k + \Delta\mathbf{R}_\alpha\times\mathbf{t}_3^{k+1}\,,$$

where

$$\Delta\mathbf{R}_\alpha = \mathbf{H}(\Delta\boldsymbol{\phi})\cdot\Delta\boldsymbol{\phi}_{,\alpha}\,,$$
$$\mathbf{H}(\Delta\boldsymbol{\phi}) = \frac{\sin\|\Delta\boldsymbol{\phi}\|}{\|\Delta\boldsymbol{\phi}\|}\mathbf{I} + \frac{1}{\|\Delta\boldsymbol{\phi}\|^2}\left(1 - \frac{\sin\|\Delta\boldsymbol{\phi}\|}{\|\Delta\boldsymbol{\phi}\|}\right)\Delta\boldsymbol{\phi}\Delta\boldsymbol{\phi} + \frac{(1-\cos\|\Delta\boldsymbol{\phi}\|)}{\|\Delta\boldsymbol{\phi}\|^2}\widehat{\Delta\boldsymbol{\phi}}\,.$$

For completeness, we record the values of $\Delta\boldsymbol{\phi}_{,\alpha}$. First, along the generator lines we have

$$\Delta\boldsymbol{\phi}_{,S} = \sum_{m=1}^{M} H^m_{,S}(S)\left(\sum_{p=1}^{P+1} R^p(\theta)\left\{\begin{array}{c}0\\0\\\Delta\phi_\theta^{mp}\end{array}\right\} + \sum_{p=1}^{P}\sin(p\theta)\left\{\begin{array}{c}-\Delta\phi_r^{mp}\\ \Delta\phi_z^{mp}\\ 0\end{array}\right\}\right)\,.$$

For the circumferential derivative we must account for the derivative of the basis vectors in the θ-direction: $\mathbf{e}_{r,\theta} = \mathbf{e}_\theta$ and $\mathbf{e}_{\theta,\theta} = -\mathbf{e}_r$. Hence, $\Delta\boldsymbol{\phi}_{,\theta} = (\Delta\phi_{r,\theta} - \Delta\phi_\theta)\mathbf{e}_r + \Delta\phi_{z,\theta}\mathbf{e}_z + (\Delta\phi_{\theta,\theta} + \Delta\phi_r)\mathbf{e}_\theta$. Introducing the interpolation function, we have

$$\Delta\boldsymbol{\phi}_{,\theta} = \left\{ \begin{array}{c} -\Delta\phi_\theta \\ 0 \\ \Delta\phi_r \end{array} \right\} + \sum_{m=1}^{M} H^m(S) \left(\sum_{p=1}^{P+1} R^p_{,\theta}(\theta) \left\{ \begin{array}{c} 0 \\ 0 \\ -\Delta\phi_\theta^{mp} \end{array} \right\} + \sum_{p=1}^{P} p\cos(p\theta) \left\{ \begin{array}{c} \Delta\phi_r^{mp} \\ \Delta\phi_z^{mp} \\ 0 \end{array} \right\} \right) ,$$

where $R^p_{,\theta}(\theta)$ are computed from the definitions of the interpolation functions $R^p(\theta)$ as given above and $\Delta\phi_\theta$ and $\Delta\phi_r$ are given by the interpolation for the incremental rotation field as given in the Rotation Field Update.

Strain increment and stress resultant update

Following the formulation of the finite-strain shell element formulation in "Finite-strain shell element formulation," Section 3.6.5, the three-dimensional (finite) strain increment is calculated as

$$\Delta E_{ab} = \Delta\epsilon_{ab} + S_3 \overline{f}^{t+\Delta t}_{33} \Delta\kappa_{ab} \, .$$

Here the increment in the membrane strain components $\Delta\epsilon_{ab}$ is given by the Hughes-Winget second-order approximation to the logarithmic membrane strain increment

$$\Delta\epsilon_{ab} = 2(\Delta\overline{f}_{ac} - \delta_{ac})(\Delta\overline{f}_{cb} + \delta_{cb})^{-1} \, ,$$

and the increment in the bending strain components $\Delta\kappa_{ab}$ is given by the expression

$$\Delta\kappa_{ab} = \mathrm{sym}\left[e^c_a \mathbf{n}^{t+\Delta t}_c \cdot \Delta\mathbf{r}_b \right] ,$$

where

$$\Delta\mathbf{r}_b = \overline{h}^\alpha_b \, \Delta\mathbf{R}_\alpha \, .$$

As an example of the stress update procedure, consider the simple case of a Saint Venant-Kirchhoff material model. In this case,

$$\sigma^{(t+\Delta t)ab} = \Delta\overline{f}^a_c \sigma^{(t)cd} \Delta\overline{f}^b_d + \Delta\overline{f}^a_e \Delta\overline{f}^b_f C^{(t)efgh} \Delta\overline{f}^c_g \Delta\overline{f}^d_h \Delta E_{cd} \, ,$$

where $C^{(0)efgh}$ are the plane stress elastic coefficients given by

$$C^{(0)efgh} = \frac{E}{(1-\nu^2)} \left\{ \nu\delta^{ef}\delta^{gh} + \frac{(1-\nu)}{2}\left(\delta^{eg}\delta^{fh} + \delta^{eh}\delta^{fg} \right) \right\} .$$

Note that $a, b, c, d, e, f, h \in \{1, 2\}$, and the components $\sigma^{(t+\Delta t)ab}$ are relative to the current orthonormal basis $\{\mathbf{n}^{t+\Delta t}_1, \mathbf{n}^{t+\Delta t}_2\}$.

Pressure loads and load stiffness

For geometrically linear problems, equivalent nodal loads due to applied surface pressure are readily calculated since the geometry is axisymmetric. For geometrically nonlinear problems, however, asymmetric deformations must be taken into consideration.

The equivalent nodal loads associated with surface pressure p can be obtained by considering the virtual work contribution to the external loading

$$-\int_A p\mathbf{n}\cdot\delta\overline{\mathbf{x}}dA=\int_0^{2\pi}\int_{-1}^{+1}p\left(\frac{\partial\overline{\mathbf{x}}}{\partial\theta}\times\frac{\partial\overline{\mathbf{x}}}{\partial S}\right)\cdot\delta\overline{\mathbf{x}}\,dSd\theta,$$

where S is the parametric surface coordinate in the R–Z plane and the reference surface position is

$$\overline{\mathbf{x}}=r\mathbf{e}_r+z\mathbf{e}_z+u_\theta\mathbf{e}_\theta,$$

with $r=R+u_r$ and $z=Z+u_z$. Recall that the current position of a point can be expressed in terms of the axial interpolator $H^m(S)$ and the circumferential interpolators $R^p(\theta)$ by

$$\left\{\begin{array}{c}r\\z\\u_\theta\end{array}\right\}=\sum_{m=1}^{M}H^m(S)\left(\sum_{p=1}^{P+1}R^p(\theta)\left\{\begin{array}{c}r^{mp}\\z^{mp}\\0\end{array}\right\}+\sum_{p=1}^{P}\sin p\theta\left\{\begin{array}{c}0\\0\\u_\theta^{mp}\end{array}\right\}\right). \tag{3.6.7–7}$$

The terms in Equation 3.6.7–7 can be worked out as follows:

$$\begin{aligned}\frac{\partial\overline{\mathbf{x}}}{\partial\theta}&=\left(\frac{\partial r}{\partial\theta}-u_\theta\right)\mathbf{e}_\mathbf{r}+\frac{\partial z}{\partial\theta}\mathbf{e}_z+\left(r+\frac{\partial u_\theta}{\partial\theta}\right)\mathbf{e}_\theta,\\ \frac{\partial\overline{\mathbf{x}}}{\partial S}&=\frac{\partial r}{\partial S}\mathbf{e}_r+\frac{\partial z}{\partial S}\mathbf{e}_z+\frac{\partial u_\theta}{\partial S}\mathbf{e}_\theta,\end{aligned}$$

and, hence,

$$\begin{aligned}\frac{\partial\overline{\mathbf{x}}}{\partial\theta}\times\frac{\partial\overline{\mathbf{x}}}{\partial S}=&\left[\frac{\partial z}{\partial S}\left(r+\frac{\partial u_\theta}{\partial\theta}\right)-\frac{\partial u_\theta}{\partial S}\frac{\partial z}{\partial\theta}\right]\mathbf{e}_r\\&+\left[\frac{\partial u_\theta}{\partial S}\left(\frac{\partial r}{\partial\theta}-u_\theta\right)-\frac{\partial r}{\partial S}\left(r+\frac{\partial u_\theta}{\partial\theta}\right)\right]\mathbf{e}_z\\&+\left[\frac{\partial r}{\partial S}\frac{\partial z}{\partial\theta}-\frac{\partial z}{\partial S}\left(\frac{\partial r}{\partial\theta}-u_\theta\right)\right]\mathbf{e}_\theta.\end{aligned}$$

The variations $\delta\overline{\mathbf{x}}$ are written $\delta\overline{\mathbf{x}}=\delta\overline{x}_r\mathbf{e}_r+\delta\overline{x}_z\mathbf{e}_z+\delta\overline{u}_\theta\mathbf{e}_\theta$, where the components have interpolations similar to those in Equation 3.6.7–7. Therefore, the virtual work contribution becomes

$$
\begin{aligned}
-\int_A p\,\mathbf{n}\cdot\delta\overline{\mathbf{x}}\,dA = \int_0^{2\pi}\int_{-1}^{+1} p\Bigg\{ & \left[\frac{\partial z}{\partial S}\left(r+\frac{\partial u_\theta}{\partial\theta}\right)-\frac{\partial u_\theta}{\partial S}\frac{\partial z}{\partial\theta}\right]\delta\overline{x}_r \\
& +\left[\frac{\partial u_\theta}{\partial S}\left(\frac{\partial r}{\partial\theta}-u_\theta\right)-\frac{\partial r}{\partial S}\left(r+\frac{\partial u_\theta}{\partial\theta}\right)\right]\delta\overline{x}_z \\
& +\left[\frac{\partial r}{\partial S}\frac{\partial z}{\partial\theta}-\frac{\partial z}{\partial S}\left(\frac{\partial r}{\partial\theta}-u_\theta\right)\right]\delta u_\theta\Bigg\} dS d\theta.
\end{aligned}
$$

With the introduction of the interpolation functions, we obtain the equivalent nodal forces:

$$
\begin{aligned}
F_r^{mp} &= \int_0^{2\pi}\int_{-1}^{+1} p\,H^m(S)R^p(\theta)\left[\frac{\partial z}{\partial S}\left(r+\frac{\partial u_\theta}{\partial\theta}\right)-\frac{\partial u_\theta}{\partial S}\frac{\partial z}{\partial\theta}\right] dS d\theta \\
F_z^{mp} &= \int_0^{2\pi}\int_{-1}^{+1} p\,H^m(S)R^p(\theta)\left[\frac{\partial u_\theta}{\partial S}x\left(\frac{\partial r}{\partial\theta}-u_\theta\right)-\frac{\partial r}{\partial S}\left(r+\frac{\partial u_\theta}{\partial\theta}\right)\right] dS d\theta \\
F_\theta^{mp} &= \int_0^{2\pi}\int_{-1}^{+1} p\,H^m(S)\sin p\theta\left[\frac{\partial r}{\partial S}\frac{\partial z}{\partial\theta}-\frac{\partial z}{\partial S}\left(\frac{\partial r}{\partial\theta}-u_\theta\right)\right] dS d\theta.
\end{aligned}
$$

For geometrically linear analysis, the equivalent nodal forces reduce to the standard axisymmetric expressions

$$
\begin{aligned}
F_r^{mp} &= \int_0^{2\pi}\int_{-1}^{+1} p\,H^m(S)R^p(\theta)\frac{\partial Z}{\partial S}\,R\,dS d\theta \\
F_z^{mp} &= \int_0^{2\pi}\int_{-1}^{+1} -p\,H^m(S)R^p(\theta)\frac{\partial R}{\partial S}\,R\,dS d\theta \\
F_\theta^{mp} &= 0.
\end{aligned}
$$

The linearization of the pressure loading term leads to the following pressure load stiffness matrix:

$$
\begin{Bmatrix} \mathrm{d}F_r^{mp} \\ \mathrm{d}F_z^{mp} \\ \mathrm{d}F_\theta^{mp} \end{Bmatrix} = \begin{bmatrix} K_{rr}^{mpnq} & K_{rz}^{mpnq} & K_{r\theta}^{mpnq} \\ K_{zr}^{mpnq} & K_{zz}^{mpnq} & K_{z\theta}^{mpnq} \\ K_{\theta r}^{mpnq} & K_{\theta z}^{mpnq} & K_{\theta\theta}^{mpnq} \end{bmatrix} \begin{Bmatrix} \mathrm{d}\overline{x}_r^{nq} \\ \mathrm{d}\overline{x}_z^{nq} \\ \mathrm{d}u_\theta^{nq} \end{Bmatrix},
$$

with

$$
\begin{aligned}
K_{rr}^{mpnq} &= \int_0^{2\pi}\int_{-1}^{+1} p\,H^m(S)R^p(\theta)\frac{\partial z}{\partial S}H^n(S)R^q(\theta)\,dS d\theta \\
K_{zr}^{mpnq} &= \int_0^{2\pi}\int_{-1}^{+1} p\,H^m(S)R^p(\theta)\left[-\frac{\partial r}{\partial S}H^n(S)R^q(\theta)-\left(r+\frac{\partial u_\theta}{\partial\theta}\right)\frac{dH^n}{dS}R^q(\theta)\right. \\
&\qquad \left.+\frac{\partial u_\theta}{\partial S}H^n(S)\frac{dR^q}{d\theta}\right] dS d\theta
\end{aligned}
$$

$$K_{\theta r}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1} p\,H^m(S)\sin p\theta\left[\frac{\partial z}{\partial\theta}\frac{dH^n}{dS}R^q(\theta) - \frac{\partial z}{\partial S}H^n(S)\frac{dR^q}{d\theta}\right]dSd\theta$$

$$K_{rz}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1} p\,H^m(S)R^p(\theta)\left[\left(r+\frac{\partial u_\theta}{\partial\theta}\right)\frac{dH^n}{dS}R^q(\theta) - \frac{\partial u_\theta}{\partial S}H^n(S)\frac{dR^q}{d\theta}\right]dSd\theta$$

$$K_{zz}^{mpnq} = 0$$

$$K_{\theta z}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1} p\,H^m(S)\sin p\theta\left[\frac{\partial r}{\partial S}H^n(S)\frac{dR^q}{d\theta} - \left(\frac{\partial r}{\partial\theta}-u_\theta\right)\frac{dH^n}{dS}R^q(\theta)\right]dSd\theta$$

$$K_{r\theta}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1} p\,H^m(S)R^p(\theta)\left[q\,\frac{\partial z}{\partial S}H^n(S)\cos q\theta - \frac{\partial z}{\partial\theta}\frac{dH^n}{dS}\sin q\theta\right]dSd\theta$$

$$K_{z\theta}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1} p\,H^m(S)R^p(\theta)\left[-\frac{\partial u_\theta}{\partial S}H^n(S)\sin q\theta + \left(\frac{\partial r}{\partial\theta}-u_\theta\right)\frac{dH^n}{dS}\sin q\theta - q\,\frac{\partial r}{\partial S}H^n(S)\cos q\theta\right]dSd\theta$$

$$K_{\theta\theta}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1} p\,H^m(S)\sin p\theta\,\frac{\partial z}{\partial S}\,H^n(S)\sin q\theta\,dSd\theta.$$

In the case of hydrostatic pressure (***p*** dependent on ***z***), additional terms must be included in the pressure load stiffness. These terms appear due to the variation of the pressure magnitude and are readily obtained from the expression

$$-\int_A \mathrm{d}p\,\mathbf{n}\cdot\delta\mathbf{u}\,dA = \int_0^{2\pi}\int_{-1}^{+1}\mathrm{d}u_z\frac{dp}{dz}\left\{\left[\frac{\partial z}{\partial S}\left(r+\frac{\partial u_\theta}{\partial\theta}\right) - \frac{\partial u_\theta}{\partial S}\frac{\partial z}{\partial\theta}\right]\delta\overline{x}_r + \left[\frac{\partial u_\theta}{\partial S}\left(\frac{\partial r}{\partial\theta}-u_\theta\right) - \frac{\partial r}{\partial S}\left(r+\frac{\partial u_\theta}{\partial\theta}\right)\right]\delta\overline{x}_z + \left[\frac{\partial r}{\partial S}\frac{\partial z}{\partial\theta} - \frac{\partial z}{\partial S}\left(\frac{\partial r}{\partial\theta}-u_\theta\right)\right]\delta u_\theta\right\}dSd\theta.$$

With use of the interpolation functions and denoting the additional contributions with an over-bar, we obtain the additional load stiffness contributions:

$$\overline{K}_{rz}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1} H^m(S)R^p(\theta)\frac{dp}{dz}\left[\frac{\partial z}{\partial S}\left(r+\frac{\partial u_\theta}{\partial\theta}\right) - \frac{\partial u_\theta}{\partial S}\frac{\partial z}{\partial\theta}\right]H^n(S)R^q(\theta)\,dSd\theta$$

$$\overline{K}_{zz}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1} H^m(S)R^p(\theta)\frac{dp}{dz}\left[\frac{\partial u_\theta}{\partial S}\left(\frac{\partial r}{\partial\theta}-u_\theta\right) - \frac{\partial r}{\partial S}\left(r+\frac{\partial u_\theta}{\partial\theta}\right)\right]H^n(S)R^q(\theta)\,dSd\theta$$

$$\overline{K}_{\theta z}^{mpnq} = \int_0^{2\pi}\int_{-1}^{+1} H^m(S)\sin p\theta\frac{dp}{dz}\left[\frac{\partial r}{\partial S}\frac{\partial z}{\partial\theta} - \frac{\partial z}{\partial S}\left(\frac{\partial r}{\partial\theta}-u_\theta\right)\right]H^n(S)R^q(\theta)\,dSd\theta.$$

Penalty constraints: transverse shear and drill rotation

It is necessary to enforce rotation constraints at selected points on the element surface to prevent singular modes of deformation. One axial transverse shear constraint is enforced on the ϕ_θ rotation field between each pair of nodes within each nodal plane. One circumferential transverse shear constraint and one drill rotation constraint on the rotation fields ϕ_r and ϕ_z are enforced between each pair of nodes on a circumferential line of nodes. In each instance the rotation field is constrained to follow the nodal displacements. To summarize, for element SAXA*MN*:

- Axial transverse shear: $M(N+1)$
- Circumferential transverse shear: $(M+1)N$
- Drill rotation: $(M+1)N$

constraints are enforced, where $M \in \{1,2\}$ is the order of integration in the axial direction and $N \in \{1,2,3,4\}$ is the number of Fourier modes.

Transverse shear

The transverse shear strain is the measure of the amount the director field $\mathbf{t}_3$ has rotated relative to the normal to the shell surface. We define the transverse shear strain as

$$\gamma_{3c} \stackrel{\text{def}}{=} \mathbf{t}_3 \cdot \overline{\mathbf{t}}_c\,, \quad \text{where} \quad \overline{\mathbf{t}}_c \stackrel{\text{def}}{=} \overline{\mathbf{x}}_{,\xi^c}/\|\overline{\mathbf{x}}_{,\xi^c}\|,$$

with no sum on c and force this quantity to be zero with a penalty constraint. Note that $\overline{\mathbf{t}}_c$ is a unit vector tangent to the shell surface defined by the displacement field along a parametric coordinate line. Recall that $\xi^1 = S$ and $\xi^2 = \theta$. Hence, γ_{31} is the axial transverse shear strain and γ_{32} is the circumferential transverse shear strain.

For convenience, record the variation of the unit vector $\overline{\mathbf{t}}_c$:

$$\delta\overline{\mathbf{t}}_c = \frac{1}{\|\overline{\mathbf{x}}_{,\xi^c}\|}\delta\overline{\mathbf{x}}_{,\xi^c} - \frac{\overline{\mathbf{x}}_{,\xi^c}\cdot\delta\overline{\mathbf{x}}_{,\xi^c}}{\|\overline{\mathbf{x}}_{,\xi^c}\|^3}\overline{\mathbf{x}}_{,\xi^c}\,,$$

with no sum on c. Equivalently, the definition of $\overline{\mathbf{t}}_c$ can be used to write

$$\delta\overline{\mathbf{t}}_c = \frac{1}{\|\overline{\mathbf{x}}_{,\xi^c}\|}\left[\delta\overline{\mathbf{x}}_{,\xi^c} - (\overline{\mathbf{t}}_c\cdot\delta\overline{\mathbf{x}}_{,\xi^c})\mathbf{t}_c\right].$$

The linearized transverse shear strain is calculated as

$$\delta\gamma_{3c} = \delta\mathbf{t}_3\cdot\overline{\mathbf{t}}_c + \mathbf{t}_3\cdot\delta\overline{\mathbf{t}}_c\,.$$

Since by definition $\delta\mathbf{t}_3 = \delta\boldsymbol{\phi}\times\mathbf{t}_3$, it follows that

$$\delta\gamma_{3c} = \delta\boldsymbol{\phi}\cdot(\mathbf{t}_3\times\overline{\mathbf{t}}_c) + \frac{1}{\|\overline{\mathbf{x}}_{,\xi^c}\|}(\mathbf{t}_3 - \gamma_{3c}\overline{\mathbf{t}}_c)\cdot\delta\overline{\mathbf{x}}_{,\xi^c}\,,$$

with no sum on c.

For completeness, the second variation of the transverse shear strain is

$$\mathrm{d}(\delta\gamma_{3c}) = \delta\boldsymbol{\phi}\cdot\tfrac{1}{2}[\mathbf{t}_3\overline{\mathbf{t}}_c+\overline{\mathbf{t}}_c\mathbf{t}_3]\cdot\mathrm{d}\boldsymbol{\phi} - \delta\overline{\mathbf{x}}_{,\xi^c}\cdot\frac{1}{\|\overline{\mathbf{x}}_{,\xi^c}\|^2}[\mathbf{t}_3\overline{\mathbf{t}}_c+\overline{\mathbf{t}}_c\mathbf{t}_3]\cdot\mathrm{d}\overline{\mathbf{x}}_{,\xi^c}$$
$$+\delta\boldsymbol{\phi}\cdot\frac{1}{\|\overline{\mathbf{x}}_{,\xi^c}\|}[\hat{\mathbf{t}}_3-(\mathbf{t}_3\times\overline{\mathbf{t}}_c)\overline{\mathbf{t}}_c]\cdot\mathrm{d}\overline{\mathbf{x}}_{,\xi^c}+\delta\overline{\mathbf{x}}_{,\xi^c}\cdot\frac{1}{\|\overline{\mathbf{x}}_{,\xi^c}\|}[-\hat{\mathbf{t}}_3-\overline{\mathbf{t}}_c(\mathbf{t}_3\times\overline{\mathbf{t}}_c)]\cdot\mathrm{d}\boldsymbol{\phi}\,,$$

where terms proportional to γ_{3c} have been neglected and the coupling between $\delta\boldsymbol{\phi}$ and $\Delta\boldsymbol{\phi}$ has been symmetrized.

Drill rotation

Mathematically, the equations governing the deformation of the shell are invariant with respect to drill rotations; that is, a rotation of the director field with axis of rotation parallel to the director. It is necessary then to assign a kinematic definition to this rotation. We define the drill strain as the difference between the rotation of the circumferential tangent vector as measured by the displacement field and that measured by the rotation field. Accordingly, we define the drill strain by

$$\gamma_d = \mathbf{a}\cdot\mathbf{b}\,.$$

Here, $\mathbf{a}$ is the rotated reference axial tangent vector and $\mathbf{b}$ is the deformed (unit) circumferential tangent vector, defined by

$$\mathbf{a} = \exp[\hat{\boldsymbol{\phi}}]\mathbf{A} \quad\text{and}\quad \mathbf{b} = \frac{\overline{\mathbf{x}}_{,\theta}}{\|\overline{\mathbf{x}}_{,\theta}\|}\,.$$

In the above $\mathbf{A}$ is a linear approximation to the axial tangent vector in the reference configuration given by $\mathbf{A}_1 = (\mathbf{X}^K-\mathbf{X}^N)/\|\mathbf{X}^K-\mathbf{X}^N\|$, where $\mathbf{X}^K$ and $\mathbf{X}^N$ are the position vectors to two adjacent nodes in an axial plane. The drill rotation constraint then requires that the component of rotation about the normal to the surface match the in-plane rotation of the surface as measured by the displacement field.

The linearized drill strain calculation is similar to the transverse shear linearized strain calculation. Without repeating the calculation,

$$\delta\gamma_d = \delta\boldsymbol{\phi}\cdot(\mathbf{a}\times\mathbf{b}) + \frac{1}{\|\overline{\mathbf{x}}_{,\theta}\|}(\mathbf{a}-\gamma_d\mathbf{b})\cdot\delta\overline{\mathbf{x}}_{,\theta}\,.$$

Similarly, the second variation of the drill strain

$$\mathrm{d}(\delta\gamma_d) = \delta\boldsymbol{\phi}\cdot\tfrac{1}{2}[\mathbf{ab}+\mathbf{ba}]\cdot\mathrm{d}\boldsymbol{\phi} - \delta\overline{\mathbf{x}}_{,\theta}\cdot\frac{1}{\|\overline{\mathbf{x}}_{,\theta}\|^2}[\mathbf{ab}+\mathbf{ba}]\cdot\mathrm{d}\overline{\mathbf{x}}_{,\theta}$$
$$+\delta\boldsymbol{\phi}\cdot\frac{1}{\|\overline{\mathbf{x}}_{,\theta}\|}[\hat{\mathbf{a}}-(\mathbf{a}\times\mathbf{b})\mathbf{b}]\cdot\mathrm{d}\overline{\mathbf{x}}_{,\theta}+\delta\overline{\mathbf{x}}_{,\theta}\cdot\frac{1}{\|\overline{\mathbf{x}}_{,\theta}\|}[-\hat{\mathbf{a}}-\mathbf{b}(\mathbf{a}\times\mathbf{b})]\cdot\mathrm{d}\boldsymbol{\phi}\,,$$

where terms proportional to γ_d have been neglected and the coupling between $\delta\boldsymbol{\phi}$ and $\Delta\boldsymbol{\phi}$ has been symmetrized.

Zero radius: collapsed edge

For the case when the reference radius of any point on the shell surface goes to zero, all of the offset nodes collapse to the same point and the edge constraints along that circumferential edge become redundant. It is, therefore, necessary to treat the zero radius case separately.

For the zero radius case all redundant degrees of freedom are constrained to follow the average motion of the nodes at $\theta = 0°$ and $\theta = 180°$. The edge constraints are broken into two parts: First, a circumferential transverse shear strain is defined that requires that the rotation in the radial direction follow the circumferential rotations at the first and last nodal plane:

$$\gamma_{32}^0 = \phi_r + \tfrac{1}{2}(\phi_\theta^1 + \phi_\theta^{P+1}) \sin\theta \,.$$

Introducing the interpolations, the linearized strain is

$$\delta\gamma_{32}^0 = \sum_{p=1}^{P} \sin p\theta \, \delta\phi_r^p + \tfrac{1}{2} \sin\theta (\delta\phi_\theta + \delta\phi_\theta^{P+1}) \,.$$

Note that $\mathrm{d}(\delta\gamma_{32}^0) = 0$. Second, a drill rotation strain is defined that requires that the rotation about the Z-axis be zero:

$$\gamma_d^0 = \phi_z \,.$$

This leads to a linearized strain given by

$$\delta\gamma_d^0 = \sum_{p=1}^{P} \sin p\theta \, \delta\phi_z^p \,.$$

Note that $\mathrm{d}(\delta\gamma_d^0) = 0$.

Mass matrix

At each material point the displacement components in the three directions (radial, axial, circumferential) are dependent only on the corresponding nodal displacement components. Hence, the mass matrix does not involve any coupling between the radial, axial, and circumferential degrees of freedom, and we can write the mass matrix in the form of three separate expressions:

$$\begin{aligned}
M_{u_r}^{mnpq} &= \int_z \rho \, \mathrm{d}z \int_A N_r^{mp} N_r^{nq} \mathrm{d}A \\
M_{u_z}^{mnpq} &= \int_z \rho \, \mathrm{d}z \int_A N_z^{mp} N_z^{nq} \mathrm{d}A \\
M_{u_\theta}^{mnpq} &= \int_z \rho \, \mathrm{d}z \int_A N_\theta^{mp} N_\theta^{nq} \mathrm{d}A.
\end{aligned}$$

Similarly, we can write the terms associated with the rotational degrees of freedom as:

$$M_{\phi_r}^{mnpq} = \int_z \rho\, z^2 \mathrm{d}z \int_A N_\theta^{mp} N_\theta^{nq} \mathrm{d}A$$
$$M_{\phi_z}^{mnpq} = \int_z \rho\, z^2 \mathrm{d}z \int_A N_\theta^{mp} N_\theta^{nq} \mathrm{d}A$$
$$M_{\phi_\theta}^{mnpq} = \int_z \rho\, z^2 \mathrm{d}z \int_A N_r^{mp} N_r^{nq} \mathrm{d}A.$$

Here the superscripts m and n refer to a particular node in the r–z plane, and the superscripts p and q refer to a particular position along the circumference. The interpolation functions N_r^{mp}, N_z^{mp}, and N_θ^{mp} are the product of interpolation functions $H^m(S)$ in the r–z plane and interpolation functions in the θ-direction:

$$N_r^{mp} = H^m(S) R^p(\theta) = N_z^{mp}$$
$$N_\theta^{mp} = H^m(S) \sin\, p\theta.$$

The area integral used to form the mass matrix can be split then into an integral along the length of the element in the r–z plane and an integral around the circumference. For the r–r component of the mass matrix this yields

$$M_{u_r}^{mnpq} = \int_z \rho \,\mathrm{d}z \int_S H^m(S) H^n(S)\, 2\pi R \,\mathrm{d}S \,\frac{1}{2\pi} \int_0^{2\pi} R^p(\theta) R^q(\theta)\, \mathrm{d}\theta,$$

and for the ϕ_r component of the mass matrix this yields:

$$M_{\phi_r}^{mnpq} = \int_z \rho\, z^2 \mathrm{d}z \int_S H^m(S) H^n(S)\, 2\pi R \,\mathrm{d}S \,\frac{1}{2\pi} \int_0^{2\pi} R^p(\theta) R^q(\theta)\, \mathrm{d}\theta.$$

The matrix can be written in a convenient form by defining the primitive mass matrix as

$$M_{prim}^{mn} = \int_z \rho \,\mathrm{d}z \int_S H^m(S) H^n(S)\, 2\pi R \,\mathrm{d}S,$$

or for the rotational components as

$$\overline{M}_{prim}^{mn} = \int_z \rho\, z^2 \mathrm{d}z \int_S H^m(S) H^n(S)\, 2\pi R \,\mathrm{d}S.$$

These primitive mass matrices are the same mass matrices that are used for the regular axisymmetric shell elements. We can also define the circumferential distribution matrices

$$f_1^{pq} = \frac{1}{2\pi} \int_0^{2\pi} R^p(\theta) R^q(\theta)\, \mathrm{d}\theta \quad \text{and}$$
$$f_2^{pq} = \frac{1}{2\pi} \int_0^{2\pi} \sin p\theta\, \sin q\theta\, \mathrm{d}\theta.$$

The various components of the mass matrix then take the form

$$
\begin{aligned}
M_{u_r}^{mnpq} &= M_{prim}^{mn} f_1^{pq} \\
M_{u_z}^{mnpq} &= M_{prim}^{mn} f_1^{pq} \\
M_{u_\theta}^{mnpq} &= M_{prim}^{mn} f_2^{pq} \\
M_{\phi_r}^{mnpq} &= \overline{M}_{prim}^{mn} f_2^{pq} \\
M_{\phi_z}^{mnpq} &= \overline{M}_{prim}^{mn} f_2^{pq} \\
M_{\phi_\theta}^{mnpq} &= \overline{M}_{prim}^{mn} f_1^{pq}.
\end{aligned}
$$

The circumferential distribution matrices can be evaluated for various values of the number of terms P in the Fourier series. After some calculations the following results are obtained:

$P = 1$:

$$
f_1^{pq} = \frac{1}{8}\begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}, \qquad f_2^{pq} = \frac{1}{2}\begin{pmatrix} 1 \end{pmatrix}.
$$

$P = 2$:

$$
f_1^{pq} = \frac{1}{32}\begin{pmatrix} 7 & 2 & -1 \\ 2 & 12 & 2 \\ -1 & 2 & 7 \end{pmatrix}, \qquad f_2^{pq} = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.
$$

$P = 3$:

$$
f_1^{pq} = \frac{1}{72}\begin{pmatrix} 11 & 2 & -2 & 1 \\ 2 & 20 & 4 & -2 \\ -2 & 4 & 20 & 2 \\ 1 & -2 & 2 & 11 \end{pmatrix}, \qquad f_2^{pq} = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.
$$

$P = 4$:

$$
f_1^{pq} = \frac{1}{128}\begin{pmatrix} 15 & 2 & -2 & 2 & -1 \\ 2 & 28 & 4 & -4 & 2 \\ -2 & 4 & 28 & 4 & -2 \\ 2 & -4 & 4 & 28 & 2 \\ -1 & 2 & -2 & 2 & 15 \end{pmatrix}, \qquad f_2^{pq} = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.
$$

Reference

- "Axisymmetric shell elements with nonlinear, asymmetric deformation," Section 29.6.10 of the Abaqus Analysis User's Guide

3.6.8 TRANSVERSE SHEAR STIFFNESS IN COMPOSITE SHELLS AND OFFSETS FROM THE MIDSURFACE

Products: Abaqus/Standard Abaqus/Explicit

Transverse shear stiffness

Abaqus supports element types S3R, S3RS, S4, S4R, S4RS, S4RSW, SC6R, SC8R, and S8R for the analysis of laminated composite shells. These elements are based on first-order transverse shear flexible theory in which the transverse shear strain is assumed to be constant through the thickness of the shell. This assumption necessitates the use of shear correction factors. The development of these factors also provides a basis for estimating interlaminar shear stresses in a composite section. This section describes the development of the transverse shear stiffness for element types S3R, S4, S4R, SC6R, SC8R, and S8R.

Finite-strain shells

The transverse shear stiffness correction factors are easily shown to be $5/6$ for isotropic plates. We want to establish the equivalent factors for laminated plates and sandwich constructions. For this purpose we calculate the distribution of transverse shear stress through the thickness of the shell, for the case of unidirectional bending and assuming linear elastic response. Then the shear strain energy, expressed in terms of section forces and strains, is equated to the strain energy of this distribution of transverse shear stresses.

This method, outlined below, provides an approximate method for calculating interlaminar shear stresses and supplies reasonable estimates of transverse shear stiffness. In this calculation Abaqus assumes that the shell section directions are the principal bending directions (bending about one principal direction does not require a restraining moment about the other direction). For composite shells with orthotropic layers that are not symmetric about the shell midsurface, the shell section directions may not be the principal bending directions. In such cases the transverse shear stiffness and interlaminar shear stresses are less accurate approximations and will change if different shell section directions are used.

Consider a plate in the (x, y) plane. Assume only bending and shear in the x-direction, without gradients in the y-direction. Then the membrane forces in the shell are zero: $N_{xx} = N_{yy} = N_{xy} = 0$, and $\partial/\partial y = 0$ for all response variables. In this case equilibrium within the section in the x-direction is

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \tau_{xz}}{\partial z} = 0. \tag{3.6.8–1}$$

Moment equilibrium about the y-axis gives

$$V_x + \frac{\partial M_{xx}}{\partial x} = 0, \tag{3.6.8–2}$$

where V_x is the transverse shear force per unit width in the plate and M_{xx} is the bending moment per unit width for bending about the y-axis.

For the bending behavior we assume the strain varies linearly across the section:

$$\epsilon_{\alpha\beta} = \overline{\epsilon}_{\alpha\beta} - (z - z_0)\overline{\kappa}_{\alpha\beta},$$

where $\overline{\epsilon}_{\alpha\beta}$ is the membrane strain of the reference surface $z = z_0$ and $\overline{\kappa}_{\alpha\beta}$ is the curvature of that surface.

If the response of the shell is linear elastic, any in-plane component of stress at a point through the shell section is given by

$$\sigma_{\alpha\beta} = D_{\alpha\beta\gamma\delta}(\overline{\epsilon}_{\gamma\delta} - (z - z_0)\overline{\kappa}_{\gamma\delta}), \tag{3.6.8–3}$$

where the plane stress elastic stiffness, $D_{\alpha\beta\gamma\delta}$, is defined from the elasticity and orientation of the material at the particular layer of the shell. Greek subscripts take the range $(1, 2)$.

Integrating through the thickness and inverting the resultant section stiffness provides the 6×6 section flexibility matrix, $[H]$:

$$\left\{ \begin{array}{c} \overline{\epsilon}_{\alpha\beta} \\ \overline{\kappa}_{\alpha\beta} \end{array} \right\} = [H] \left\{ \begin{array}{c} N_{\alpha\beta} \\ M_{\alpha\beta} \end{array} \right\}.$$

We have already assumed that $N_{xx} = N_{yy} = N_{xy}$. We now also assume that $M_{yy} = M_{xy} = 0$; that is, that it is possible to have no bending in the y-direction without any restraining moments associated with the y-direction. This is clearly not the case for an unbalanced composite section, but we still use it as a simplifying assumption to obtain the shear correction factors. Thus,

$$\left\{ \begin{array}{c} \overline{\epsilon}_{\alpha\beta} \\ \overline{\kappa}_{\alpha\beta} \end{array} \right\} = \{H_{i4}\}\, M_{xx}, \tag{3.6.8–4}$$

where $\{H_{i4}\}$ is the fourth column of $[H]$. Combining this result with the elastic stiffness at a point through the shell thickness provides the in-plane stress components in terms of M_{xx} as

$$\sigma_{xx} = (B_{x1} - (z - z_0)B_{x2})\, M_{xx}, \tag{3.6.8–5}$$

where

$$B_{x1} = D_{xxxx}H_{14} + D_{xxyy}H_{24} + D_{xxxy}H_{34}$$

and

$$B_{x2} = D_{xxxx}H_{44} + D_{xxyy}H_{54} + D_{xxxy}H_{64}.$$

Combining the gradient of this equation in the x-direction with the equilibrium equations Equation 3.6.8–1 and Equation 3.6.8–2 yields a description of the variation of the transverse shear stress through the thickness of the plate:

$$\frac{\partial \tau_{xz}}{\partial z} = \left(B_{x1} - (z - z_0)B_{x2}\right) V_x. \tag{3.6.8–6}$$

In calculating $\partial \sigma_{xx}/\partial x$ we have assumed that the elasticity and thickness of the composite section do not vary (or vary slowly) with position along the shell.

A laminated composite shell section consists of N layers $1, 2, 3, \ldots$ with different values of (B^1_{x1},B^1_{x2}) at layer 1, (B^2_{x1},B^2_{x2}) at layer 2, ... (B^N_{x1},B^N_{x2}) at layer N. Layer i extends from z_i to z_{i+1} and its thickness is $t_i = z_{i+1} - z_i$. Integrating Equation 3.6.8–6 through the shell, using the boundary conditions $\tau_{xz} = 0$ at $z = 0$, $\tau^i_{xz} = \tau^{i+1}_{xz}$ at $z = z_{i+1}$ and $\tau_{xz} = 0$ at $z = z_{N+1}$, gives the transverse shear stress in layer i as

$$\tau^i_{xz} = \left[B^i_{x1}(z - z_i) - \left(\frac{1}{2}\left(z^2 - {z_i}^2\right) - z_{x0}(z - z_i)\right) B^i_{x2} + B^i_{x0}\right] V_x, \tag{3.6.8–7}$$

where

$$B^i_{x0} = \sum_{j=1}^{i-1} t_j \left[B^j_{x1} - \left(\frac{1}{2}(z_{j+1} + z_j) - z_{x0}\right) B^j_{x2}\right]$$

and

$$z_{x0} = \frac{\sum_{i=1}^{N} t_i \left(\frac{1}{2}(z_{i+1} + z_i)B^i_{x2} - B^i_{x1}\right)}{\sum_{i=1}^{N} t_i B^i_{x2}}.$$

The subscript z_{x0} is used instead of z_0 in this case because the result is associated with pure bending in the x-direction.

The variation of τ_{yz} through the shell thickness is obtained using a similar procedure, based on pure bending in the y-direction.

These results provide the estimates of interlaminar shear stresses.

We define the shear flexibility of the section by matching the shear strain energy obtained by integrating the elastic strain energy density associated with transverse shear stress distribution obtained above:

$$\frac{1}{2} \lfloor V_x \quad V_y \rfloor \left[F^s\right] \left\{ \begin{matrix} V_x \\ V_y \end{matrix} \right\} = \frac{1}{2} \sum_{i=1}^{N} \int_{z_i}^{z_{i+1}} \lfloor \tau_{xz} \quad \tau_{yz} \rfloor \left[F^i\right] \left\{ \begin{matrix} \tau_{xz} \\ \tau_{yz} \end{matrix} \right\} dz,$$

where $\left[F^s\right]$ is the shear flexibility of the section and $\left[F^i\right]$ is the continuum transverse shear flexibility within layer i. Here we introduce the assumption that the transverse shear flexibility within a layer is not coupled to the in-plane flexibility. This is usually the case for shell constructions.

Substituting the relations for τ^i_{xz} and τ^i_{yz} into the above equation and integrating defines the shear flexibility of the section as

$$F^s_{xx} = \sum_{i=1}^{N} F^i_{xx} t_i \Bigg[(B^i_{x0})^2 + t_i B^i_{x0} \left(B^i_{x1} - (z_i - z_{x0}) B^i_{x2} \right) + \frac{1}{3} t_i^2 \left(B^i_{x1} - (z_i - z_{x0}) B^i_{x2} \right)^2 - \\ - \frac{1}{4} t_i^3 B^i_{x2} \left(B_{x1} - (z_i - z_{x0}) B^i_{x2} \right)^2 + \frac{1}{20} t_i^4 (B^i_{x2})^2 \Bigg]$$

$$F^s_{yy} = \sum_{i=1}^{N} F^i_{yy} t_i \Bigg[(B^i_{y0})^2 + t_i B^i_{y0} \left(B^i_{y1} - (z_i - z_{y0}) B^i_{y2} \right) + \frac{1}{3} t_i^2 \left(B^i_{y1} - (z_i - z_{y0}) B^i_{y2} \right)^2 - \\ - \frac{1}{4} t_i^3 B^i_{y2} (B_{y1} - (z_i - z_{y0}) B^i_{y2})^2 + \frac{1}{20} t_i^4 (B^i_{y2})^2 \Bigg]$$

$$F^s_{xy} = \sum_{i=1}^{N} F^i_{xy} t_i \Bigg[B^i_{x0} B^i_{y0} + \frac{1}{2} t_i \left[B^i_{x0} \left(B^i_{y1} - (z_i - z_{y0}) B^i_{y2} \right) + B^i_{y0} (B_{x1} - (z_i - z_{x0}) B^i_{x2} \right] + \\ + \frac{1}{3} t_i^2 (B^i_{x1} - (z_i - z_{x0}) B^i_{x2})(B^i_{y1} - (z_i - z_{y0}) B^i_{y2}) - \\ - \frac{1}{8} t_i^3 \left[B^i_{x2} (B^i_{y1} - (z_0 - z_{y0}) B^i_{y2}) + B^i_{y2} (B^i_{x1} - (z_0 - z_{x0}) B^i_{x2}) \right] + \\ + \frac{1}{20} t_i^4 B^i_{x2} B^i_{y2} \Bigg].$$

The transverse shear stiffness of the section is then available as $\left[F^s_{\alpha\beta} \right]^{-1}$. Notice that F^s_{12} will be nonzero if any layer is anisotropic or orthotropic in a local system (since then F^i_{xy} will be nonzero).

Small-strain shells

When the shell resultant forces at each increment are computed for pre-integrated sections, the transverse shear forces for small-strain shell elements S3RS, S4RS, and S4RSW are computed using the transverse shear stiffness derived for finite-strain shells. For numerically integrated sections the transverse shear behavior is based on a simplified stiffness for improved computational performance. For single or multilayer isotropic sections and single layer orthotropic sections, the transverse shear force converges to the proper thin and thick shell transverse shear solution and the transverse shear stress is assumed to have a constant distribution. The transverse shear stiffness is approximate for multilayer orthotropic sections, where the transverse shear stress distribution is assumed piecewise constant. Convergence to the proper transverse shear behavior for this case may not be obtained as shells become thick and principal material directions deviate from the principal section directions.

Offset: reference surface versus midsurface

In Abaqus the geometry of the shell is defined by kinematic variables that exist at the nodes on the shell reference surface. The kinematics of the shell theory consist of measuring membrane strain on the reference surface and bending strain from the derivatives of the unit normal vector on the reference

surface. The default reference surface is the shell midsurface. However, many situations arise in which it is more convenient to define the reference surface as offset from the midsurface. In this case we assume that the in-plane strain at any material point varies linearly across the section:

$$\epsilon_{\alpha\beta} = \overline{\epsilon}_{\alpha\beta} - (z - z_0)\overline{\kappa}_{\alpha\beta},$$

where α and β represent the two orthogonal axes on the reference surface, $\overline{\epsilon}_{\alpha\beta}$ is the membrane strain of the reference surface, z_0 is the distance to the reference surface measured from the midsurface, and $\overline{\kappa}_{\alpha\beta}$ is the curvature of that surface. The positive values of z_0 are in the positive normal direction. When $z_0 = t/2$, the top surface of the shell is the reference surface, where t is the shell thickness. The bottom surface of the shell becomes the reference surface when $z_0 = -t/2$. When $z_0 = 0$, the midsurface represents the reference surface.

If the response of the shell is linear elastic, any in-plane component of stress at a point through the shell section, $\sigma_{\alpha\beta}$, is given by

$$D_{\alpha\beta\gamma\delta}(\overline{\epsilon}_{\gamma\delta} - (z - z_0)\overline{\kappa}_{\gamma\delta}),$$

where the plane stress elastic stiffness, $D_{\alpha\beta\gamma\delta}$, is defined from the elasticity and orientation of the material at the particular layer of the shell. Greek subscripts take the range $(1, 2)$.

The section force and moment resultants per unit length can then be defined as

$$N_{\alpha\beta} = \int_{-z_0-t/2}^{-z_0+t/2} \sigma_{\alpha\beta} dz,$$
$$M_{\alpha\beta} = \int_{-z_0-t/2}^{-z_0+t/2} \sigma_{\alpha\beta} z dz.$$

Integrating the above equations through the thickness leads to the resultant section stiffness, $[A]$:

$$\left\{ \begin{array}{c} N_{\alpha\beta} \\ M_{\alpha\beta} \end{array} \right\} = [A] \left\{ \begin{array}{c} \overline{\epsilon}_{\alpha\beta} \\ \overline{\kappa}_{\alpha\beta} \end{array} \right\}.$$

Reference

- "Shell section behavior," Section 29.6.4 of the Abaqus Analysis User's Guide

3.6.9 ROTARY INERTIA FOR 5 DEGREE OF FREEDOM SHELL ELEMENTS

Product: Abaqus/Standard

Some of the shell elements in Abaqus/Standard (S4R5, S8R5, S9R5, and STRI65) use two variables at a node to define the change in the shell normal at the node, $\mathbf{n}^N$, during an increment. At some nodes in these elements and in other elements we use the three components of the rotation triplet as the rotation degrees of freedom. To provide inertia for all of these nodal variables, at a node N with two "rotation" variables we define

$$\partial^p \mathbf{n}^N = \partial^p \phi_2^{\overline{N}} \mathbf{e}_1^{\overline{N}} - \partial^p \phi_1^{\overline{N}} \mathbf{e}_2^{\overline{N}},$$

where ∂^p denotes any time derivative or variation of the following quantity and the $\mathbf{e}_\alpha^N$ are the basis vectors used to define the rotational variables at the node during this increment. Barred superscripts are not summed.

This expression neglects any rotation of the basis system that occurs during the increment. This is an approximation in large-displacement analysis: it is adopted for the sake of simplicity, based on the argument that we are not attempting to model the rotary inertia accurately.

At a node at which we use the three global rotation components we define

$$\partial^p \mathbf{n}^N = \partial^p \phi_i^{\overline{N}} \mathbf{e}_i^{\overline{N}},$$

where in this case the $\mathbf{e}_i$ are the global Cartesian basis vectors. This definition is artificial and serves simply to associate a reasonable measure of inertia with the rotational degrees of freedom.

The first-order elements in Abaqus use a lumped mass matrix. In this case M^{NM} is diagonal, so that the rotary inertia contribution at node N is

$$\eta M^{\overline{NN}} \delta\phi_\alpha^{\overline{N}} \ddot{\phi}_\alpha^{\overline{N}},$$

where α sums over the number of rotation components used at the node (2 or 3).

For a consistent mass element the rotary inertia contribution is

$$\sum_{M=1}^{\text{nodes}} \sum_{N=1}^{\text{nodes}} M^{NM} \delta\phi_\alpha^N \ddot{\phi}_\beta^M \mathbf{e}_\alpha^N \cdot \mathbf{e}_\beta^M,$$

where α and β sum over the number of rotation components used at each node (2 or 3).

The time integration algorithms require initial conditions for each increment. For implicit integration these are the velocities and accelerations of the variables at the start of the increment, $\dot{u}|^0$ and $\ddot{u}|^0$.

At a node where three global rotation components are used, these initial values are directly available from the solution to the previous increment. At a node where only two variables define the rotation, we convert variables from the basis of one increment to that of the next through the approximation

$$\partial^p \phi_\alpha|^+ = \partial^p \phi_\beta|^- \mathbf{e}_\alpha|^+ \cdot \mathbf{e}_\beta|^-,$$

where $^-$ indicates a variable associated with the previous increment and $^+$ indicates a variable associated with the current increment. The justifications for this approximation are that it is simple, the incremental rotation will not be large anyway, and we are not trying to model the rotary inertia effect with high accuracy.

Reference

- "Choosing a shell element," Section 29.6.2 of the Abaqus Analysis User's Guide

3.7 Rebar

- “Rebar modeling in two dimensions,” Section 3.7.1
- “Rebar modeling in three dimensions,” Section 3.7.2
- “Rebar modeling in shell, membrane, and surface elements,” Section 3.7.3

3.7.1 REBAR MODELING IN TWO DIMENSIONS

Products: Abaqus/Standard Abaqus/Explicit

Let $g_i,\ i = 1,\ 2$ be the element's usual isoparametric coordinates. Let r be an isoparametric coordinate along the line where the face of the element intersects the plane of reinforcement, with $-1 \le r \le 1$ in an element (see Figure 3.7.1–1).

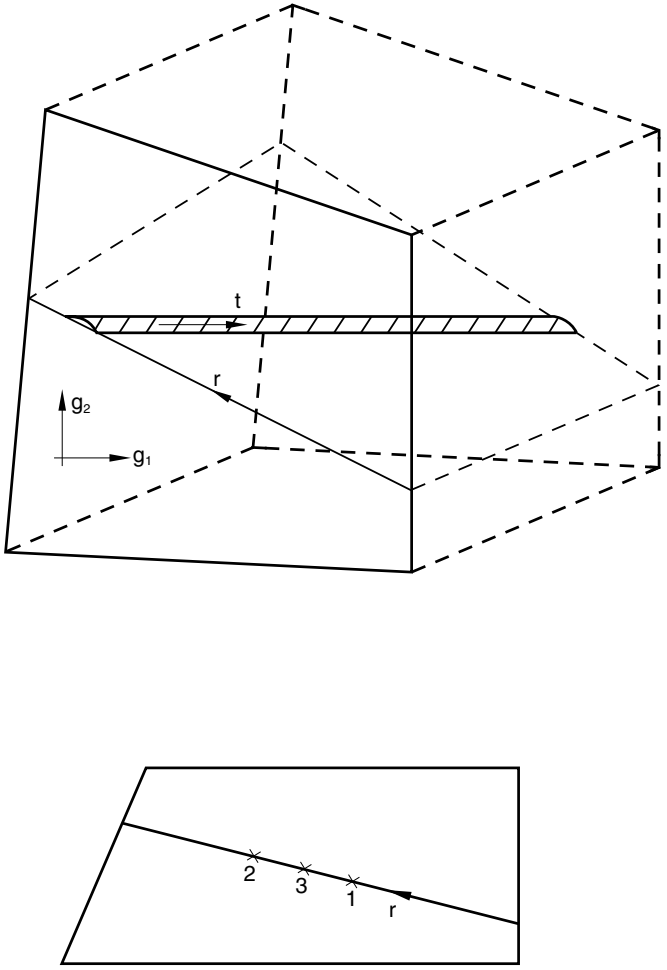


Figure 3.7.1–1 Rebar in a solid, two-dimensional element.

The plane of reinforcement is always perpendicular to the element face.

The rebar will be integrated at one or two points, depending on the order of interpolation in underlying elements. The volume of integration (ΔV), position, rebar strain (ε), and first and second variations of rebar strain ($\delta\varepsilon$ and $d\delta\varepsilon$) at each point are calculated as

$$\Delta V = \frac{A_r}{S_r}\left(\frac{\partial \mathbf{x}}{\partial r}\cdot\frac{\partial \mathbf{x}}{\partial r}\right)^{\frac{1}{2}} t_0\, W_N,$$

where

t_0	is the original thickness for plane elements and $2\pi x_1$ for axisymmetric elements;
A_r	is the rebar cross-sectional area;
S_r	is the spacing of rebar (for axisymmetric elements $S_r = (x_1/x_1^0)S_r^0$, where x_1^0 is the radius where the spacing S_r^0 is given);
W_N	is the Gauss weight associated with the integration point along the (r) line;
$\mathbf{x} = \mathbf{x}(g_i)$	is position; and

$$\frac{\partial \mathbf{x}}{\partial r} = \frac{\partial \mathbf{x}}{\partial g_i}\frac{\partial g_i}{\partial r}.$$

Strain is

$$\varepsilon = \frac{1}{2}\ln\left(\frac{dl^2}{dl_o^2}\right),$$

where dl and dl_o measure length along the rebar in the current and initial configurations, respectively.

For the deformations allowed in these elements,

$$\left(\frac{dl}{dl_o}\right)^2 = \cos^2\alpha\lambda_r^2 + \sin^2\alpha\lambda_t^2,$$

where α is the orientation of the rebar from the plane of the model,

$$\lambda_r^2 = \frac{\partial \mathbf{x}}{\partial r}\cdot\frac{\partial \mathbf{x}}{\partial r}\Big/\frac{\partial \mathbf{x}_o}{\partial r}\cdot\frac{\partial \mathbf{x}_o}{\partial r}$$

is the squared stretch ratio in the r-direction, and λ_t is the stretch ratio in the thickness direction:

$\lambda_t = 1$	for plane stress or plane strain;
$\lambda_t = t/t_o$	for generalized plane strain, where t is given in "Generalized plane strain elements," Section 3.2.7; and
$\lambda_t = x_1/x_{1_o}$	for axisymmetric elements.

From these results the first variation of strain is

$$\delta\varepsilon = \left(\frac{dl_o}{dl}\right)^2 (\cos^2\alpha \frac{\partial \mathbf{x}}{\partial r} \cdot \frac{\partial \delta\mathbf{x}}{\partial r} / \frac{\partial \mathbf{x}_o}{\partial r} \cdot \frac{\partial \mathbf{x}_o}{\partial r} + \delta p_t),$$

where

$\delta p_t = 0$ for plane stress and plane strain,
$\delta p_t = \sin^2\alpha t\, \delta t / t_o^2$ for generalized plane strain, and
$\delta p_t = \sin^2\alpha x_1 \delta x_1 / x_{1_o}^2$ for axisymmetric cases.

The second variation of strain is then

$$\begin{aligned} d\delta\varepsilon = -2\left(\frac{dl_o}{dl}\right)^2 & \left(\cos^2\alpha \frac{\partial \mathbf{x}}{\partial r} \cdot \frac{\partial \delta\mathbf{x}}{\partial r} / \frac{\partial \mathbf{x}_o}{\partial r} \cdot \frac{\partial \mathbf{x}_o}{\partial r} + \delta p_t\right) \\ & \cdot \left(\cos^2\alpha \frac{\partial \mathbf{x}}{\partial r} \cdot \frac{\partial d\mathbf{x}}{\partial r} / \frac{\partial \mathbf{x}_o}{\partial r} \cdot \frac{\partial \mathbf{x}_o}{\partial r} + dp_t\right) \\ & + \left(\frac{dl_o}{dl}\right)^2 \left(\cos^2\alpha \frac{\partial d\mathbf{x}}{\partial r} \cdot \frac{\partial \delta\mathbf{x}}{\partial r} / \frac{\partial \mathbf{x}_o}{\partial r} \cdot \frac{\partial \mathbf{x}_o}{\partial r} + d\delta p_t\right). \end{aligned}$$

Reference

- "Defining rebar as an element property," Section 2.2.4 of the Abaqus Analysis User's Guide

3.7.2 REBAR MODELING IN THREE DIMENSIONS

Products: Abaqus/Standard Abaqus/Explicit

Let $g_i, i = 1, 2, 3$, be the isoparametric coordinates of the basic finite element in which the rebar are placed. Let $r_\alpha, \alpha = 1, 2$, be isoparametric coordinates on the surface of reinforcement, with $-1 \leq r_\alpha \leq 1$. Let t be a material coordinate along the rebar direction. See Figure 3.7.2–1.

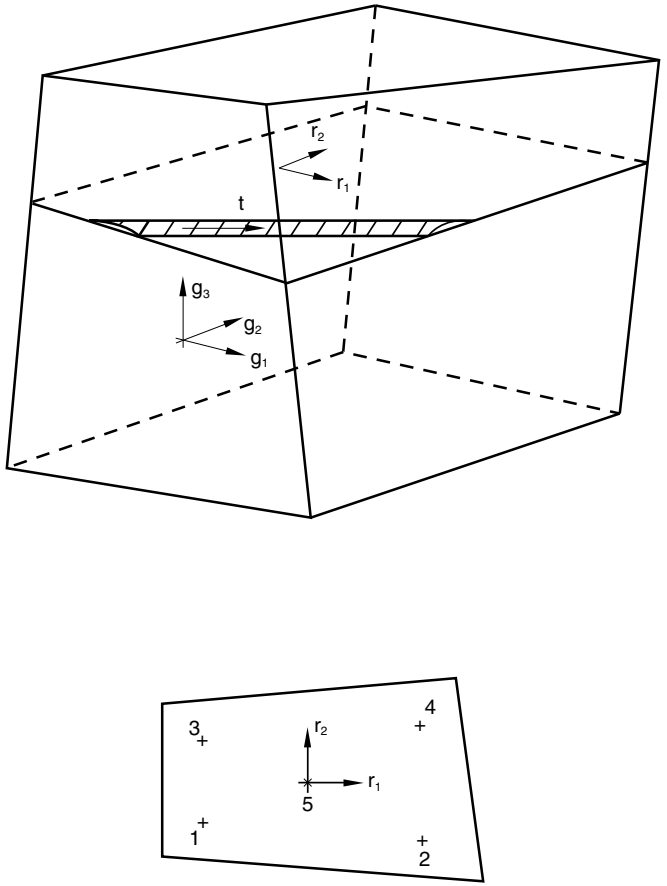


Figure 3.7.2–1 Rebar in a solid, three-dimensional element.

The rebar is integrated using 2×2 or 1×1 Gauss points, depending on the order of the underlying element. The volume of integration at a Gauss point is

$$\Delta V = \frac{A_r}{S_r}\left|\frac{\partial \mathbf{X}}{\partial r_1} \times \frac{\partial \mathbf{X}}{\partial r_2}\right| W_N,$$

where A_r is the cross-sectional area of each rebar, S_r is the rebar spacing, W_N is the Gauss weighting associated with the integration point, $\mathbf{X}$ is the position of the Gauss point, and

$$\frac{\partial \mathbf{X}}{\partial r_\alpha} = \frac{\partial \mathbf{X}}{\partial g_i}\frac{\partial g_i}{\partial r_\alpha}.$$

In these expressions all quantities are taken in the reference configuration, and so Abaqus ignores changes in rebar cross-sectional area due to straining of the rebar and changes in the rebar spacing due to straining of the finite element in which the rebar is placed.

The strain in the rebar is

$$\varepsilon = \frac{1}{2}\ln\left(\frac{g}{G}\right),$$

where

$$g = \frac{\partial \mathbf{x}}{\partial t}\cdot\frac{\partial \mathbf{x}}{\partial t} \qquad \text{and} \qquad \frac{\partial \mathbf{x}}{\partial t} = \frac{\partial \mathbf{x}}{\partial r_i}\frac{\partial r_i}{\partial t},$$

and G is the value of g in the original configuration.

For convenience we define s, a material coordinate that is distance measuring along the rebar in the current configuration:

$$ds = \sqrt{g}\, dt.$$

The first variation of strain is

$$\delta\varepsilon = \frac{\partial \mathbf{x}}{\partial s}\cdot\frac{\partial \delta\mathbf{u}}{\partial s},$$

and the second variation of strain is

$$d\delta\varepsilon = \frac{\partial \delta\mathbf{u}}{\partial s}\cdot\frac{\partial d\mathbf{u}}{\partial s} - 2\frac{\partial \delta\mathbf{u}}{\partial s}\cdot\frac{\partial \mathbf{x}}{\partial s}\frac{\partial \mathbf{x}}{\partial s}\cdot\frac{\partial d\mathbf{u}}{\partial s}.$$

Reference

- "Defining rebar as an element property," Section 2.2.4 of the Abaqus Analysis User's Guide

3.7.3 REBAR MODELING IN SHELL, MEMBRANE, AND SURFACE ELEMENTS

Products: Abaqus/Standard Abaqus/Explicit

The definition of rebar in shell, membrane, or surface elements is based on three geometric properties: the cross-sectional area of each individual rebar, the spacing between the rebars, and the orientation of the rebar with respect to the local coordinate system of the element. For shell elements the rebar definition also requires the distance from the midsurface to the rebar. In Abaqus an equivalent "smeared" orthotropic layer is created based on these geometric properties and the elastic modulus of the rebar material. The equivalent rebar layer lies parallel to the midsurface of the element. For membrane and surface elements this layer coincides with the plane of the element, and for shell elements this layer can be offset by an amount up to half of the shell's thickness. In geometrically linear analyses the geometric properties of the equivalent rebar layer remain constant. However, in geometrically nonlinear analyses each of these properties can change as a result of finite-strain effects.

The user has many options for defining which direction the rebar acts in the element. In each case an angle Θ is determined between the reinforcement and one of the element's isoparametric coordinate directions (selected by the user), measured positive with the axis of rotation along the normal to the element. Let the unit vector $\mathbf{T}$ define the rebar direction at a point in the element. The isoparametric directions are given by the tangent vectors $\mathbf{A}_\alpha$ defined

$$\mathbf{A}_\alpha = \frac{\partial \mathbf{X}}{\partial \xi^\alpha} = \frac{\partial N^A}{\partial \xi^\alpha}\mathbf{X}_A \,,$$

where $\mathbf{X}$ is the reference midsurface position, ξ^α are the isoparametric coordinate functions ($\alpha = 1$ or 2), N^A are the element's shape functions, and $\mathbf{X}_A$ are the element's reference nodal positions.

The reference or initial rebar angle, Θ, is calculated from the inner product between the rebar unit vector, $\mathbf{T}$, and the isoparametric direction, $\mathbf{A}_\alpha$, where α is specified by the user.

$$\Theta = \cos^{-1}\left\{\frac{< \mathbf{T}, \mathbf{A}_\alpha >}{\|\mathbf{A}_\alpha\|}\right\}, \quad \text{no sum on } \alpha.$$

Both the rebar direction, $\mathbf{T}$, and the user-selected isoparametric direction, $\mathbf{A}_\alpha$, lie in a tangent plane parallel to the midsurface. The in-plane unit vector perpendicular to the rebar direction, $\mathbf{P}$, is defined by rotating $\mathbf{T}$ through 90° around the normal to the midsurface, $\mathbf{N}$. The normal direction, $\mathbf{N}$, is found as

$$\mathbf{N} = \frac{\mathbf{A}_1 \times \mathbf{A}_2}{\|\mathbf{A}_1 \times \mathbf{A}_2\|}.$$

As the rebar-reinforced element deforms, the rebars change in length and spacing. The smeared rebar layer assumption implies that the deformation of the rebar layer is determined from the deformation gradient $\mathbf{F}$ of the underlying element. Following from this assumption, the rebar stretch λ_r is

$$\lambda_r = \|\mathbf{F} \cdot \mathbf{T}\| = \|\mathbf{t}\| \,,$$

where $\mathbf{t} = \mathbf{F} \cdot \mathbf{T}$ is the deformed rebar material fiber. Since the deformation gradient maps material lines that are etched into the reference body into the deformed configuration, the length change of these material lines defines the stretch. The rebar logarithmic strain, ε_r, is

$$\varepsilon_r = \ln \lambda_r = \ln \|\mathbf{t}\| \, .$$

The rebar spacing stretch λ_p is the stretch in the plane of the rebar in the direction perpendicular to the rebar. To determine the spacing stretch λ_p, use the fact that the unit normal, $\mathbf{p}$, perpendicular to the deformed rebar direction, $\mathbf{t}$, in the plane of the rebar is

$$\mathbf{p} = \frac{\mathbf{F}^{-T} \cdot \mathbf{P}}{\|\mathbf{F}^{-T} \cdot \mathbf{P}\|} \, .$$

It is easily verified that $\mathbf{p}$ is a unit vector. To see that it is perpendicular to $\mathbf{t}$, take the inner product:

$$< \mathbf{p}, \mathbf{t} >= \frac{1}{\|\mathbf{F}^{-T} \cdot \mathbf{P}\|} < \mathbf{F}^{-T} \cdot \mathbf{P}, \mathbf{F} \cdot \mathbf{T} >= \frac{1}{\|\mathbf{F}^{-T} \cdot \mathbf{P}\|} < \mathbf{P}, \mathbf{T} >= 0 \, .$$

The spacing stretch, λ_p, can be defined as the component along $\mathbf{p}$ of the deformation of the direction $\mathbf{P}$ perpendicular to the reference rebar direction. Since the deformation of $\mathbf{P}$ is $\mathbf{F} \cdot \mathbf{P}$, the spacing stretch follows from

$$\lambda_p =< \mathbf{p}, \mathbf{F} \cdot \mathbf{P} >= \frac{1}{\|\mathbf{F}^{-T} \cdot \mathbf{P}\|} < \mathbf{F}^{-T} \cdot \mathbf{P}, \mathbf{F} \cdot \mathbf{P} >= \frac{< \mathbf{P}, \mathbf{P} >}{\|\mathbf{F}^{-T} \cdot \mathbf{P}\|} \, .$$

Since $\mathbf{P}$ is a unit vector,

$$\lambda_p = \frac{1}{\|\mathbf{F}^{-T} \cdot \mathbf{P}\|} \, .$$

The final angle θ that the rebar direction makes with respect to the user-selected isoparametric direction is

$$\theta = \cos^{-1} \left\{ \frac{< \mathbf{t}, \mathbf{a}_\alpha >}{\|\mathbf{t}\| \|\mathbf{a}_\alpha\|} \right\} \, , \quad \text{no sum on } \alpha.$$

The rebar rotation or change in rebar angle, $\Delta\theta$, is the difference between the final angle and the original angle:

$$\Delta\theta = \theta - \Theta \, .$$

Abaqus reports the current angle, θ, and the change in the rebar angle, $\Delta\theta$, for each rebar definition at each integration location of the element.

The equivalent thickness of the smeared layer is equal to the area of the rebar divided by the rebar's spacing; Abaqus assumes that the volume of the rebar remains constant throughout the analysis. This

assumption implies that the area and spacing of the rebar may change as a result of finite-strain effects. The rebar's area and spacing in the deformed configuration are defined as follows:

$$A_r = \frac{A_r^0}{\lambda_r} \quad \text{and} \quad S_r = S_r^0 \lambda_p \,,$$

where

$$\begin{aligned} A_r^0 &= \text{ original rebar cross-sectional area and} \\ S_r^0 &= \text{ original spacing of rebar.} \end{aligned}$$

In shell elements the rebar layer can be defined initially at a distance above or below the midsurface. In shell elements that permit finite strain, the shell's thickness can change as a function of the in-plane deformation. In Abaqus/Standard the thickness change is defined by the section Poisson's ratio, which can be specified by the user. In Abaqus/Explicit this behavior is based on the actual material properties through the shell's thickness. To account for the change in the shell's thickness, the rebar layer's distance from the midsurface is scaled by the thickness stretch.

Reference

- "Defining reinforcement," Section 2.2.3 of the Abaqus Analysis User's Guide

3.8 Hydrostatic fluid calculations

- “Hydrostatic fluid calculations,” Section 3.8.1

3.8.1 HYDROSTATIC FLUID CALCULATIONS

Products: Abaqus/Standard Abaqus/Explicit

Abaqus provides a capability that can be used to represent fluid-filled cavities under hydrostatic conditions. The capability provides the coupling between the deformation of the fluid-filled structure and the pressure exerted by the contained fluid on the boundary of the cavity. In Abaqus/Explicit the fluid must be compressible and the pressure is calculated from the cavity volume. In Abaqus/Standard the fluid inside the cavity can be compressible or incompressible, with the fluid volume given as a function of the fluid pressure, ***p***; the fluid temperature, θ; and the fluid mass, ***m***, in the cavity:

$$\overline{V} = \overline{V}(p, \theta, m).$$

We refer to the incompressible case as a "hydraulic" fluid and to the compressible case as a "pneumatic" fluid. The volume, $\overline{V}$, derived from the fluid pressure and temperature should equal the actual volume, ***V***, of the cavity. In Abaqus/Standard this is achieved by augmenting the virtual work expression for the structure with the constraint equation

$$V - \overline{V} = 0$$

and the virtual work contribution due to the cavity pressure:

$$\delta\Pi^* = \delta\Pi - p\delta V - \delta p(V - \overline{V}),$$

where $\delta\Pi^*$ is the augmented virtual work expression and $\delta\Pi$ is the virtual work expression for the structure without the cavity. The negative signs imply that an increase in the cavity volume releases energy from the fluid. This represents a mixed formulation in which the structural displacements and fluid pressure are primary variables. The rate of the augmented virtual work expression is obtained as

$$\begin{aligned} d\delta\Pi^* &= d\delta\Pi - pd\delta V - dp\delta V - (dV - d\overline{V})\delta p \\ &= d\delta\Pi - pd\delta V - dp\delta V - dV\delta p + \frac{d\overline{V}}{dp}dp\delta p. \end{aligned}$$

Here, $-pd\delta V$ represents the pressure load stiffness, and $d\overline{V}/dp$ is the volume-pressure compliance of the fluid.

Since the pressure is the same for all surface facets (or elements) in the cavity, the augmented virtual work expression can be written as the sum of the expressions for the individual surface facets:

$$\begin{aligned} \delta\Pi^* &= \delta\Pi - p\sum_e \delta V^e - \delta p\left[\sum_e V^e - \sum_e \overline{V}^e\right] \\ &= \sum_e \left[\delta\Pi^e - p\delta V^e - \delta p(V^e - \overline{V}^e)\right]. \end{aligned}$$

Moreover, since the temperature is the same for all cavity facets, the fluid volume can be calculated for each facet individually:

$$\overline{V}^e = \overline{V}^e(p, \theta, m^e),$$

where m^e is the element mass. In the solution the actual volume of the element may be different from the element volume:

$$V^e - \overline{V}^e \neq 0;$$

however, the total fluid volume will match the volume of the cavity.

Hydraulic fluid with thermal expansion

In Abaqus/Standard the fluid is incompressible by default and the fluid volume, $\overline{V}$, is dependent upon temperature but independent of the fluid pressure:

$$\overline{V} = \overline{V}(\theta, m), \qquad \frac{d\overline{V}}{dp} = 0.$$

If compressibility is introduced, the fluid volume depends upon both the temperature and pressure:

$$\overline{V} = \overline{V}(p, \theta, m), \qquad \frac{d\overline{V}}{dp} = -\frac{m}{\rho_R K},$$

where $\boldsymbol{K}$ is the fluid bulk modulus and ρ_R is the reference fluid density at zero pressure and the initial temperature.

The total fluid mass in the cavity is the sum of the fluid masses of the elements making up the cavity:

$$m = \sum_e m^e.$$

The mass of a fluid element in the cavity, m^e, is calculated from the initial fluid density, $\rho(p_I, \theta_I)$, and the initial element volume, V_I^e:

$$m^e = \rho(p_I, \theta_I)\, V_I^e,$$

where p_I is the initial fluid pressure and θ_I is the initial temperature. The initial fluid density follows from the user-defined reference density, ρ_R:

$$\rho(p_I, \theta_I) = \frac{\rho_R}{(1 - p_I/K)}.$$

The fluid density at the current pressure and temperature, $\rho(p, \theta)$, is obtained as

$$\rho(p, \theta) = \rho_R \big[1 + 3\alpha(\theta)(\theta - \theta_0) - 3\alpha(\theta_I)(\theta_I - \theta_0) - p/K\big]^{-1},$$

where θ_0 is the reference temperature for the coefficient of thermal expansion and $\alpha(\theta)$ is the mean (secant) coefficient of thermal expansion, and it is assumed that $|\rho/\rho_R - 1| << 1$.

Thus, the fluid volume at the current pressure and temperature is

$$\overline{V}(p,\theta) = m/\rho(p,\theta).$$

This volume can be calculated on an element by element basis:

$$\overline{V}(p,\theta) = \sum_e \overline{V}^e(p,\theta) = \sum_e m^e/\rho(p,\theta) = m/\rho(p,\theta).$$

Fluid can be added to or removed from the cavity. The amount of fluid added is given as the change in (fluid) mass Δm. Consequently, the change of the fluid volume at the current cavity temperature is

$$\Delta\overline{V}(p,\theta) = \Delta m/\rho(p,\theta).$$

Ideal gas

In this case the fluid is compressible, and the volume is a function of the pressure and the temperature in the cavity:

$$\overline{V} = \overline{V}(p,\theta,m),$$

where, as before, the total fluid mass in the cavity is the sum of the masses of the elements in the cavity. The fluid is assumed to behave like an ideal gas; hence, the density of the fluid in the cavity can be calculated as

$$\rho(p,\theta) = \frac{p + p_A}{R(\theta - \theta_A)},$$

where R is the gas constant, θ_A is the temperature at absolute zero, and p_A is the ambient pressure. The current fluid volume can again be calculated on an element by element basis:

$$\overline{V}(p,\theta) = \sum_e \overline{V}^e(p,\theta) = \sum_e m^e/\rho(p,\theta) = m/\rho(p,\theta).$$

The corresponding volume-pressure compliance is

$$\frac{d\overline{V}}{dp} = -\frac{m}{\rho^2}\frac{d\rho}{dp} = -\frac{mR(\theta - \theta_A)}{(p + p_A)^2}.$$

Fluid can be added to or removed from the cavity. The amount of fluid added is again given as the change in (fluid) mass. Consequently, the change of the fluid volume at the current cavity temperature is

$$\Delta\overline{V}(p,\theta) = \Delta m/\rho(p,\theta).$$

Volume calculation

The hydrostatic fluid elements appear as surface elements that cover the cavity boundary, but they are actually volume elements when the cavity reference node is accounted for. Figure 3.8.1–1 depicts the 4-node hydrostatic fluid volume element, F3D4. The dashed lines indicate that the element is actually pyramidal in shape.

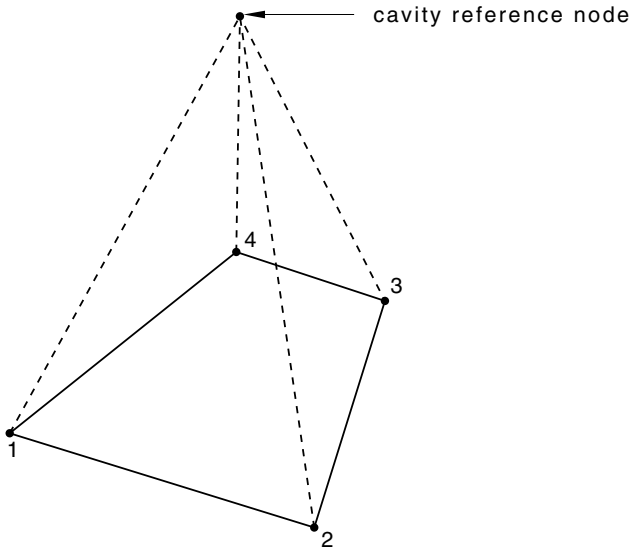


Figure 3.8.1–1 F3D4 hydrostatic fluid element.

The volume, V^e, of each element must be calculated. The coordinates of any point on the base of the pyramid element can be found by

$$\mathbf{x} = \sum_N N^N(g,h)\mathbf{x}^N,$$

where N^N are the interpolation functions for the base of the pyramid ("Solid isoparametric quadrilaterals and hexahedra," Section 3.2.4), expressed in terms of parametric coordinates g and h; $\mathbf{x}^N$ are the nodal coordinates; and the summation extends over all nodes on the base. For three-dimensional elements the Jacobian on the surface is calculated as

$$\frac{\partial \mathbf{x}}{\partial g} = \sum_N \frac{\partial N^N}{\partial g}\mathbf{x}^N, \qquad \frac{\partial \mathbf{x}}{\partial h} = \sum_N \frac{\partial N^N}{\partial h}\mathbf{x}^N.$$

The normal to the element face, $\mathbf{n}$, multiplied by an infinitesimal area, dA, of the element face is, hence, obtained as

$$\mathbf{n}\,dA = \left(\frac{\partial \mathbf{x}}{\partial g} \times \frac{\partial \mathbf{x}}{\partial h}\right) dg\,dh.$$

The infinitesimal volume, dV, associated with this infinitesimal area is

$$dV = \frac{1}{3}\left(\mathbf{x}_R - \mathbf{x}\right) \cdot \mathbf{n}\,dA,$$

where $\mathbf{x}_R$ is the position of the cavity reference node. The volume of the element, V^e, is then obtained by integration. For a quadrilateral base this yields

$$V^e = \int_{V^e} dV = \int_{-1}^{+1}\int_{-1}^{+1} \frac{1}{3}\left(\mathbf{x}_R - \mathbf{x}\right) \cdot \left(\frac{\partial \mathbf{x}}{\partial g} \times \frac{\partial \mathbf{x}}{\partial h}\right) dg\,dh.$$

The integration boundaries will be different for a triangular base. Introducing the relative position, $\overline{\mathbf{x}} = \mathbf{x} - \mathbf{x}_R$, this becomes

$$V^e = \int_{-1}^{+1}\int_{-1}^{+1} -\frac{1}{3}\,\overline{\mathbf{x}} \cdot \left(\frac{\partial \overline{\mathbf{x}}}{\partial g} \times \frac{\partial \overline{\mathbf{x}}}{\partial h}\right) dg\,dh.$$

The variation in the element volume is readily obtained as

$$\delta V^e = \int_{-1}^{+1}\int_{-1}^{+1} -\frac{1}{3}\left[\delta\overline{\mathbf{x}} \cdot \left(\frac{\partial \overline{\mathbf{x}}}{\partial g} \times \frac{\partial \overline{\mathbf{x}}}{\partial h}\right) + \overline{\mathbf{x}} \cdot \left(\frac{\partial \delta\overline{\mathbf{x}}}{\partial g} \times \frac{\partial \overline{\mathbf{x}}}{\partial h} + \frac{\partial \overline{\mathbf{x}}}{\partial g} \times \frac{\partial \delta\overline{\mathbf{x}}}{\partial h}\right)\right] dg\,dh.$$

This expression includes contributions to the volume change due to variation in the "sides" of the pyramid element. Hence, the equivalent forces will also include the effect of the pressure on these sides. The pressure on the sides will be balanced by the pressure on the side of the adjacent pyramid element; hence, we only need to calculate the contribution from the pressure on the "base" of the pyramid. This can be done by separating the contributions by using partial integration, which yields

$$\delta V_b^e = \int_{-1}^{+1}\int_{-1}^{+1} -\delta\overline{\mathbf{x}} \cdot \left(\frac{\partial \overline{\mathbf{x}}}{\partial g} \times \frac{\partial \overline{\mathbf{x}}}{\partial h}\right) dg\,dh - \int_{-1}^{+1} \left[\overline{\mathbf{x}} \cdot \left(\delta\overline{\mathbf{x}} \times \frac{\partial \overline{\mathbf{x}}}{\partial h} + \overline{\mathbf{x}} \times \frac{\partial \delta\overline{\mathbf{x}}}{\partial h}\right)\right]_{g=-1}^{g=1} dh$$
$$- \int_{-1}^{+1} \left[\overline{\mathbf{x}} \cdot \left(\frac{\partial \delta\overline{\mathbf{x}}}{\partial g} \times \overline{\mathbf{x}} + \frac{\partial \overline{\mathbf{x}}}{\partial g} \times \delta\overline{\mathbf{x}}\right)\right]_{h=-1}^{h=1} dg.$$

The last two integrals represent the contributions on the sides of the pyramid; hence, the resulting expression is

$$\delta V_b^e = \int_{-1}^{+1}\int_{-1}^{+1} -\delta\overline{\mathbf{x}} \cdot \left(\frac{\partial \overline{\mathbf{x}}}{\partial g} \times \frac{\partial \overline{\mathbf{x}}}{\partial h}\right) dg\,dh.$$

This equation can readily be obtained directly (see "Pressure load stiffness," Section 6.5.1).

The second variation of the expression for the volume is

$$d\delta V_b^e = \int_{-1}^{+1}\int_{-1}^{+1} -\frac{1}{3}\left[\delta\overline{\mathbf{x}}\cdot\left(\frac{\partial d\overline{\mathbf{x}}}{\partial g}\times\frac{\partial\overline{\mathbf{x}}}{\partial h}+\frac{\partial\overline{\mathbf{x}}}{\partial g}\times\frac{\partial d\overline{\mathbf{x}}}{\partial h}\right)+d\overline{\mathbf{x}}\cdot\left(\frac{\partial\delta\overline{\mathbf{x}}}{\partial g}\times\frac{\partial\overline{\mathbf{x}}}{\partial h}+\frac{\partial\overline{\mathbf{x}}}{\partial g}\times\frac{\partial\delta\overline{\mathbf{x}}}{\partial h}\right)+ \overline{\mathbf{x}}\cdot\left(\frac{\partial\delta\overline{\mathbf{x}}}{\partial g}\times\frac{\partial d\overline{\mathbf{x}}}{\partial h}+\frac{\partial d\overline{\mathbf{x}}}{\partial g}\times\frac{\partial\delta\overline{\mathbf{x}}}{\partial h}\right)\right] dg\,dh.$$

Similar expressions can be obtained for planar and axisymmetric fluid elements. For a planar element the volume is readily obtained as

$$V^e = -\frac{1}{2}\mathbf{e}_3\cdot\int_{-1}^{+1}\overline{\mathbf{x}}\times\frac{\partial\overline{\mathbf{x}}}{\partial g}\,dg,$$

where $\mathbf{e}_3$ is the unit vector in the direction perpendicular to the plane. Similarly, for axisymmetric elements

$$V^e = -\frac{2}{3}\pi\,\mathbf{e}_\theta\cdot\int_{-1}^{+1}\left(\overline{\mathbf{x}}\times\frac{\partial\overline{\mathbf{x}}}{\partial g}\right)(\overline{\mathbf{x}}\cdot\mathbf{e}_r)\,dg.$$

The first and second variations are obtained in the same way as for three-dimensional elements.

The integrations can be carried out analytically. For example, for element type F3D4 the above expressions yield, after some manipulation,

$$V^e = \frac{1}{12}\left[(\overline{\mathbf{x}}^3-\overline{\mathbf{x}}^1)\cdot(\overline{\mathbf{x}}^2\times\overline{\mathbf{x}}^4)+(\overline{\mathbf{x}}^4-\overline{\mathbf{x}}^2)\cdot(\overline{\mathbf{x}}^3\times\overline{\mathbf{x}}^1)\right],$$

where $\overline{\mathbf{x}}^N$, $N=1,4$ denote the relative nodal coordinates. The first variation (involving the base only) is equal to

$$\delta V_b^e = \frac{1}{12}\Big[\delta\overline{\mathbf{x}}^1\cdot[(\overline{\mathbf{x}}^2-\overline{\mathbf{x}}^4)\times(2\overline{\mathbf{x}}^1-\overline{\mathbf{x}}^2-\overline{\mathbf{x}}^3)]+\delta\overline{\mathbf{x}}^2\cdot[(\overline{\mathbf{x}}^3-\overline{\mathbf{x}}^1)\times(2\overline{\mathbf{x}}^2-\overline{\mathbf{x}}^3-\overline{\mathbf{x}}^4)]+ \delta\overline{\mathbf{x}}^3\cdot[(\overline{\mathbf{x}}^4-\overline{\mathbf{x}}^2)\times(2\overline{\mathbf{x}}^3-\overline{\mathbf{x}}^4-\overline{\mathbf{x}}^1)]+\delta\overline{\mathbf{x}}^4\cdot[(\overline{\mathbf{x}}^1-\overline{\mathbf{x}}^3)\times(2\overline{\mathbf{x}}^4-\overline{\mathbf{x}}^1-\overline{\mathbf{x}}^2)]\Big],$$

and the second variation of the volume is

$$d\delta V^e = \frac{1}{12}\Big[(\delta\overline{\mathbf{x}}^2\times d\overline{\mathbf{x}}^1-\delta\overline{\mathbf{x}}^1\times d\overline{\mathbf{x}}^2)\cdot(\overline{\mathbf{x}}^3+\overline{\mathbf{x}}^4)+(\delta\overline{\mathbf{x}}^3\times d\overline{\mathbf{x}}^2-\delta\overline{\mathbf{x}}^2\times d\overline{\mathbf{x}}^3)\cdot(\overline{\mathbf{x}}^4+\overline{\mathbf{x}}^1)+ (\delta\overline{\mathbf{x}}^4\times d\overline{\mathbf{x}}^3-\delta\overline{\mathbf{x}}^3\times d\overline{\mathbf{x}}^4)\cdot(\overline{\mathbf{x}}^1+\overline{\mathbf{x}}^2)+(\delta\overline{\mathbf{x}}^1\times d\overline{\mathbf{x}}^4-\delta\overline{\mathbf{x}}^4\times d\overline{\mathbf{x}}^1)\cdot(\overline{\mathbf{x}}^2+\overline{\mathbf{x}}^3)+ (\delta\overline{\mathbf{x}}^1\times d\overline{\mathbf{x}}^3-\delta\overline{\mathbf{x}}^3\times d\overline{\mathbf{x}}^1)\cdot(\overline{\mathbf{x}}^2-\overline{\mathbf{x}}^4)+(\delta\overline{\mathbf{x}}^2\times d\overline{\mathbf{x}}^4-\delta\overline{\mathbf{x}}^4\times d\overline{\mathbf{x}}^2)\cdot(\overline{\mathbf{x}}^3-\overline{\mathbf{x}}^1)\Big].$$

Fluid exchange

Abaqus provides a capability that can be used to model fluid flow between two cavities or between a cavity and the outside world. This is typically used when the fluid has to flow through a narrow orifice.

It is assumed that the mass flow rate, q, is a homogeneous function of the pressure difference, $\Delta p = p_1 - p_2$. In addition, it is assumed that the flow rate may depend on the average temperature—$\overline{\theta} = (\theta_1 + \theta_2)/2$—and, for a compressible fluid, on the average pressure—$\overline{p} = (p_1 + p_2)/2$:

$$q = q(\Delta p, \overline{p}, \overline{\theta}). \tag{3.8.1–1}$$

The flow rate needs to be integrated over a finite increment. We assume that the dependence on the average pressure, $\overline{p}$, is weak. Hence, we use a semi-implicit method: we use Δp at the end of the increment and $\overline{p}^0$ at the start of the increment. For the temperature we choose $\overline{\theta}^m$ to be the average of $\overline{\theta}$ at the start and end of the increment, because this is likely to be the most accurate. Hence, we obtain the mass flow as:

$$\Delta m = q(\Delta p, \overline{p}^0, \overline{\theta}^m)\Delta t.$$

This mass change needs to be converted to a volume change in each cavity. It is assumed that the fluid in both cavities is the same, but the pressures (and possibly the temperatures) may be different. With use of the pressure and temperature-dependent density $\rho_i(p, \theta)$ for each cavity i, the relations become

$$\Delta \overline{V}_1 = -\Delta m/\rho_1, \qquad \Delta \overline{V}_2 = \Delta m/\rho_2,$$

and

$$\begin{aligned} \frac{\partial \Delta \overline{V}_1}{\partial p_1} &= -\frac{1}{\rho_1}\frac{\partial \Delta m}{\partial \Delta p} + \frac{\Delta m}{\rho_1^2}\frac{\partial \rho_1}{\partial p_1}, & \frac{\partial \Delta \overline{V}_1}{\partial p_2} &= \frac{1}{\rho_1}\frac{\partial \Delta m}{\partial \Delta p}, \\ \frac{\partial \Delta \overline{V}_2}{\partial p_1} &= \frac{1}{\rho_2}\frac{\partial \Delta m}{\partial \Delta p}, & \frac{\partial \Delta \overline{V}_2}{\partial p_2} &= -\frac{1}{\rho_2}\frac{\partial \Delta m}{\partial \Delta p} - \frac{\Delta m}{\rho_2^2}\frac{\partial \rho_2}{\partial p_2}. \end{aligned} \tag{3.8.1–2}$$

Observe that, except for isothermal incompressible fluids, the resulting matrix is nonsymmetric. For an incompressible fluid with thermal expansion, the nonsymmetric contribution is small and the equations can be symmetrized without significant degradation in convergence rate. For an ideal gas, the nonsymmetric terms can have a significant effect if the drop in pressure is large.

Many applications of fluid flow between cavities involve dynamic loading in the form of steady-state vibration; and often in such cases the dissipative losses in the link between the cavities must be modeled to obtain useful results. In most problems of this class the fluids on either side of the fluid link are first pressurized statically. In the implementation in Abaqus/Standard, it is assumed that the vibration amplitude is sufficiently small that the fluid link response in the dynamic phase of the problem can be treated as linear perturbations about the prepressurized state.

For small vibrations about a prepressurized state we linearize Equation 3.8.1–1 to give

$$q = \left.\frac{\partial q(\Delta p, \overline{p}, \overline{\theta})}{\partial \Delta p}\right|_0 \Delta p.$$

The mass flow through the link can, therefore, be derived as follows:

$$\begin{aligned}\Delta m &= \int_{-\infty}^{t} q\,dt = \int_{-\infty}^{t} \left.\frac{\partial q}{\partial \Delta p}\right|_0 \Delta p\,dt \\ &= \int_{-\infty}^{t} \left.\frac{\partial q}{\partial \Delta p}\right|_0 \Delta \bar{p}\exp(i\omega t)dt = -\frac{i}{\omega}\left.\frac{\partial q}{\partial \Delta p}\right|_0 \Delta \bar{p}\exp(i\omega t) \\ &= -\frac{i}{\omega}\left.\frac{\partial q}{\partial \Delta p}\right|_0 \Delta p.\end{aligned}$$

Substituting the above expression for mass flow into Equation 3.8.1–2 and noting that $\Delta m|_0 = 0$ yields

$$\left.\frac{\partial \Delta\overline{V}_1}{\partial p_1}\right|_0 = \frac{i}{\omega\rho_1}\left.\frac{\partial q}{\partial \Delta p}\right|_0, \qquad \left.\frac{\partial \Delta\overline{V}_1}{\partial p_2}\right|_0 = -\frac{i}{\omega\rho_1}\left.\frac{\partial q}{\partial \Delta p}\right|_0,$$
$$\left.\frac{\partial \Delta\overline{V}_2}{\partial p_1}\right|_0 = -\frac{i}{\omega\rho_2}\left.\frac{\partial q}{\partial \Delta p}\right|_0, \qquad \left.\frac{\partial \Delta\overline{V}_2}{\partial p_2}\right|_0 = \frac{i}{\omega\rho_2}\left.\frac{\partial q}{\partial \Delta p}\right|_0.$$

In Abaqus this model is provided only for the direct-solution steady-state dynamic analysis procedure.

Negative eigenvalues

It is possible that negative eigenvalues will be encountered in the solution of certain hydrostatic fluid element problems. With standard elements this can indicate that a bifurcation or buckling load has been exceeded. However, this is not necessarily true with hydrostatic fluid elements; negative eigenvalues can result solely from the numerical implementation.

Consider the simple hydrostatic fluid model depicted in Figure 3.8.1–2. If the fluid is considered incompressible, application of the downward force causes the fluid to compress vertically and expand horizontally, while maintaining the original fluid volume. Thus, the model can be adequately discretized as a three degree of freedom system: the horizontal displacement of the right platen, u; the vertical displacement of the top platen, v; and the fluid pressure, p. The corresponding system of equations in matrix form is

$$\begin{Bmatrix} dF_1 \\ dF_2 \\ -dV \end{Bmatrix} = \begin{bmatrix} k_1 & -p & -(1+v) \\ -p & k_2 & -(1+u) \\ -(1+v) & -(1+u) & 0 \end{bmatrix} \begin{Bmatrix} du \\ dv \\ dp \end{Bmatrix}.$$

The equation solver processes the equations in sequence. Hence, it will process the 2×2 submatrix relating the displacements to the forces prior to processing the constraints. This leads to a

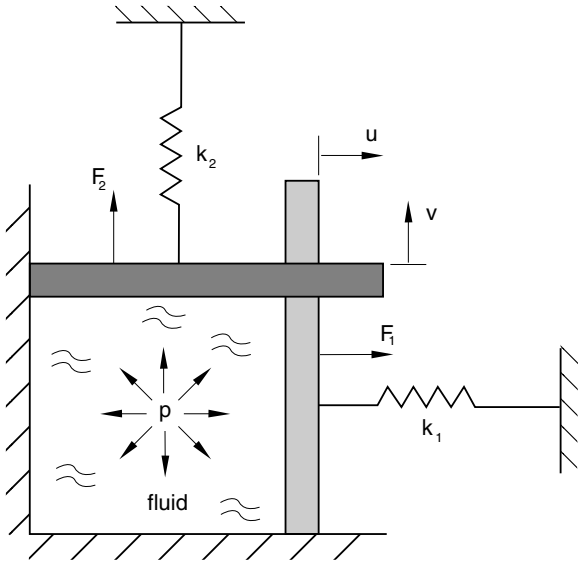


Figure 3.8.1–2 Simple hydrostatic fluid model.

negative eigenvalue if $p^2 > k_1 k_2$. However since the mode associated with the negative eigenvalue is subsequently constrained by the continuity equation, no instability occurs.

Reference

- "Surface-based fluid cavities: overview," Section 11.5.1 of the Abaqus Analysis User's Guide

3.9 Special-purpose elements

- “Elbow elements,” Section 3.9.1
- “Frame elements with lumped plasticity,” Section 3.9.2
- “Buckling strut response for frame elements,” Section 3.9.3
- “Tube support elements,” Section 3.9.4
- “Line spring elements,” Section 3.9.5
- “Flexible joint element,” Section 3.9.6
- “Rotary inertia element,” Section 3.9.7
- “Distributing coupling elements,” Section 3.9.8

3.9.1 ELBOW ELEMENTS

Product: Abaqus/Standard

The design of piping systems relies heavily on the use of elbows (pipe bends) to provide flexibility in response to thermal strains. For this reason the elbows themselves are the critical components in the design, and it is essential to predict their response accurately to qualify a pipeline structurally. This is especially true in accident conditions, where inelastic behavior—plasticity or viscoplasticity—dominates the response. It is well known that piping elbows achieve their flexibility through a shell-type behavior—responding to bending loads with significant ovalization of the pipe cross-section—in contrast to the beam response of straight pipes, where the cross-section does not deform to any significant extent until Brazier buckling occurs. Ovalization gives the elbows elastic flexibility that is 5–20 times that of the same size straight pipe in bending (for typical primary piping loop elbows in nuclear reactors). This flexibility is accompanied by stresses and strains that are typically 3–12 times those of a straight pipe under the same load—with of course—very high strain gradients around the pipe and through its thickness, caused by the ovalization of the cross-section. For the purpose of linear elastic analysis these effects present little difficulty because the flexibility factors and stress intensity factors have been accurately calculated and verified with experiment and are, thus, used directly in response prediction using three-dimensional beam theory (see Dodge and Moore, 1972, for a survey of such work).

For nonlinear analysis or pipes of noncircular section, these solutions are not useful. Two possible approaches suggest themselves for the nonlinear case: nonlinear equivalents to the linear elastic flexibility and strain intensity factors can be sought or detailed analysis based on the shell-type behavior of the pipe bends can be undertaken. The concept of nonlinear equivalents to the linear elastic flexibility and stress intensity factors has been pursued by a number of workers (e.g., Spence, 1972), although the success of such methods does not yet warrant their use in design (Scheller and Mallett, 1979). Detailed inelastic analysis remains, therefore, as the obvious approach to qualifying piping systems that cannot be designed on an elastic basis. The main difficulty with such analysis is its cost in both engineering time and computer time. The structure is a shell exhibiting severe stress gradients, so that the numerical models required to provide detailed analysis of adequate accuracy are large and complicated. Many attempts have been made to provide such models. Hibbitt (1974) described the formulation of a shell-type finite element that uses piecewise cubic interpolation of displacement around the pipe and piecewise constant strain along the pipe. The latter simplification gives rise to undesirable displacement discontinuities along the pipe. Nevertheless, the approach has documented accuracy (Sobel, 1977) and has been used in design evaluation (Chen and Weiner, 1978). Ohtsubo and Watanabe (1976) develop shell elements of “ring” geometry, based on the virtual work statement for small strains of a thin walled torus section. They interpolate the normal and two tangential displacement components of the middle surface of the shell each by a complete Fourier series—truncated after M terms—around the pipe and a piecewise cubic along the pipe. Their model allows both ovalization and warping of the section, thus potentially describing the behavior of the pipe to high accuracy. Their results show that, with such an approach, for typical primary piping elbows four to six Fourier modes are needed around the pipe to obtain a convergent solution, with fewer modes necessary in less flexible elbows. However, their formulation is not well-suited to practical application in multi-elbow pipelines because strains result from rigid body motions

of the elbows. This is due to the use of cylindrical shell theory with polynomial and Fourier interpolation of displacement components in a curvilinear coordinate system.

Takeda et al. (1979) have developed an extension of the Ohtsubo and Watanabe shell element, using both Fourier and piecewise cubic interpolation along the pipe. The element has been applied to some well-documented, nonlinear benchmark problems (e.g., Kwata, 1978) and shown to be accurate but relatively expensive. In Kwata (1978), Takeda et al. compare the element to the single point integration, bilinear interpolation shell element of Hughes et al. (1977) and conclude that the latter element is more economical for given accuracy. To a large extent this conclusion would appear to depend on the architecture of the computer code in which the element is implemented: unless the code architecture allows economic calculation of such elements, they will be very expensive. In Kwata (1978) the simple bilinear shell is shown to need about 40 elements (and hence 40 constitutive points) around the pipe to provide accurate stresses (the paper actually shows 20 elements around the half-pipe in a case of in-plane bending), thus introducing about 240 degrees of freedom per segment; while in Ohtsubo and Wanatabe (1976) six Fourier modes (and hence 12 degrees of freedom for in-plane modeling, 24 degrees of freedom for out-of-plane modeling, per segment) give the same level of accuracy—with again, about 40 constitutive calculation points. With the formulation in Abaqus, we also conclude that 32–48 constitutive calculation stations around the pipe are needed. The contrast in number of degrees of freedom per segment suggests that—in spite of the relative complexity of elements like that of Ohtsubo and Watanabe—given suitable code architecture designed to handle such elements efficiently, elements that take advantage of the particular geometry and behavior of elbows should be significantly cheaper than most simple shell elements such as that of Hughes et al. (1977).

The element of Bathe and Almeida (1980) is a direct finite element implementation of the classical von Karmann approach. Warping of the section is neglected, and in the plane of the cross-section only pure bending of the pipe wall is allowed. This latter restriction precludes the direct application of thermal loads to the element. In addition, it does not allow for the "circumferential membrane force correction" added by Gross to the von Karmann theory (see Dodge and Moore, 1972). As an additional simplification, the gradients of axial and shear strain through the thickness of the pipe wall are ignored. Around the pipe a Fourier series is used for the tangential displacement component of the pipe wall middle surface in the plane of the pipe section. In this element only the even numbered Fourier terms $(\sin 2m\phi,\ \cos 2m\phi,\ m = 1, 2...M)$ are used, as in von Karmann's analysis. This implies that these displacements are symmetric about the crown of the elbow (about $\phi = \pi/2,\ \phi = 3\pi/2$ in Figure 3.9.1–1). This is satisfactory for cases where the pipe radius is small compared to the elbow torus radius but is likely to be inadequate when the pipe radius is of the same order as the torus radius. The interpolation along the pipe is a piecewise Lagrange cubic, in contrast to the Hermite cubic of Takeda et al. (1979), so that considerably more degrees of freedom are introduced per segment. The reason for this choice is not made obvious in Bathe and Almeida (1980). The authors restrict themselves to 3 Fourier modes and 24 integration stations (constitutive calculation points) around the pipe; the element would probably require some extension to allow more modes and constitutive calculation points around the pipe for application to more flexible elbows.

In Abaqus/Standard we build on the experience summarized above and provide a capability for detailed inelastic analysis of pipelines at reasonable cost. The achievement of such complex response prediction in a reasonable amount of computer time requires that the geometric modeling be as efficient as possible—that is, that the model be "tuned" to the characteristics of the specific structural geometry and response. Pipe bends behave as shells; however, they have the simplifying characteristic that the strain gradients along the

pipe are usually mild compared to those around the pipe. The choices of geometric modeling used for the elbow elements in Abaqus use this simplification to minimize the computer costs associated with the nonlinear analysis of pipelines.

Geometric definitions

We assume that the undeformed pipe is of circular or nearly circular section. Let $(\phi,\ S)$ be the material coordinates of a point in the pipe wall middle surface, defined as follows (see Figure 3.9.1–1).

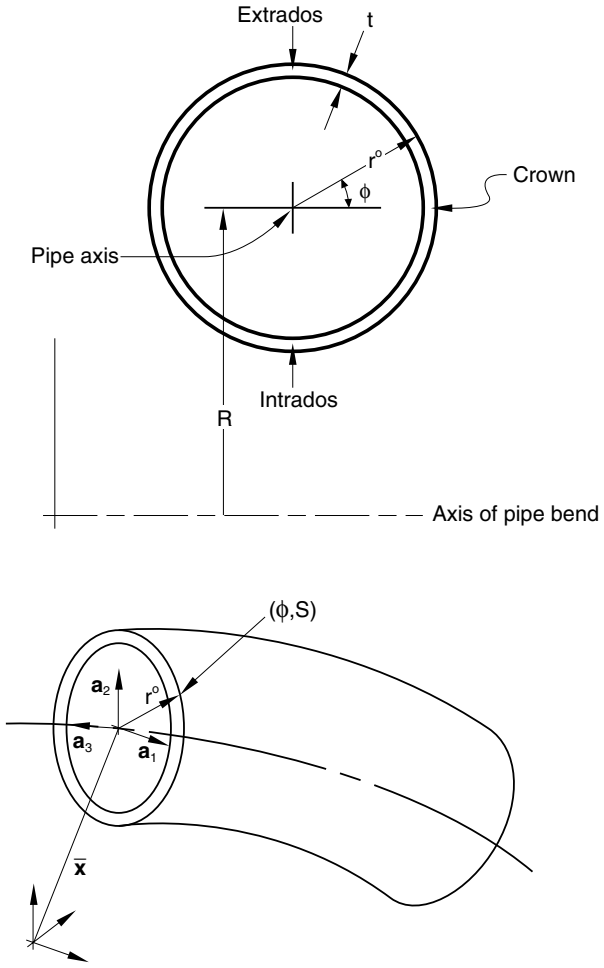


Figure 3.9.1–1 Elbow geometry.

In the undeformed configuration ϕ measures the angular position of the point from the crown of the section such that ϕ increases towards the extrados (on a straight segment the "crown" and "extrados" are arbitrarily defined as fixed positions around the pipe section) and S measures the distance along the pipe axis from some arbitrary origin. Let $\mathbf{x}(\phi, S)$ be the current and $\mathbf{X}(\phi, S)$ the reference positions of a point. Let $\overline{\mathbf{x}}(S)$ and $\overline{\mathbf{x}}(S)$ be the current and reference positions of a point on the pipe axis, and define $\mathbf{y} = \mathbf{x} - \overline{\mathbf{x}}$ as the offset of a midwall point from the pipe axis. Let $\mathbf{a} = [\mathbf{a}_1,\ \mathbf{a}_2\ , \mathbf{a}_3]$ be a right-handed director set of orthogonal, unit vectors, with $\mathbf{a}_3$ approximately tangent to the pipe axis. $\mathbf{A} = [\mathbf{A}_1,\ \mathbf{A}_2,\ \mathbf{A}_3]$ are the same directors in the reference configuration: we choose $\mathbf{A}$ so that the point $\phi = 0$ (the crown of an elbow section) lies on $\mathbf{A}_1$ and $\mathbf{A}_2$ points to the extrados ($\phi = \pi/2$). For simplicity we rewrite

$$\mathbf{y} = \mathbf{y}^i \mathbf{a}_i = (r^o + u^r)\mathbf{r} + u^t \mathbf{t} + y^3 \mathbf{a}_3,$$

where

$$\mathbf{r} = \mathbf{a}_1 \cos\phi + \mathbf{a}_2 \sin\phi$$

and

$$\mathbf{t} = -\mathbf{a}_1 \sin\phi + \mathbf{a}_2 \cos\phi.$$

Finally, we define a unit vector, $\mathbf{n}$, which will be made approximately outwardly normal to the pipe wall middle surface, as follows. Let

$$\mathbf{t}_t = \frac{\partial \mathbf{x}}{\partial \phi} \left(\frac{\partial \mathbf{x}}{\partial \phi} \cdot \frac{\partial \mathbf{x}}{\partial \phi} \right)^{-\frac{1}{2}}$$

be a unit vector pointing around the pipe section S =constant, and let

$$\mathbf{t}_n = \mathbf{a}_3 - \mathbf{a}_3 \cdot \mathbf{t}_t \, \mathbf{t}_t.$$

Then write

$$\mathbf{n} = \overline{\mathbf{n}} (\overline{\mathbf{n}} \cdot \overline{\mathbf{n}})^{-\frac{1}{2}},$$

where

$$\overline{\mathbf{n}} = \frac{\partial \mathbf{x}}{\partial \phi} \times \mathbf{a}_3 + \gamma^3 \mathbf{t} n,$$

with

$$\gamma^3 = \gamma^3(\phi,\ S).$$

This definition ensures that $\mathbf{n} \cdot \partial\mathbf{x}/\partial\phi = 0$: we will later introduce a penalty term to ensure that $\left|\mathbf{n} \cdot \partial\mathbf{x}/\partial S\right|$ is small as well.

Interpolation

The concept is that of a beam with deforming section, so that we first interpolate $\overline{\mathbf{x}}$ and $\mathbf{a}$ as functions of S and then interpolate $\mathbf{y}$ and $\mathbf{n}$ as functions of S and ϕ.

Pipe axis quantities

We assume the pipe axis position, $\overline{\mathbf{x}}$, is a polynomial in S:

$$\overline{\mathbf{x}} = \sum_{n=1}^{N} H_n(S)\overline{\mathbf{x}}^n,$$

where $H_n(S)$ are the usual polynomial interpolation functions of order $(N-1)$. The same functions are used to interpolate a rotation triplet,

$$\overline{\boldsymbol{\omega}}(S) = \sum_{n=1}^{N} H_n(S)\overline{\boldsymbol{\omega}}^n,$$

which then gives

$$\mathbf{a} = \mathbf{C}(\overline{\boldsymbol{\omega}}) \cdot \mathbf{A},$$

where $\mathbf{C}$ is the rigid rotation matrix.

Cross-sectional deformation quantities

Numerical experiments with standard shell models have shown that warping effects are important in the geometries and loadings of interest. We, therefore, introduce the Fourier/polynomial interpolation

$$\begin{aligned} y^3(\phi, S) = &\sum_{m=1}^{M}\sum_{p=2}^{P} \overline{H}_m(S) Q_{\text{sym}}(p\phi)(\overline{u}_I^3)_p^m \qquad \text{in the plane} \\ &+ \sum_{m=1}^{M}\sum_{p=2}^{P} \overline{H}_m(S) Q_{\text{asym}}(p\phi)(\overline{u}_0^3)_p^m \qquad \text{out-of-plane} \end{aligned}$$

where H_m are polynomials of the same or lower order as the H_n above

$$Q_{\text{sym}}(p\phi) = \begin{cases} \cos p\phi & \text{for } p \text{ even} \\ \sin p\phi & \text{for } p \text{ odd} \end{cases}$$

$$Q_{\text{asym}}(p\phi) = \begin{cases} \sin p\phi & \text{for } p \text{ even} \\ \cos p\phi & \text{for } p \text{ odd} \end{cases}$$

The number ***P*** gives the order of Fourier interpolation, and the terms "in-plane" and "out-of-plane" refer to symmetries that occur if the motion is symmetric about the plane of the pipe's initial curvature or antisymmetric about that plane.

To model ovalization we assume that

$$\begin{aligned} u^r = & \sum_{m=1}^{M} \overline{H}_m(S)(\overline{u}_U^r)^m \qquad \text{(uniform radial expansion)} \\ & + \sum_{m=1}^{M}\sum_{p=1}^{P} \overline{H}_m(S) Q_{\text{sym}}(p\phi)(\overline{u}_I^r)_p^m \qquad \text{(in-plane)} \\ & + \sum_{m=1}^{M}\sum_{p=1}^{P} \overline{H}_m(S) Q_{\text{asym}}(p\phi)(\overline{u}_0^r)_p^m \qquad \text{(out-of-plane)} \end{aligned}$$

and

$$\begin{aligned} u^t = & \sum_{m=1}^{M}\sum_{p=2}^{P} \overline{H}_m(S) Q_{\text{asym}}(p\phi)(\overline{u}_I^t)_p^m \qquad \text{(in-plane)} \\ & + \sum_{m=1}^{M}\sum_{p=2}^{P} \overline{H}_m(S) Q_{\text{sym}}(p\phi)(\overline{u}_0^t)_p^m \qquad \text{(out-of-plane)} \end{aligned}$$

and for the direction $\mathbf{n}$

$$\begin{aligned} \gamma^3 = & \sum_{m=1}^{M} \overline{H}_m(S)(\overline{\gamma}_u^3)^m \\ & + \sum_{m=1}^{M}\sum_{p=1}^{P} \overline{H}_m(S) Q_{\text{sym}}(p\phi)(\overline{\gamma}_I^3)_p^m \qquad \text{(in-plane)} \\ & + \sum_{m=1}^{M}\sum_{p=2}^{P} \overline{H}_m(S) Q_{\text{asym}}(p\phi)(\overline{\gamma}_0^3)_p^m \qquad \text{(out-of-plane).} \end{aligned}$$

The $p = 0,\ 1$ terms in y^3 and u^t and the $p = 1$ term in γ^3 are omitted since these relative quantities should not include rigid body motion. The implementation of the formulation has been based on choosing linear polynomials for the $H_n(S)$ and $\overline{H}_m(S)$ (referred to as element type ELBOW31) and quadratic polynomials for these functions (element type ELBOW32). Obviously element type ELBOW31 is the lowest order possible for such interpolation. However, we also provide an element in which the $H_n(S)$ are linear, while $M = 1$ and $\overline{H}_1(S) = 1$; that is, $\mathbf{y}$ and γ^3 are constant within an element and, thus, discontinuous from element to element. This is the level of approximation used in routine linear design

formula analyses (Dodge and Moore, 1972) and that has been used previously in nonlinear cases (Hibbitt, 1974). We refer to this as element type ELBOW31B. For the linear elements (ELBOW31, ELBOW31B) one-point integration is used in each element with respect to S; the quadratic element uses a two-point Gauss rule.

The choice of P—the number of terms taken in the Fourier series—has been based on numerical experiments. Ohtsubo and Watanabe (1976) documented a detailed numerical investigation of this choice for their original element, and their results should be relevant to our formulation. For most practical thin-walled piping system applications, we find that six modes are sufficient to predict strains within a few percent.

Strains and constraint

The formulation is discrete Kirchhoff, using the Koiter-Sanders generalized section strains at each point of the pipe wall middle surface:

$$\varepsilon_{\alpha\beta}(\phi, S) = \frac{1}{2}(g_{\alpha\beta} - G_{\alpha\beta})$$

and

$$\kappa_{\alpha\beta}(\phi, S) = B_{\alpha\beta} - b_{\alpha\beta} + \mathrm{sym}(B_{\alpha\gamma}\varepsilon_{\gamma\beta}),$$

where

$$g_{\alpha\beta}(\phi, S) = \frac{\partial \mathbf{x}}{\partial \theta^\alpha} \cdot \frac{\partial \mathbf{x}}{\partial \theta^\beta}$$

is the metric of the pipe wall middle surface and

$$b_{\alpha\beta}(\phi, S) = -\mathrm{sym}\left(\frac{\partial \mathbf{x}}{\partial \theta^\alpha} \cdot \frac{\partial \mathbf{n}}{\partial \theta^\beta}\right)$$

is the curvature of the surface, where $\mathbf{n}$ is made normal to the surface and $G_{\alpha\beta}$ and $B_{\alpha\beta}$ are the same measures in the reference configuration—$\theta^1 = S$ and $\theta^2 = \phi$—and Greek indices span the range (1, 2).

The strain components at any point through the pipe wall are then defined by the Kirchhoff assumption as

$$e_{\alpha\beta}(h) = e_{\alpha\beta} + h\kappa_{\alpha\beta},$$

where h is the material coordinate measuring position through the pipe wall: $-t/2 \leq h \leq t/2$. In applications numerical integration is used through the wall. The integration rule varies from case to case but is usually a five, seven or nine point Simpson rule for inelastic material response involving creep and plasticity.

The discrete Kirchhoff constraint is imposed by associating a penalty with the strain

$$\gamma_t = \mathbf{n} \cdot \frac{\partial \mathbf{x}}{\partial S}.$$

The penalty is obtained from the transverse shear stiffness of the pipe wall, suitably modified for thin walls in the manner of Hughes et al. (1977).

Some simplifying approximations

It seems reasonable to assume that the warping will be small, so that

$$\left|\mathbf{a}_3 \cdot \mathbf{t}_t\right| << 1, \text{ and thus } \mathbf{t}_n = \mathbf{a}_3.$$

Based on this, we write $\overline{\mathbf{n}} = \partial\mathbf{x}/\partial\theta \times \mathbf{a}_3 + \gamma^3\mathbf{a}_3$.

Since $\mathbf{n} = \overline{\mathbf{n}}\left(\overline{\mathbf{n}} \cdot \overline{\mathbf{n}}\right)^{-\frac{1}{2}}$,

$$\frac{\partial\mathbf{n}}{\partial S} = \left(\overline{\mathbf{n}} \cdot \overline{\mathbf{n}}\right)^{-\frac{1}{2}}[\mathbf{I} - \mathbf{nn}] \cdot \frac{\partial\mathbf{n}}{\partial S},$$

where $\mathbf{I}$ is the unit matrix.

We choose

$$\mathbf{n} \cdot \frac{\partial\mathbf{x}}{\partial\theta} = 0,$$

and we impose a penalty to make $\left|\mathbf{n} \cdot \partial\mathbf{x}/\partial S\right|$ small. Then in the computation of the curvatures, $b_{\alpha\beta}$, we approximate $\mathbf{n} \cdot \partial\mathbf{x}/\partial S = 0$, so that

$$b_{\alpha\beta} \approx -(\overline{\mathbf{n}} \cdot \overline{\mathbf{n}})^{-\frac{1}{2}}\mathrm{sym}\left(\frac{\partial\mathbf{x}}{\partial\theta^\alpha} \cdot \frac{\partial\overline{\mathbf{n}}}{\partial\theta^\beta}\right).$$

This completes the basic definition of the element formulation. The formulation allows arbitrary rigid body motions to occur with no strain and uniform thermal expansion with constant strain. The detailed derivation of the strain variations, initial stress matrix, etc., follow directly from the definitions given above, although the forms are complicated. In the actual implementation the nonsymmetric terms in the initial stress matrix caused by our choice of definition of rotation—$\overline{\boldsymbol{\omega}}$—as well as nonsymmetric terms in some load stiffnesses, are neglected. A major implementation approximation is in the modeling of inertia terms in the form of a mass matrix defined by

$$\int_A \rho t\left[\overline{x} \cdot \delta\overline{x} + y \cdot \delta y\right] dA + \int_S \rho t\pi r^3\overline{\boldsymbol{\omega}} \cdot \delta\overline{\boldsymbol{\omega}} dS,$$

where ρ is the mass density of the material.

This is a coarse approximation locally, but the intended applications concern analysis of pipelines, where you are mainly interested in inertia effects associated with locally low-mode response. Table 3.9.1–1 shows a comparison of free vibration frequencies for a single 90° bend restrained against rigid body motion on its axis at one end and prevented from warping (but allowed to ovalize) at both ends. The agreement between the various models and the "standard shell element" is not very close, but we feel it is adequate for the intended use. More accurate modeling of inertia would incur a large penalty in computational cost.

Table 3.9.1–1 Free vibration frequencies for a 90° elbow.

Model			Free vibration frequencies, Hertz				
			Mode 1	Mode 2	Mode 3	Mode 4	Mode 5
Standard shell elements			79.3	83.0	193	195	473
Element type ELBOW31B	2 elements	P=4	73.5	79.0	171	172	474
		P=6	73.0	78.3	169.2	170.6	473
	3 elements	P=4	75.6	81.7	168.7	169.3	184.7
		P=6	75.1	81.0	167.2	167.8	183
Element type ELBOW31	2 elements	P=4	80.0	85.2	170.2	172	418.7
		P=6	79.2	84.4	164.7	165.5	415.5
	3 elements	P=4	82.9	89.0	183.5	185	427.5
		P=6	82.0	88.1	178.1	179.6	415.3

Dimensions:

$R = 0.9144$ m (36 in)

$r = 0.2921$ m (11.5 in)

$t = 0.0127$m (0.5 in)

90° bend

Material properties:

Young's modulus:	2.165×10^{11} N/m^2 (31.4×10^6 lb/in^2)
Poisson's ratio:	0.3
Density:	7822.8 kg/m^3 (7.32×10^{-4} lb sec^2/in^4)
Boundary conditions:	One end restrained on its axis against rigid body motion; both ends restrained against warping.

Reference

- "Pipes and pipebends with deforming cross-sections: elbow elements," Section 29.5.1 of the Abaqus Analysis User's Guide

3.9.2 FRAME ELEMENTS WITH LUMPED PLASTICITY

Product: Abaqus/Standard

Frame elements are designed for the analysis of initially straight, slender beams. The elements are implemented for large displacements and large rotations but small strains. The elastic response of the frame elements follows Euler-Bernoulli beam theory. Plasticity is included in the element's response through a lumped plasticity model with kinematic hardening, which permits yielding only at the ends of the beam. The hardening data are given as a relationship between the generalized force and generalized displacement. Hence, the plastic response of the element is length dependent. The elastic-plastic frame elements are designed to represent plastic hinge formation in frame-like structures, where a single frame element can be used as a member between the structure's nodes.

Degrees of freedom on the element

Frame elements are formulated in terms of the solution variables at user-defined end nodes and extra internal degrees of freedom associated with an internal node. The three-dimensional version of the element is discussed here. The two-dimensional version is found by appropriate reduction of the three-dimensional degrees of freedom. The element has three nodes (two user-defined and one internal), 12 external degrees of freedom, and three internal degrees of freedom. Each of the two end nodes has six external degrees of freedom: three displacements and three rotations. An internal node (at the center of the element) has three displacement degrees of freedom only, as shown in Figure 3.9.2–1.

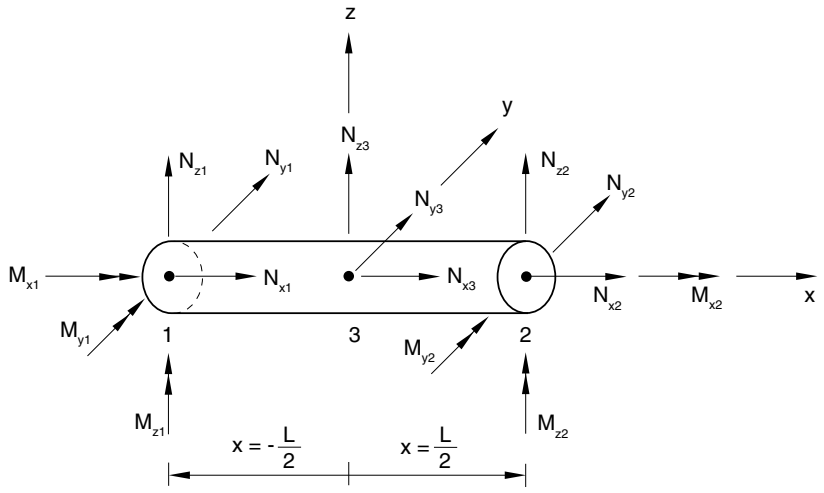


Figure 3.9.2–1 Frame element in space.

The element is formulated in a local system with the *x*-direction representing the axial direction and the *y*- and *z*-directions representing the directions transverse to the frame element axis. In this local coordinate system the element's degrees of freedom can be written

$$\mathbf{q}_1 = \{u_{x1}, u_{y1}, u_{z1}, \phi_{x1}, \phi_{y1}, \phi_{z1}\}^T,$$
$$\mathbf{q}_2 = \{u_{x2}, u_{y2}, u_{z2}, \phi_{x2}, \phi_{y2}, \phi_{z2}\}^T,$$

$$\mathbf{q}_3 = \{u_{x3}, u_{y3}, u_{z3}\}^T, \quad \text{and}$$

$$\mathbf{q} = \left\{\mathbf{q}_1^T, \mathbf{q}_2^T, \mathbf{q}_3^T\right\}^T.$$

The elastic formulation

The elastic response of the element is governed by Euler-Bernoulli beam theory. The displacement interpolation for the deflections transverse to the frame element axis (the *y*- and *z*-directions) uses fourth-order polynomials, allowing for quadratic variation of the curvature along the beam axis. Let $\xi \in [-1, 1]$ be the isoparametric coordinate along the length of the beam. Then,

$$u_y = a_0 + a_1\xi + a_2\xi^2 + a_3\xi^3 + a_4\xi^4,$$

$$u_z = b_0 + b_1\xi + b_2\xi^2 + b_3\xi^3 + b_4\xi^4.$$

The transverse displacement interpolations incorporate exact solutions to force and moment loading at the ends and constant distributed loads along the beam axis (such as gravity loading). The displacement interpolation function along the frame element axis (the *x*-direction) is a second-order polynomial, allowing for linear variation of the axial strain along the frame element axis:

$$u_x = c_0 + c_1\xi + c_2\xi^2.$$

The twist rotation degree of freedom interpolation along the beam axis (rotation about the *x*-axis) is linear, allowing for constant twist strain:

$$\phi_x = d_0 + d_1\xi.$$

The generalized strains, following Euler-Bernoulli beam theory, are

$$\epsilon_x = \frac{2}{L}\frac{\mathrm{d}u_x}{\mathrm{d}\xi}, \quad \kappa_z = \frac{4}{L^2}\frac{\mathrm{d}^2u_y}{\mathrm{d}\xi^2}, \quad \kappa_y = -\frac{4}{L^2}\frac{\mathrm{d}^2u_z}{\mathrm{d}\xi^2}, \quad \epsilon_t = \frac{2}{L}\frac{\mathrm{d}\phi_x}{\mathrm{d}\xi},$$

where ϵ_x is the axial strain, κ_z and κ_y are the beam curvatures, and ϵ_t is the twist strain. The 15 undetermined constants in the interpolation equations for the displacements are determined by introducing the nodal degrees of freedom; that is,

$$u_y(\xi = -1) \stackrel{\text{def}}{=} u_{y1}, \quad \frac{\partial u_z}{\partial \xi}(\xi = 1) \stackrel{\text{def}}{=} -\frac{L}{2}\phi_{y2}, \quad u_x(\xi = 0) \stackrel{\text{def}}{=} u_{x3}, \quad \text{etc.}$$

The interpolations in terms of the nodal degrees of freedom are described below in the section discussing the large-displacement formulation.

The strain-displacement relationship is written in matrix form as

$$\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{q},$$

where $\mathbf{B}$ is a 4×15 matrix and

$$\boldsymbol{\varepsilon} = \{\epsilon_x, \kappa_z, \kappa_y, \epsilon_t\}^T.$$

The elastic stiffness matrix is integrated numerically and used to calculate 15 nodal forces and moments—12 forces/moments (also called generalized forces) associated with the two end nodes,

$$\mathbf{F}_1 = \{N_{x1}, N_{y1}, N_{z1}, M_{x1}, M_{y1}, M_{z1}\}^T,$$

$$\mathbf{F}_2 = \{N_{x2}, N_{y2}, N_{z2}, M_{x2}, M_{y2}, M_{z2}\}^T,$$

and three forces associated with the internal node,

$$\mathbf{F}_3 = \{N_{x3}, N_{y3}, N_{z3}\}^T.$$

The vector of forces and moments for the frame element can be written as

$$\mathbf{F} = \left\{\mathbf{F_1^T}, \mathbf{F_2^T}, \mathbf{F_3^T}\right\}^T.$$

The elastic stiffness is, therefore, a 15×15 matrix relating the force vector, $\mathbf{F}$, and the nodal displacement vector, $\mathbf{q}$:

$$\mathbf{F} = \mathbf{K^e} \cdot \mathbf{q}.$$

Material properties of frame elements can, in general, be temperature dependent. Let us define the elastic strain vector as

$$\boldsymbol{\varepsilon}^{elastic} = \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{th} + \boldsymbol{\varepsilon}^{initial},$$

where $\boldsymbol{\varepsilon}$ denotes the total strain and $\boldsymbol{\varepsilon}^{th}$ denotes the thermal expansion strain, where only the axial strain is nonzero and is given by

$$\epsilon_x^{th} = \alpha(\theta)(\theta - \theta^0) - \alpha(\theta_I)(\theta_I - \theta^0),$$

where

$\alpha(\theta)$ is the thermal expansion coefficient,

θ is the current temperature at the frame element section,

θ^0 is the reference temperature for α, and

θ_I is the user-defined initial temperature at this point ("Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 34.2.1 of the Abaqus Analysis User's Guide).

The temperature field is defined by the user at the element's ends and is assumed to be linear along the element axis but constant within the element cross-section. If the thermal expansion coefficient is temperature dependent, it is evaluated at the nodes. Thermal strains are calculated at the element's end nodes, and thermal strains at the integration points are interpolated from the nodal points using appropriate interpolation schemes: axial strains are interpolated linearly, curvatures are interpolated quadratically, and twist strain is constant along the frame element axis.

Initial generalized strains, $\boldsymbol{\varepsilon}^{initial}$, are calculated from the initial generalized forces given by the user, using the relationship

$$\boldsymbol{\varepsilon}^{initial} = \mathbf{D}^{-1} \cdot \mathbf{F}^{initial},$$

where $\mathbf{D}$ denotes the 4×4 material matrix evaluated at the nodal temperature θ:

$$\begin{bmatrix} AE(\theta) & 0 & 0 & 0 \\ 0 & I_{22}E(\theta) & -I_{12}E(\theta) & 0 \\ 0 & -I_{12}E(\theta) & I_{11}E(\theta) & 0 \\ 0 & 0 & 0 & J_pG(\theta) \end{bmatrix},$$

and $\boldsymbol{A}$ is the cross-section area; $\boldsymbol{E}$ is Young's modulus; $\boldsymbol{G}$ is the shear modulus; I_{11}, I_{12}, and I_{22} are cross-section moments of inertia; and J_pG is the torsional stiffness. The vector of generalized initial forces includes the following components:

$$\mathbf{F}^{initial} = \{N_x, M_z, M_y, M_x\}^T.$$

Initial strains, when needed, are interpolated from the nodal values to the integration points using appropriate interpolators: linear for the axial component, quadratic for the bending components, and constant for the torsional component.

Abaqus integrates the elastic stiffness matrix numerically:

$$\mathbf{K}^e = \int_{-1}^{1} \mathbf{B}^T \mathbf{D}(\theta) \mathbf{B} \frac{L}{2} d\xi,$$

where temperature-dependent material properties are evaluated at the integration points, assuming a linear variation of temperature along the element axis.

For the simplest case of a temperature-independent material and a pipe cross-section, the elastic stiffness matrix can be integrated analytically to give:

Twisting moments at the end nodes:

$$\left\{ \begin{array}{c} M_{x1} \\ M_{x2} \end{array} \right\} = \frac{GJ_p}{L} \left[\begin{array}{cc} 1 & -1 \\ -1 & 1 \end{array} \right] \left\{ \begin{array}{c} \phi_{x1} \\ \phi_{x2} \end{array} \right\}.$$

Axial forces at the nodes:

$$\left\{ \begin{array}{c} N_{x1} \\ N_{x2} \\ N_{x3} \end{array} \right\} = \frac{AE}{3L} \left[\begin{array}{ccc} 7 & 1 & -8 \\ 1 & 7 & -8 \\ -8 & -8 & 16 \end{array} \right] \left\{ \begin{array}{c} u_{x1} \\ u_{x2} \\ u_{x3} \end{array} \right\}.$$

Bending moments at the end nodes and transverse forces for all three nodes:

$$\left\{ \begin{array}{c} N_{y1} \\ N_{z1} \\ M_{y1} \\ M_{z1} \\ N_{y2} \\ N_{z2} \\ M_{y2} \\ M_{z2} \\ N_{y3} \\ N_{z3} \end{array} \right\} = \frac{8EI}{10L^3} [\mathbf{K_m^e}] \left\{ \begin{array}{c} u_{y1} \\ u_{z1} \\ \phi_{y1} \\ \phi_{z1} \\ u_{y2} \\ u_{z2} \\ \phi_{y2} \\ \phi_{z2} \\ u_{y3} \\ u_{z3} \end{array} \right\},$$

where $\mathbf{K_m^e}$ is the bending part of the elastic stiffness matrix and takes the following form:

$$\left[\begin{array}{cccccccccc} 79 & 0 & 0 & 47\frac{L}{2} & 79 & 0 & 0 & -17\frac{L}{2} & -128 & 0 \\ & 79 & -47\frac{L}{2} & 0 & 0 & 79 & 17\frac{L}{2} & 0 & 0 & -128 \\ & & 9L^2 & 0 & 0 & -17\frac{L}{2} & -3\frac{L^2}{2} & 0 & 0 & 32L \\ & & & 9L^2 & 17\frac{L}{2} & 0 & 0 & -3\frac{L^2}{2} & -32L & 0 \\ & & & & 79 & 0 & 0 & -47\frac{L}{2} & -128 & 0 \\ & & & & & 79 & 47\frac{L}{2} & 0 & 0 & -128 \\ & & & & & & 9L^2 & 0 & 0 & -32L \\ & & & & & & & 9L^2 & 32L & 0 \\ & & & sym & & & & & 256 & 0 \\ & & & & & & & & & 256 \end{array} \right].$$

Lumped plasticity model

We assume that the displacement and rotation increments admit an additive decomposition into elastic and plastic parts. Hence,

$$\Delta \mathbf{q} = \Delta \mathbf{q}^{e} + \Delta \mathbf{q}^{pl}.$$

The total forces and moments result from the elastic constitutive relation

$$\mathbf{F} = \mathbf{K}^{e} \cdot \left(\mathbf{q} - \mathbf{q}^{pl} \right).$$

Introduce the lumped plasticity concept such that plastic deformation can develop at the beam external (end) nodes only and develops through plastic rotations (hinges) and plastic axial displacement at one or both nodes. Further assume that the plastic deformation at the external nodes can be caused by the interaction of all three moments and the axial force. Therefore, the vector of plastic deformation at those end nodes has the following form:

$$\mathbf{q^{Pl}} = \left\{ \mathbf{q}_1^{pl}, \mathbf{q}_2^{pl}, \mathbf{0} \right\}^T,$$

$$\mathbf{q}_1^{pl} = \left\{ u_{x1}^{pl}, 0, 0, \phi_{x1}^{pl}, \phi_{y1}^{pl}, \phi_{z1}^{pl} \right\}^T,$$

$$\mathbf{q}_2^{pl} = \left\{ u_{x2}^{pl}, 0, 0, \phi_{x2}^{pl}, \phi_{y2}^{pl}, \phi_{z2}^{pl} \right\}^T.$$

The yield function Φ, called here the plastic interaction surface, is written in terms of the nodal forces and moments. To calculate the increment of plastic deformation, the plastic interaction surface, Φ, is checked at each of the two ends of the frame element during the loading history. In general, the plastic interaction surface is a function of the sectional forces, its plastic cross-sectional capacities, and the hardening parameters. The frame element is elastic if the following conditions are fulfilled at both frame ends:

$$\Phi_1 < 0 \qquad \text{and} \qquad \Phi_2 < 0.$$

The frame element is elastic-plastic if the plastic interaction surface is exceeded at one or both frame ends:

$$(\Phi_1 < 0 \quad \text{and} \quad \Phi_2 \geq 0)$$

$$\text{or} \quad (\Phi_1 \geq 0 \quad \text{and} \quad \Phi_2 < 0)$$

$$\text{or} \quad (\Phi_1 \geq 0 \quad \text{and} \quad \Phi_2 \geq 0).$$

Assuming associated plasticity with the direction of the increment of plastic deformation along the outward normal to the plastic interaction surface, the following relationship holds:

$$\Delta \mathbf{q}_I^{pl} = \Delta \lambda_I \mathbf{V}_I,$$

where $\Delta \lambda_I$ denotes the magnitude of the plastic deformation at end I and $\mathbf{V}_I$ denotes the direction of the plastic flow at that end.

The hardening model

The hardening model follows a nonlinear kinematic hardening rule generalized from the linear Ziegler hardening law and the relaxation term (the recall term), which introduces the nonlinearity. For details on the hardening model, see "Models for metals subjected to cyclic loading," Section 4.3.5. Now introduce the generalized backstress vector, $\boldsymbol{\alpha}_I$, which defines the origin of the moving plastic interaction surface, and define it as

$$\boldsymbol{\alpha}_I = \left\{ \alpha_{N_{xI}}, 0, 0, \alpha_{M_{xI}}, \alpha_{M_{yI}}, \alpha_{M_{zI}} \right\}^T, \qquad I = 1, 2.$$

Thus, the plastic interaction surface can be expressed as a function of the generalized force $\mathbf{S}_I$, $\Phi_I = \Phi_I(\mathbf{S}_I)$ at the end I, where for each component i,

$$S_{iI} = F_{iI} - \alpha_{iI}.$$

The nonlinear kinematic hardening evolution law for the increment of the backstress takes the form

$$\Delta \alpha_{iI} = (C_i V_{iI} - \gamma_i \alpha_{iI}) \Delta \lambda_I, \; i = 1, 4, \; I = 1, 2, \tag{3.9.2–1}$$

where

$$V_{iI} = \frac{\partial \Phi_I}{\partial S_{iI}}.$$

C_i and γ_i are sectional parameters for the ith plastic component, which must be calibrated from the test data defining the hardening response of the cross-section. The parameters C_i are the initial kinematic hardening moduli, and the parameters γ_i determine the rate at which the kinematic hardening modulus decreases with increasing plastic deformation. The test data required for this implementation are the values of the sectional force and moment components as a function of generalized plastic displacements. This will be either the axial force versus plastic axial displacement or a moment versus plastic rotation of a hinge. The data can be given as pairs of values—generalized sectional force versus conjugate generalized plastic displacement—or by specifying a yield stress for the section. The curve fitting algorithm will determine the parameters C_i and γ_i for each component i, since the hardening law is written separately for each sectional force component, F_i, where the range of i depends on the number of forces and moments entering the plastic interaction surface. Integrating the hardening rule, Equation 3.9.2–1, the following evolution law for the backstress is obtained:

$$\alpha_{iI} = \frac{C_i}{\gamma_i} V_{iI} \left(1 - e^{-\gamma_i \Delta\lambda_I}\right) + \alpha_{iI}^0 e^{-\gamma_i \Delta\lambda_I},$$

where the backstress α_{iI}^0 indicates the value of the backstress at the beginning of the increment.

Plastic interaction surface

Plastic interaction surfaces formulated in the generalized sectional variables depend on the cross-section profile. Frame elements with lumped plasticity are valid for tubular cross-sections only, and in the simplest form the interaction surface can be expressed as an ellipsoid in a space of four sectional components: axial force and three moments. Normalized with the ultimate forces and moments for each of the sectional components, the plastic interaction surface, Φ_I, can be written for each end ***I*** of the frame element as

$$\left(\frac{N_{xI} - \alpha_{N_{xI}}}{N_{x0}}\right)^2 + \left(\frac{M_{xI} - \alpha_{M_{xI}}}{M_{x0}}\right)^2 + \left(\frac{M_{yI} - \alpha_{M_{yI}}}{M_{y0}}\right)^2 + \left(\frac{M_{zI} - \alpha_{M_{zI}}}{M_{z0}}\right)^2 - 1 = 0, \quad (3.9.2\text{–}2)$$

where N_{x0}, M_{x0}, M_{y0}, and M_{z0} represent the cross-sectional capacities at initial yield: the axial force and three moments, respectively. Any other cross-sectional profile for which plastic interaction can be approximated well enough by the above ellipsoidal surface can be used within the lumped plasticity concept for frame elements. For two-dimensional problems modeled with frame elements the plastic interaction condition becomes an ellipsoid in the axial force and bending moment plane. By checking the plastic interaction condition at any time of the deformation at both frame element ends, it is determined that if

1. $\Phi_1 < 0$ and $\Phi_2 < 0$, the frame element stays elastic.
2. $\Phi_I < 0$ and $\Phi_J \geq 0\,(I \neq J)$, the frame element is elastic-plastic. If the plastic condition at the end ***J*** is exceeded, an iterative procedure is needed to find the final deformation state at the end of the increment. Either end ***I*** stays elastic and end ***J*** becomes plastic, or both ends become plastic.
3. $\Phi_I \geq 0$ and $\Phi_J \geq 0$, the frame element is elastic-plastic. If the plastic condition is exceeded at one or both nodes, an iterative procedure is needed to find the final deformation state at the end of the increment. Depending upon the ratio of plastic deformation at both ends, one or both ends will become plastic.

The integration of the plasticity model for frame elements follows the same general rule as described in "Integration of plasticity models," Section 4.2.2.

To solve for the value of the deformation and sectional forces at the end of the increment for an arbitrary load increment, an iterative process is required. To set up an appropriate Newton loop, the following relationships are used, with some of them linearized:

Elastic equilibrium equation:

$$F_{iI} = \sum_{J=1}^{2} \sum_{j=1}^{6} K_{iIjJ}^e \left(q_{jJ} - q_{jJ}^{pl}\right),$$

where $\boldsymbol{F}$ stands for the generalized force at the end of the increment.

The associated flow rule:

$$\Delta \mathbf{q}_I^{pl} = \Delta \lambda_I \frac{\partial \Phi_I}{\partial \mathbf{S}_I}. \tag{3.9.2–3}$$

The hardening evolution law:

$$\alpha_{iI} = \frac{C_i}{\gamma_i} V_{iI} \left(1 - e^{-\gamma_i \Delta \lambda_I}\right) + \alpha_{iI}^0 e^{-\gamma_i \Delta \lambda_I}. \tag{3.9.2–4}$$

The backstress definition:

$$S_{iI} = F_{iI} - \alpha_{iI}. \tag{3.9.2–5}$$

The plastic interaction surfaces at both frame ends:

$$\Phi_I \left(\mathbf{S_I}\right) = 0. \tag{3.9.2–6}$$

Large-displacement and large-rotation formulation

The frame elements admit large overall displacements and rotations; however, it is assumed that the strains are small. Accordingly, the nonlinear geometric formulation corresponds to Euler-Bernoulli beam theory superposed on a rotating reference frame. An Euler-Bernoulli displacement field, $\mathbf{q}^{eb}$, is defined relative to this rotation reference configuration, which causes straining. The Euler-Bernoulli displacement field is defined as follows.

Let $\bar{\mathbf{x}}$ and $\bar{\phi}$ be the average position and average rotation of the frame element in the deformed configuration:

$$\bar{\mathbf{x}} \stackrel{\text{def}}{=} \frac{1}{2}(\mathbf{x}_1 + \mathbf{x}_2) \qquad \text{and} \qquad \bar{\boldsymbol{\phi}} = \frac{1}{2}(\boldsymbol{\phi}_1 + \boldsymbol{\phi}_2).$$

The motion of the rotating reference system is defined by the rigid body motion, where the translational part of the motion is the displacement vector $(\bar{\mathbf{x}} - \mathbf{X}_3)$, since $\mathbf{X}_3$ initially corresponds to the element centroid. The rotation part of the motion is the rotation matrix created from the average rotation vector:

$$\bar{\mathbf{R}} = \exp[\hat{\bar{\boldsymbol{\phi}}}].$$

The average rotation defines the rotation of the element's local directions from the reference values $(\mathbf{T}, \mathbf{N}_1, \mathbf{N}_2)$ to the current values $(\mathbf{t}, \mathbf{n}_1, \mathbf{n}_2)$ through

$$\mathbf{t} = \bar{\mathbf{R}} \cdot \mathbf{T} \quad \text{axial direction and}$$

$$\mathbf{n}_\alpha = \bar{\mathbf{R}} \cdot \mathbf{N}_\alpha \quad \text{cross-section directions.}$$

To define the strain-inducing rotation contributions, we multiplicatively decompose the rotation at each node,

$$\mathbf{R}_I \stackrel{\text{def}}{=} \exp[\hat{\boldsymbol{\phi}}_I] = \bar{\mathbf{R}} \cdot \mathbf{r}_I, \tag{3.9.2–7}$$

where $\mathbf{r}_I$ is the strain-inducing part of the nodal rotation. Since the strains are assumed small, define the Euler-Bernoulli rotations at node I as the axial vector $\hat{\boldsymbol{\phi}}_I^{eb}$, such that

$$\mathbf{r}_I = \exp[\hat{\boldsymbol{\phi}}_I^{eb}]. \tag{3.9.2–8}$$

Using Equation 3.9.2–8 in Equation 3.9.2–7, we can solve for $\hat{\boldsymbol{\phi}}_I^{eb}$ by quaternion extraction. In components relative to the reference element coordinate directions

$$\phi_{yI}^{eb} = \hat{\boldsymbol{\phi}}_I^{eb} \cdot \mathbf{N}_1,$$

$$\phi_{zI}^{eb} = \hat{\boldsymbol{\phi}}_I^{eb} \cdot \mathbf{N}_2,$$

$$\phi_{xI}^{eb} = \hat{\boldsymbol{\phi}}_I^{eb} \cdot \mathbf{T}.$$

The Euler-Bernoulli displacement field is the difference between the position of the node relative to the element center and the reference position of the node relative to the reference center rotated to the current configuration by the average rotation:

$$\mathbf{u}_I^{eb} = (\mathbf{x}_I - \bar{\mathbf{x}}) - \bar{\mathbf{R}} \cdot (\mathbf{X}_I - \bar{\mathbf{X}}),$$

or in components relative to the rotated local element coordinate directions

$$u_{xI}^{eb} = \mathbf{u}_I^{eb} \cdot \mathbf{t},$$

$$u_{yI}^{eb} = \mathbf{u}_I^{eb} \cdot \mathbf{n}_1,$$

$$u_{zI}^{eb} = \mathbf{u}_I^{eb} \cdot \mathbf{n}_2.$$

Once the equivalent Euler-Bernoulli displacements and rotations are determined from the nonlinear displacements and rotations, standard expressions following Euler-Bernoulli beam theory are used. The element interpolations are

$$
\begin{aligned}
u_y^{eb} &= \frac{1}{4}(-3\xi + 4\xi^2 + \xi^3 - 2\xi^4)u_{y1}^{eb} + \frac{1}{4}(3\xi + 4\xi^2 - \xi^3 - 2\xi^4)u_{y2}^{eb} + (1 - 2\xi^2 + \xi^4)u_{y3}^{eb} \\
&\quad + \frac{L}{8}(-\xi + \xi^2 + \xi^3 - \xi^4)\phi_{z1}^{eb} + \frac{L}{8}(-\xi - \xi^2 + \xi^3 + \xi^4)\phi_{z2}^{eb}\,, \\
u_z^{eb} &= \frac{1}{4}(-3\xi + 4\xi^2 + \xi^3 - 2\xi^4)u_{z1}^{eb} + \frac{1}{4}(3\xi + 4\xi^2 - \xi^3 - 2\xi^4)u_{z2}^{eb} + (1 - 2\xi^2 + \xi^4)u_{z3}^{eb} \\
&\quad - \frac{L}{8}(-\xi + \xi^2 + \xi^3 - \xi^4)\phi_{y1}^{eb} - \frac{L}{8}(-\xi - \xi^2 + \xi^3 + \xi^4)\phi_{y2}^{eb}\,, \\
u_x^{eb} &= \frac{1}{2}(-\xi + \xi^2)u_{x1}^{eb} + \frac{1}{2}(\xi + \xi^2)u_{x2}^{eb} + (1 - \xi^2)u_{x3}^{eb}\,, \\
\phi_x^{eb} &= \frac{1}{2}(1 - \xi)\phi_{x1}^{eb} + \frac{1}{2}(1 + \xi)\phi_{x2}^{eb}\,.
\end{aligned}
$$

The strain increments, following Euler-Bernoulli beam theory, are

$$
\epsilon_x = \frac{2}{L}\frac{\mathrm{d}u_x^{eb}}{\mathrm{d}\xi}, \quad \kappa_z = \frac{4}{L^2}\frac{\mathrm{d}^2u_y^{eb}}{\mathrm{d}\xi^2}, \quad \kappa_y = -\frac{4}{L^2}\frac{\mathrm{d}^2u_z^{eb}}{\mathrm{d}\xi^2}, \quad \epsilon_t = \frac{2}{L}\frac{\mathrm{d}\phi_x^{eb}}{\mathrm{d}\xi},
$$

where ϵ_x is the axial strain, κ_z and κ_y are the bending strains, and ϵ_t is the twist strain.

Additional data

The user can supply hardening data for sectional forces in the form of pairs of values relating the axial nodal force to the plastic extension and the bending and twisting moments to the plastic rotations. If both the hardening data and the yield stress value for the frame section are omitted, Abaqus assumes that the frame element will remain elastic.

The curve fitting algorithm is used for the evolution hardening equation. At least three pairs of data are required for Abaqus to fit the curve and solve for the constants C_i and γ_i for each generalized force component, i, as shown in Figure 3.9.2–2.

The parameters γ_i are scaled according to the following formula:

$$
\gamma_i = \bar{\gamma}_i V_i^0,
$$

where V_i^0 denotes the component i of the plastic direction at the beginning of plastic deformation. For the plastic interaction surface in the form of Equation 3.9.2–2, the scaling factor is equal to $V_i^0 = 2/S_i^0$.

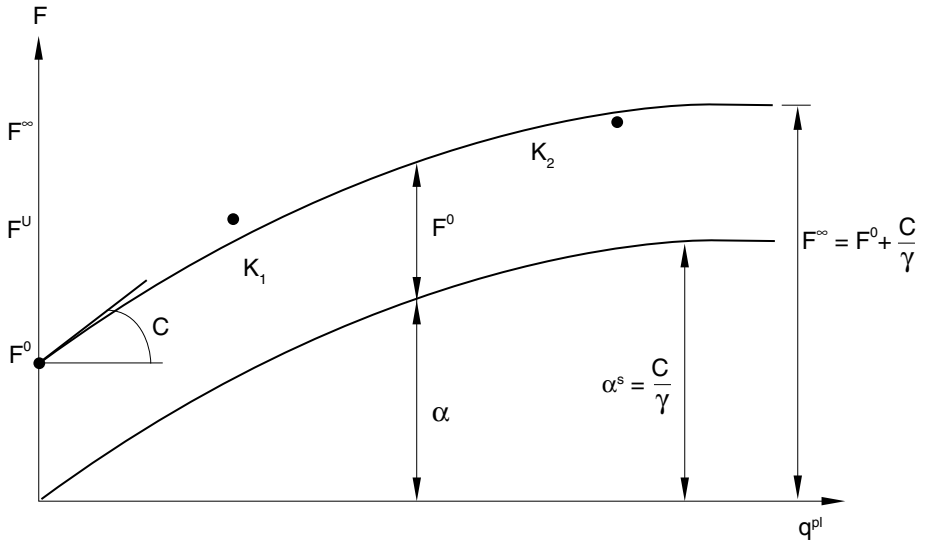


Figure 3.9.2–2 Hardening model for a frame element.

Reference

- "Frame elements," Section 29.4 of the Abaqus Analysis User's Guide

3.9.3 BUCKLING STRUT RESPONSE FOR FRAME ELEMENTS

Product: Abaqus/Standard

Frame elements admit an optional force response in which axial force only is supported by the element. Furthermore, the axial force is constant along the element; all transverse forces and all moments in the element are zero. The axial forces in the element may be linear elastic or may admit a buckling strut response where the force versus axial strain is characterized by a buckling envelope with hysteresis, as described below. For details on the standard frame element response, see "Frame elements with lumped plasticity," Section 3.9.2.

In compression the buckling strut response models, in a simple way, the highly nonlinear buckling and postbuckling damage of slender members when loaded monotonically or cyclically. In tension the response is modeled by isotropic hardening plasticity. The buckling strut envelope is phenomenological, derived from experiments with pipe-like members. Since the description of the buckling envelope includes the outer pipe diameter and the pipe thickness, only pipe cross-section types are permitted with buckling strut response.

The buckling strut response is linear elastic until the compressive loading exceeds P_{cr}, the critical load to cause buckling. The value of P_{cr} is determined with the ISO (International Organization for Standardization) equation, as described below.

Buckling prediction and the ISO equation

The ISO equation is used to predict the onset of buckling in slender members with pipe-like cross-sections. All quantities with dimensions have dimensions of stress. We define I, which is a function of the axial compressive stress, f_c, and the maximum bending stresses about the local *1* and *2* axes, f_{b1} and f_{b2}, by the expression

$$I(f_c, f_{b1}, f_{b2}) = \frac{f_c}{F_c} + \frac{1}{F_b}\sqrt{\left[\frac{c_{m1} f_{b1}}{1 - \frac{f_c}{F_{e1}}}\right]^2 + \left[\frac{c_{m2} f_{b2}}{1 - \frac{f_c}{F_{e2}}}\right]^2}\,.$$

Here, F_c is a characteristic axial compressive stress, F_b is a characteristic bending stress, c_{m1} and c_{m2} are reduction factors corresponding to the cross-section directions *1* and *2*, and F_{e1} and F_{e2} are the Euler buckling stresses corresponding to the *1*- and *2*-directions. The ISO equation states that buckling does not occur as long as

$$I(f_c, f_{b1}, f_{b2}) < 1.0\,.$$

To define the terms in I, we use the following notation:

σ^0	is the yield stress,
E	is Young's modulus of elasticity,
A	is the cross-sectional area,
k_1, k_2	are the effective length factors in the 1- and 2-directions (user-defined),
L_1, L_2	are the unbraced lengths for the 1- and 2-directions (user-defined),

I_{11}, I_{22} are the bending moments of inertia about the local cross-section directions,

Z_e is the elastic section modulus,

Z_p is the plastic section modulus,

$\boldsymbol{r}$ is the radius of gyration.

For pipe sections if $\boldsymbol{D}$ is the outside diameter and $\boldsymbol{t}$ is the pipe wall thickness,

$I_{11} = I_{22} = \frac{\pi}{64}(D^4 - (D-2t)^4)$,

$Z_e = 2I_{22}/D = \frac{\pi}{32D}(D^4 - (D-2t)^4)$,

$Z_p = [D^3 - (D-2t)^3]/6$,

$\boldsymbol{r} = \sqrt{I_{22}/A} = \frac{1}{4}\sqrt{D^2 + (D-2t)^2}$.

The terms in the ISO equation are calculated as follows:

f_c is the axial compressive stress, $f_c = P/A$ with $\boldsymbol{P}$ the axial force in the element.

$f_{b1},\ f_{b2}$ are the maximum bending stresses about the 1 or 2 cross-section axis,
$f_{b1} = M_1/Z_e$, $f_{b2} = M_2/Z_e$ with M_1 and M_2
the bending moments about the 1 and 2 direction.

F_{yc} is the characteristic local buckling stress,
$F_{yc} = \sigma^0$ for $\frac{\sigma^0}{F_e} \le 0.170$,
$F_{yc} = (c_2 - c_3\frac{\sigma^0}{F_e})\sigma^0$ for $\frac{\sigma^0}{F_e} \le 1.911$, where $c_2 = 1.04654873$ and $c_3 = 0.27381606$,
$F_{yc} = F_e$ for $\frac{\sigma^0}{F_e} > 1.911$,
$F_e = 2CE(t/D)$,
$C = 0.3$ is a critical buckling coefficient.

F_c is the characteristic axial compressive stress,
$F_c = [1.0 - 0.28\lambda^2]F_{yc}$ for $\lambda \le 1.34$,
$F_c = \frac{c_1}{\lambda^2}F_{yc}$ for $\lambda > 1.34$, where $c_1 = 0.89282978$,
with $\lambda = \max(\lambda_1, \lambda_2)$,
$\lambda_1 = \frac{k_1 L_1}{\pi r}\sqrt{\frac{F_{yc}}{E}}$, $\lambda_2 = \frac{k_2 L_2}{\pi r}\sqrt{\frac{F_{yc}}{E}}$.

F_b is the characteristic bending stress (for pipe sections $\bar{\xi} = \sigma^0 D/Et$),
$F_b = (Z_p/Z_e)\sigma^0$ for $\bar{\xi} \le 0.0517$,
$F_b = (c_4 - 2.58\bar{\xi})(Z_p/Z_e)\sigma^0$ for $0.0517 < \bar{\xi} \le 0.1034$,
$F_b = (c_5 - 0.76\bar{\xi})(Z_p/Z_e)\sigma^0$ for $0.1034 < \bar{\xi} \le 120\sigma^0/E$,
where $c_4 = 1.133386$ and $c_5 = 0.945198$.

F_{e1}, F_{e2} are Euler buckling stresses corresponding to the 1 or 2 directions,
$F_{e1} = \frac{F_{yc}}{\lambda_1^2}$ and $F_{e2} = \frac{F_{yc}}{\lambda_2^2}$.

c_{m1}, c_{m2} are reduction factors corresponding to the cross-section directions 1 and 2, respectively. These factors are user-defined as functions of the end moments, compression stress, and Euler buckling stresses. The default value for each factor is 0.85.

If switching between standard frame element response and buckling strut response is permitted, the one-time-only switch to buckling strut response occurs when $I(f_c, b_{b1}, f_{b2}) = 1.0$. The ISO equation provides the value of critical load, P_{cr}, which is defined as the value of axial force $f_c A$ in the element when the ISO equation is satisfied. To prevent switching in cases where negligible axial force exists with large bending moments, an additional inequality is used. This additional check, called the strength equation, takes the following form:

$$S = \frac{f_c}{F_{yc}} + \frac{1}{F_b}\sqrt{f_{b1}^2 + f_{b2}^2}.$$

For a frame element to switch to buckling strut behavior, both the ISO equation and the strength equation must be satisfied, $I = 1.0$ and $S \leq 1.0$. If buckling strut response is requested for the element from the beginning of the analysis, bending moments cannot be supported by the element. In this case the ISO equation becomes the simplified statement that no buckling occurs for

$$f_c < F_c \quad \text{and} \quad P_{cr} = F_c A.$$

Marshall strut envelope

The Marshall strut envelope defines the postbuckling damaged elasticity model and the hysteretic loop response. To define the Marshall strut envelope, the value of P_{cr} and the following seven constants are needed:

- ξ is the coefficient defining $P_y = \xi \sigma^0 A$ ($\xi = 0.95$),
- γ is the isotropic hardening slope coefficient (0.02),
- α_0 is the coefficient defining $\alpha = \alpha_0 + \alpha_1 \frac{L}{D}$, ($\alpha_0 = 0.03$),
- α_1 is the coefficient defining $\alpha = \alpha_0 + \alpha_1 \frac{L}{D}$, ($\alpha_1 = 0.004$),
- κ is the force coefficient (0.28),
- β is the slope coefficient (0.02), and
- ζ is the force coefficient (min(1.0, $5.8(\frac{t}{D})^{0.7}/\xi$)).

The values in parentheses are the default values supplied by Abaqus, and the value of P_{cr} is found from the ISO equation as explained above.

The Marshall envelope governs the compressive and tensile response of the strut as shown in Figure 3.9.3–1. The dotted lines in the interior of the envelope indicate the damaged-elastic modulus defining the loading-unloading force versus strain path.

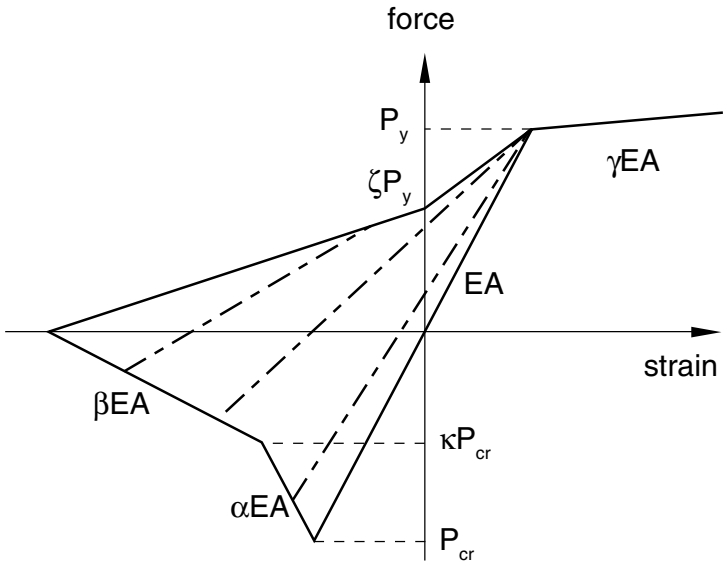


Figure 3.9.3–1 Marshall strut theory buckling envelope.

References

- "Frame elements," Section 29.4.1 of the Abaqus Analysis User's Guide
- "Frame section behavior," Section 29.4.2 of the Abaqus Analysis User's Guide

3.9.4 TUBE SUPPORT ELEMENTS

Product: Abaqus/Standard

These elements are provided for the specific case of modeling the interaction between a tube and a support that is not always in contact with the tube during dynamic events. The tube is assumed to have a circular section and can interact with one of two tube support geometries: a circular hole and an "egg-crate" support. Two interface elements of this type are provided, one for each geometry, as shown in Figure 3.9.4–1 and Figure 3.9.4–2. As indicated in Figure 3.9.4–2, one cylindrical geometry interface is needed to model the interaction of the tube with a circular hole, while Figure 3.9.4–1 shows that several unidirectional geometry elements are needed to model the interaction with an egg-crate—one element perpendicular to each pair of egg-crate faces.

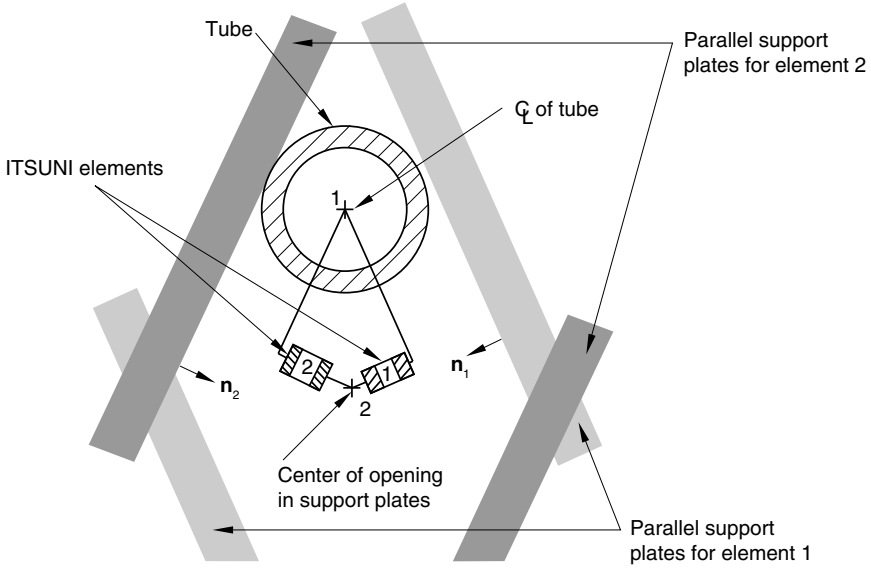


Figure 3.9.4–1 ITSUNI elements for tube/"egg-crate" support interaction.

The interface elements themselves consist of a spring and friction link and a dashpot, as shown in Figure 3.9.4–3. The spring is assumed to behave as shown in Figure 3.9.4–4: when there is no contact between the tube and the support, no force is transmitted by the spring; when the tube is in contact with the support, the force increases as the tube wall is deformed. This force can be modeled as a linear or a nonlinear function of the relative displacement between the axis of the tube and the center of the hole in the support.

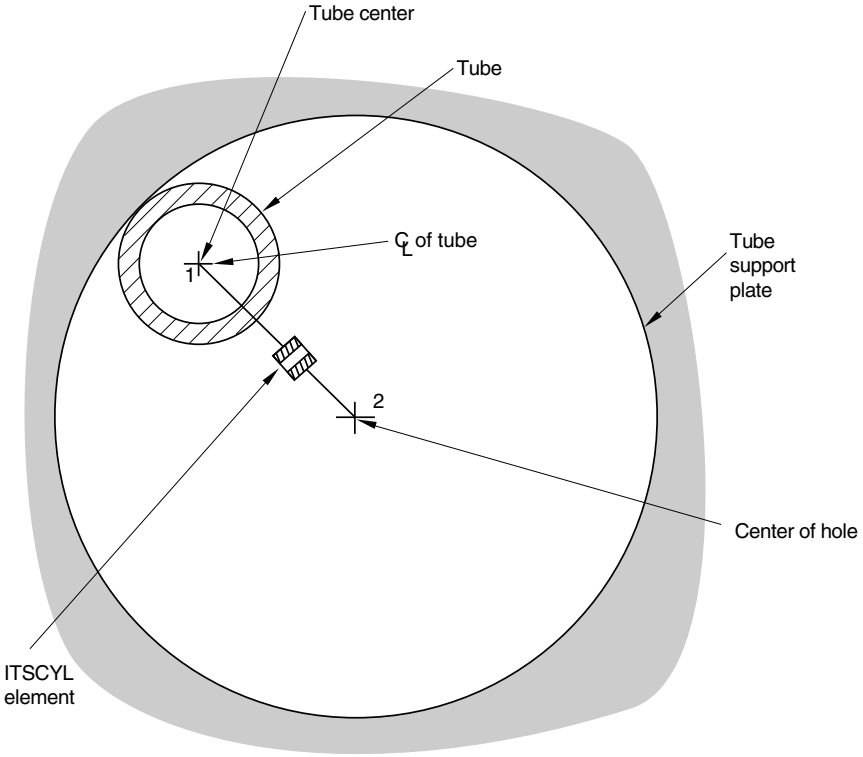


Figure 3.9.4–2 ITSCYL element for tube/drilled hole support interaction.

The frictional part of the spring and friction link uses the Coulomb friction model in Abaqus: that model is described in "Coulomb friction," Section 5.2.3.

The dashpot is provided to model fluid effects in the annulus between the tube and the support plate. Its behavior can be linear or nonlinear. The model assumes that shear forces created by the fluid are negligible, so that the only shear forces transmitted by one of these interface elements are the frictional forces caused by direct contact between the tube and the support.

A major simplification in these elements that saves considerable computational effort in dynamic applications is the assumption that impacts between the tube and its support plates involve no instantaneous transfer of momentum or energy loss: the standard impact algorithm of Abaqus/Standard used with gap and other interface elements (and described in "Intermittent contact/impact," Section 2.4.2) is not needed. This simplification derives from the assumption that these elements will be used in conjunction with beam element models of the tube, so the tube section is defined by the position and orientation of its axis and local deformation of the cross-section of the tube is neglected. In reality, when the tube hits a support, initially only a small part of the tube wall loses momentum so that there is—instantaneously—only a small loss of kinetic energy. This instantaneous energy loss is neglected when these elements are used. The subsequent flattening of the tube wall is modeled by the spring link in the element, acting between the node on the tube

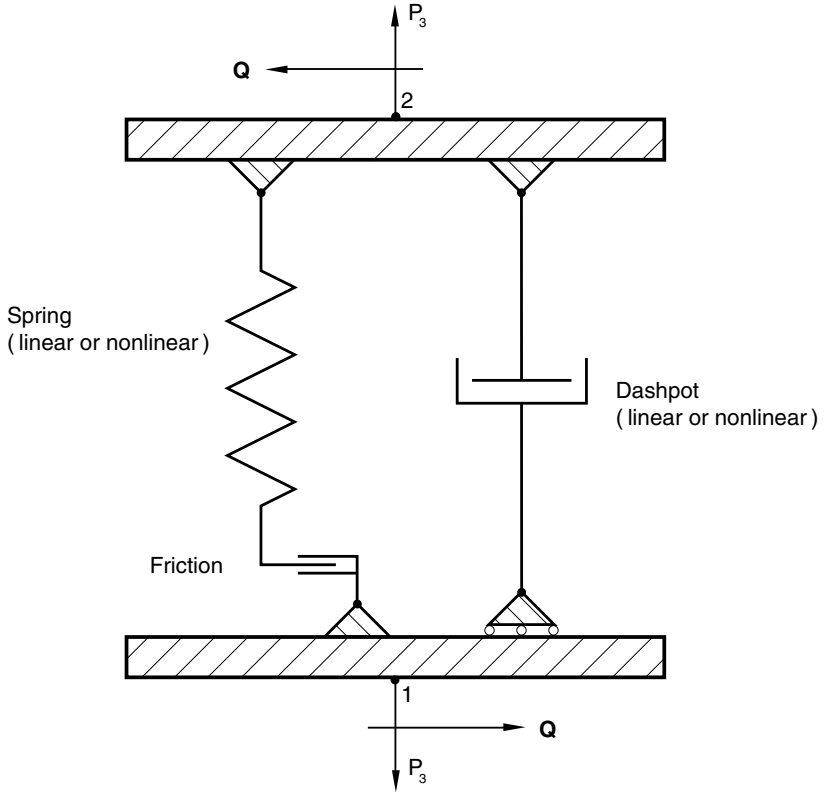


Figure 3.9.4–3 Tube support element behavior.

axis and the node representing the center of the hole. Thus, the modeling of this local flattening behavior as an equivalent spring provides the simplification that instantaneous impact calculations are not needed. In cases where this approach is not reasonable, gap elements can be used instead of these special interface elements, at the cost of more computational effort.

The remainder of this section discusses the kinematic definitions used in these elements and their contributions to the overall equilibrium equations and to the Jacobian (stiffness) matrix needed in the Newton solution of those equations.

Geometry and kinematics

Each tube support element has two nodes. One node represents the axis of the tube, the other is the center of the hole in the support plate (or midway between a pair of parallel sides of an "egg-crate").

Let $\mathbf{a}_2$ be a unit vector along the axis of the tube, and let $\mathbf{a}_3$ be a unit vector along the axis of the interface element (that is, perpendicular to the parallel sides of the support in the "unidirectional" interface that is used with egg-crate supports, and parallel to the line joining the two nodes of the element

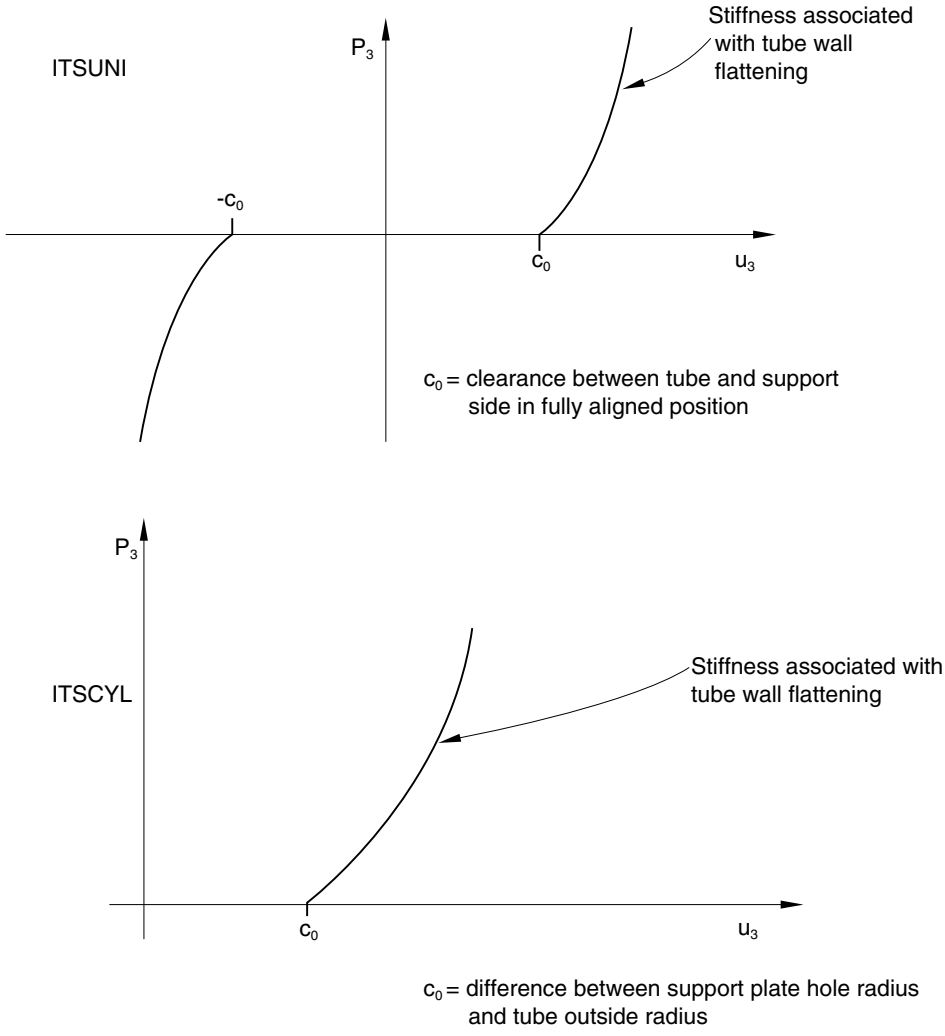


Figure 3.9.4–4 Nonlinear spring behavior in ITS elements to model clearance and tube flattening.

in the "cylindrical" interface that is used with circular holes). It is assumed that $\mathbf{a}_3$ is in the cross-section of the tube and, hence, orthogonal to $\mathbf{a}_2$. We define a third basis vector as

$$\mathbf{a}_1 = \mathbf{a}_2 \times \mathbf{a}_3.$$

Let $\mathbf{x}^N$ be the current position of node N of the element at any point in time (here N is 1 or 2).

Relative displacements in the element are measured from the position when the tube and the hole in the support plate are exactly aligned; that is, when the nodes of the element are at the same location. They are defined as follows:

axial to the interface element,

$$u_3 = (\mathbf{x}^1 - \mathbf{x}^2) \cdot \mathbf{a}_3;$$

axial to the tube,

$$u_2 = (\mathbf{x}^1 - \mathbf{x}^2) \cdot \mathbf{a}_2;$$

and

tangential, in the plane of the tube's cross-section,

for the unidirectional case:

$$u_1 = (\mathbf{x}^1 - \mathbf{x}^2) \cdot \mathbf{a}_1,$$

and, for the cylindrical case:

$$u_1 = \int (d\mathbf{x}^1 - d\mathbf{x}^2) \cdot \mathbf{a}_1.$$

The basis vector—$\mathbf{a}_2$, along the axis of the tube and of the hole in the support plate—is assumed to be fixed. In the unidirectional element the $\mathbf{a}_1$ and $\mathbf{a}_3$ vectors are also fixed. In the cylindrical interface $\mathbf{a}_3$ is parallel to the line joining the nodes of the element, so

$$\mathbf{a}_3 = (\mathbf{x}^1 - \mathbf{x}^2)/l,$$

where

$$l = \sqrt{(\mathbf{x}^1 - \mathbf{x}^2) \cdot (\mathbf{x}^1 - \mathbf{x}^2)}.$$

Therefore,

$$\mathbf{a}_1 = \mathbf{a}_2 \times \mathbf{a}_3 = \mathbf{a}_2 \times (\mathbf{x}^1 - \mathbf{x}^2)/l.$$

Thus, for this element

$$u_1 = \int \frac{1}{l} \mathbf{a}_2 \times (\mathbf{x}^1 - \mathbf{x}^2) \cdot (d\mathbf{x}^1 - d\mathbf{x}^2)$$

and

$$du_1 = \frac{1}{l} \mathbf{a}_2 \times (\mathbf{x}^1 - \mathbf{x}^2) \cdot (d\mathbf{x}^1 - d\mathbf{x}^2).$$

For simplicity we replace the integral with the backward difference approximation

$$\Delta u_1 = [\frac{1}{l}\mathbf{a}_2 \times (\mathbf{x}^1 - \mathbf{x}^2)]\Big|_{t+\Delta t} \cdot (\Delta\mathbf{x}^1 - \Delta\mathbf{x}^2).$$

Forces in the element

The element generates an axial force—P_3, parallel to $\mathbf{a}_3$—and two shear forces—Q_1, parallel to $\mathbf{a}_1$, and Q_2, parallel to $\mathbf{a}_2$. In addition, because the nodes of the element are at the center of the tube and at the center of the hole in the support, while the interaction forces between the tube and its support are transmitted at the point of contact of the tube with the support, these forces also cause moments at the nodes of the element. It is assumed that the moments caused by P_3 and by Q_2 are not significant and can be neglected. The only moments considered are $Q_1\, d_1/2$ at node 1, the center of the tube, and $Q_1\, d_2/2$ at node 2, the center of the hole. Here d_1 is the outside diameter of the tube and d_2 is the diameter of the hole for the cylindrical interface or is the distance between the parallel support plates for the uniaxial interface (see Figure 3.9.4–5).

The virtual work contribution of the element is, then,

$$\delta W^I = P_3\, \delta u_3 + Q_1\, \delta u_1 + Q_2\, \delta u_2 + Q_1 \frac{d_1}{2}\, \delta\phi_2^1 - Q_1 \frac{d_2}{2}\, \delta\phi_2^2,$$

where $\delta\phi_2^N$ is the virtual rate of rotation about the $\mathbf{a}_2$-axis at node $\boldsymbol{N}$.

From this expression the contribution of the element to the Jacobian (stiffness) matrix of the equilibrium equations is immediately available as

$$\begin{aligned} d\delta W^I = {} & \delta u_3\, dP_3 + (\delta u_1 + \frac{d_1}{2}\, \delta\phi_2^1 + \frac{d_2}{2}\, \delta\phi_2^2)\, dQ_1 \\ & + \delta u_2\, dQ_2 + P_3\, d\delta u_3 + Q_1\, d\delta u_1. \end{aligned}$$

The "initial stress" terms,

$$P_3\, d\delta u_3 + Q_1\, d\delta u_1,$$

are only nonzero for the cylindrical interface, for which

$$u_3 = \sqrt{(\mathbf{x}^1 - \mathbf{x}^2) \cdot (\mathbf{x}^1 - \mathbf{x}^2)} = l,$$

so that

$$\delta u_3 = \mathbf{a}_3 \cdot (\delta\mathbf{x}^1 - \delta\mathbf{x}^2),$$

and so

$$d\delta u_3 = \frac{1}{u_3}(\delta\mathbf{x}^1 - \delta\mathbf{x}^2) \cdot [\mathbf{I} - \mathbf{a}_3\, \mathbf{a}_3] \cdot (d\mathbf{x}^1 - d\mathbf{x}^2).$$

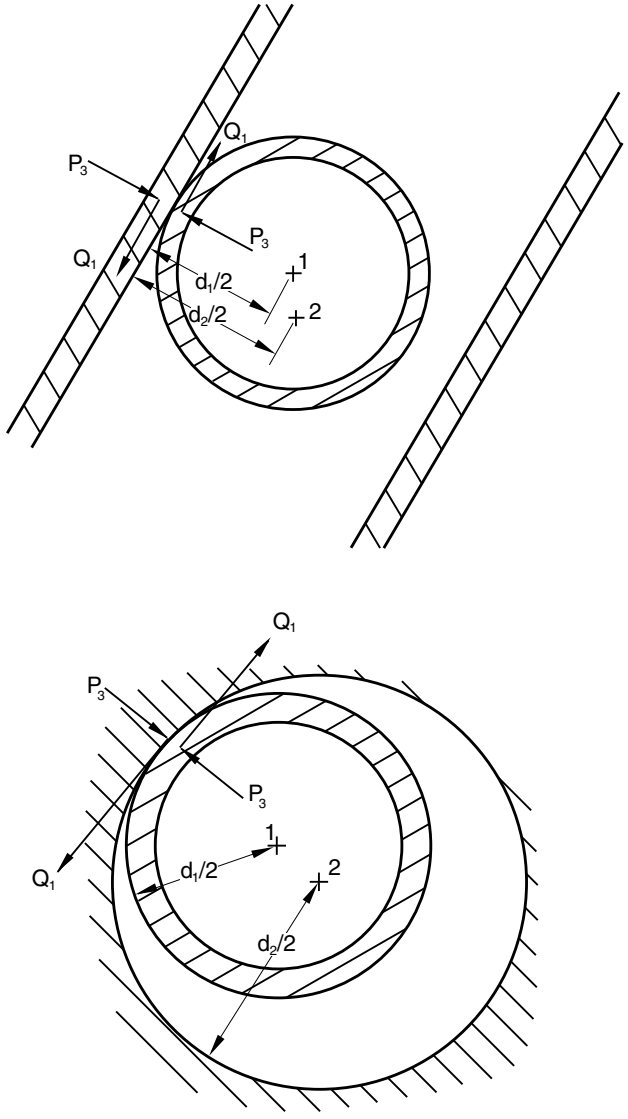


Figure 3.9.4–5 Contact forces in the cross-section.

Also, for this element,

$$\delta u_1 = \frac{1}{u_3}\mathbf{a}_2 \times (\mathbf{x}^1 - \mathbf{x}^2) \cdot (\delta\mathbf{x}^1 - \delta\mathbf{x}^2),$$

and so

$$d\delta u_1 = \frac{1}{u_3}(\delta \mathbf{x}^1 - \delta \mathbf{x}^2) \cdot [\mathbf{a}_3\, \mathbf{a}_1 - 2\mathbf{a}_1\, \mathbf{a}_3] \cdot (d\mathbf{x}^1 - d\mathbf{x}^2).$$

This term is not symmetric.

The "initial stress" terms for the cylindrical interface are, therefore,

$$P_3\, d\delta u_3 + Q_1\, d\delta u_1 =$$
$$(\delta \mathbf{x}^1 - \delta \mathbf{x}^2) \cdot \frac{1}{u_3}\left[P_3[\mathbf{I} - \mathbf{a}_3\, \mathbf{a}_3] + Q_1[\mathbf{a}_3\, \mathbf{a}_1 - 2\mathbf{a}_1\, \mathbf{a}_3]\right] \cdot (d\mathbf{x}^1 - d\mathbf{x}^2).$$

The other terms in the stiffness matrix are associated with changes in the forces in the element, dP_3, dQ_1, and dQ_2. We assume P_3 is made up of a spring force that is a function of u_3 and a dashpot force that is a function of $\dot{u}_3$:

$$P_3 = P_3^s + P_3^d,$$

$$P_3^s = P_3^s(u_3), \quad P_3^d = P_3^d(\dot{u}_3),$$

so that

$$dP_3 = \frac{dP_3^s}{du_3}\, du_3 + \frac{dP_3^d}{d\dot{u}_3}\, d\dot{u}_3.$$

For the implicit integration operator used for nonlinear dynamic analysis in Abaqus/Standard,

$$d\dot{u}_3 = \left(\frac{d\dot{u}}{du}\right) du_3,$$

where

$$\frac{d\dot{u}}{du} = \frac{1}{\Delta t}\, \frac{\gamma}{\beta},$$

where Δt is the time increment and γ and β are the parameters of the integration operator.

Thus,

$$dP_3 = \left(\frac{dP_3^s}{du_3} + \frac{d\dot{u}}{du}\, \frac{dP_3^d}{d\dot{u}_3}\right) du_3.$$

The values of dQ_1 and dQ_2 come from the friction theory and are defined from du_1, du_2, and du_3 by that theory (see "Coulomb friction," Section 5.2.3).

In summary,

$$dP_3 = K_{33}\, du_3$$

and

$$dQ_\alpha = K_{\alpha\beta}\, du_\beta + K_{\alpha 3}\, du_3, \quad \alpha, \beta = 1, 2,$$

so the stiffness contribution of the element is

$$d\delta W^I = \left\lfloor \delta u_3 \ \delta u_1 \ \delta u_2 \ \delta\phi_2^1 \ \delta\phi_2^2 \right\rfloor \begin{bmatrix} K_{33} & 0 & 0 & 0 & 0 \\ K_{13} & K_{11} & K_{12} & 0 & 0 \\ K_{23} & K_{21} & K_{22} & 0 & 0 \\ \frac{d_1}{2}K_{13} & \frac{d_1}{2}K_{11} & \frac{d_1}{2}K_{12} & 0 & 0 \\ \frac{d_2}{2}K_{13} & \frac{d_2}{2}K_{11} & \frac{d_2}{2}K_{12} & 0 & 0 \end{bmatrix} \begin{Bmatrix} du_3 \\ du_1 \\ du_2 \\ d\phi_2^1 \\ d\phi_2^2 \end{Bmatrix} + P_3\, d\delta u_3 + Q_1\, d\delta u_1 \ .$$

This matrix is not symmetric if Q_1, dQ_1, or dQ_2 is nonzero. Without friction they are zero, and the terms in K_{33} and P_3 are the only nonzero terms. With relatively small friction coefficients in dynamic applications the terms $K_{\alpha 3}$ and—if the tube diameter is not very large—$(d_1/2)K_{1\alpha}$, $(d_1/2)K_{13}$, $(d_2/2)K_{1\alpha}$, and $(d_2/2)K_{13}$ can be neglected and, thus, a symmetric approximation to the Jacobian matrix used without serious degradation of the convergence rate of the Newton solution of the equilibrium equations.

Reference

- "Tube support elements," Section 32.8.1 of the Abaqus Analysis User's Guide

3.9.5 LINE SPRING ELEMENTS

Product: Abaqus/Standard

The line spring elements in Abaqus/Standard provide a computationally inexpensive tool for the analysis of part-through cracks in plates and shells. The basic concept was first proposed by Rice (1972) and has been further discussed by Parks and White (1982). The "line spring" is a series of one-dimensional finite elements placed along the part-through flaw, which allows local flexibility of one side of the flaw with respect to the other (points *A* and *B* in Figure 3.9.5–1). This local flexibility is calculated from existing solutions for single edge notch specimens in plane strain (Figure 3.9.5–2). The approach is computationally inexpensive compared to fully three-dimensional models of the vicinity of the flaw; it is also approximate because of the use of two-dimensional solutions embedded in the shell model. Practical experience with the method on typical geometries has shown that, for several important geometries, the method provides acceptable accuracy.

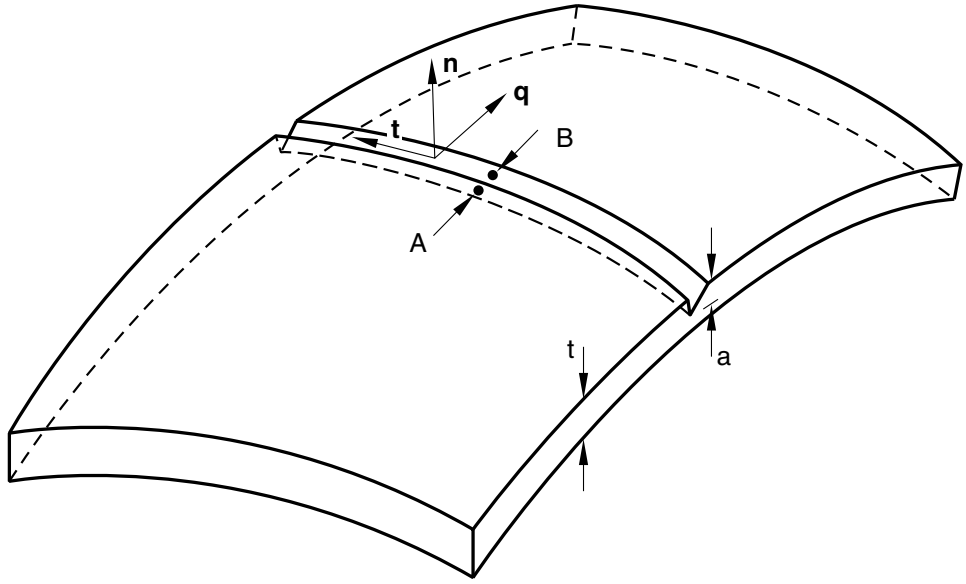


Figure 3.9.5–1 Surface geometry; line spring modeling. Side *B* of the element contains nodes *1*, *2*, and *3*; and for LS6 elements side *A* contains nodes *4*, *5*, and *6*.

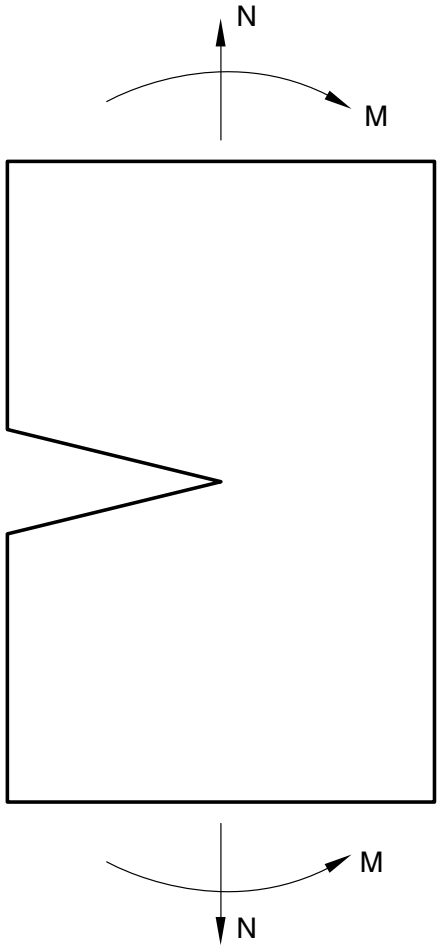


Figure 3.9.5–2 Line spring compliance calibration model.

This section discusses the geometric and kinematic basis of the elements as well as the equilibrium statement and the development of the local solutions that define the constitutive relationships. The constitutive relations are expressed in terms of the forces and moments carried across the crack and the relative displacements and rotations of points on opposite sides of the crack (A and B in Figure 3.9.5–1), and are derived from local solutions to single edge cracked plane strain specimens. Elastic and fully plastic (limit analysis) solutions are used to construct an approximate elastic-plastic model.

At each point along the flaw a local orthonormal basis system is defined $(\mathbf{t},\ \mathbf{n},\ \mathbf{q})$, with $\mathbf{t}$ the tangent to the shell along the flaw, $\mathbf{n}$ the normal to the shell, and $\mathbf{q}$ defined as

$$\mathbf{q} = \mathbf{t} \times \mathbf{n}.$$

We use the shell normal, $\mathbf{n}$, to determine the side of the shell on which the flaw occurs; flaws that open on the positive $\mathbf{n}$ side are given positive flaw depths to indicate this, and those on the negative $\mathbf{n}$ side are given negative flaw depths. The relative motion between two points—*A* and *B* in Figure 3.9.5–1—on opposite sides of the flaw but otherwise at the same place, then defines a set of six generalized strains as follows. Side *B* of the element contains nodes *1*, *2*, *3*; and for LS6 elements side *A* contains nodes *4*, *5*, and *6*.

Mode I:

opening displacement	$\Delta u_I = (\mathbf{u}_B - \mathbf{u}_A) \cdot \mathbf{q}$
opening rotation	$\Delta \phi_I = (\boldsymbol{\phi}_B - \boldsymbol{\phi}_A) \cdot \mathbf{t}$

Mode II: through-thickness shear is defined by the relative displacement,

$$\Delta u_{II} = (\mathbf{u}_B - \mathbf{u}_A) \cdot \mathbf{n}$$

Mode III: antiplane shear is defined by the relative displacement,

$$\Delta u_{III} = (\mathbf{u}_B - \mathbf{u}_A) \cdot \mathbf{t},$$

and by the relative rotation,

$$\Delta \phi_{III} = (\boldsymbol{\phi}_B - \boldsymbol{\phi}_A) \cdot \mathbf{q}.$$

The relative rotation $\Delta \phi_{II} = (\boldsymbol{\phi}_B - \boldsymbol{\phi}_A) \cdot \mathbf{n}$ plays no role in the deformation.

These relative motions form the kinematic basis of the element.

Since the line spring elements are written for geometrically linear analysis only, the first variations of these relative motions are identical to the above definitions, with the total values replaced by their variations.

Virtual work contribution

The virtual work contribution of the element is defined as

$$\delta W = \int_L (N_I \Delta\delta u_I + M_I \Delta\delta \phi_I + N_{II} \Delta\delta u_{II} + M_{II} \Delta\delta \phi_{II} + N_{III} \Delta\delta u_{III} + M_{III} \Delta\delta \phi_{III})\, dL,$$

where N_I, M_I, etc., are forces and moments per unit length of flaw that are conjugate to the corresponding relative displacement and rotation values. In the above expression the integration is taken along the entire flaw.

Interpolation

The elements use quadratic interpolation of displacement and rotation components along the crack, so they are compatible with the second-order shell elements (S8R, S8R5, S9R5, STRI65).

Two line spring elements are provided—LS6 is a general element for use with arbitrary flaws in a shell, while LS3S is provided for Mode I use in cases when the crack lies on a plane of symmetry and the deformation will be symmetric about the same plane, so that only one-half of the geometry must be modeled.

Elasticity

The Mode I line spring compliance is based on a single edge notched specimen subject to far-field tension and bending, as shown in Figure 3.9.5–2. This compliance is

$$\left\{ \begin{array}{c} u \\ \theta \end{array} \right\} = \left[G \right] \left\{ \begin{array}{c} N \\ M \end{array} \right\},$$

where the matrix $\left[G\right]$ can be obtained from the energy compliance calibrations of Rice (1972). The inverse of $\left[G\right]$ provides the Mode I stiffness per unit length of flaw, relating N_I and M_I to Δu_I and $\Delta \phi_I$. Similar results in Mode II and Mode III complete the elastic stiffness.

Stress intensity factors are calculated as

$$K_I = \frac{(\pi a)^{1/2}}{t^2} \left[F_1 \left(\frac{a}{t} \right) N_I t + F_2 \left(\frac{a}{t} \right) M_I \right],$$

where approximate expressions for F_1 and F_2 are given by Tada et al. (1973), and similar expressions apply for Mode III and (without F_2) for Mode II. The ***J***-integral is then calculated by combining these stress intensities as

$$J = \frac{K_I^2}{E'} + \frac{K_{II}^2}{E'} + \frac{K_{III}^2}{2G},$$

where

$$E' = E/(1-\nu^2) \quad \text{and} \quad G = E/2(1+\nu);$$

E is Young's modulus and ν is Poisson's ratio.

Plasticity

The elastic-plastic line spring model in Abaqus is based on Mode I response only, since no theory is available for an elastic-plastic line spring model in mixed mode loading. For convenience we define a generalized "strain" vector as

$$q_1 = \Delta u_I$$
$$q_2 = \Delta \phi_I$$
$$\mathbf{q} = \left\{ \begin{array}{c} q_1 \\ q_2 \end{array} \right\}$$

and conjugate generalized "forces" as

$$Q_1 = N_I$$
$$Q_2 = M_I$$
$$\mathbf{Q} = \left\{ \begin{array}{c} Q_1 \\ Q_2 \end{array} \right\}.$$

The formalism of a simple associated flow, isotropic hardening plasticity model is used as follows. The generalized strain rate is decomposed into elastic and plastic response as

$$d\mathbf{q} = d\mathbf{q}^{el} + d\mathbf{q}^{pl},$$

and the Mode I elasticity relationship described above is used to define the generalized stresses:

$$\mathbf{Q} = \mathbf{D}^{el} \cdot \mathbf{q}^{el}.$$

The plastic strain rate is defined to be normal to a yield surface, Φ:

$$d\mathbf{q}^{pl} = d\overline{q}^{pl} \frac{\partial \Phi}{\partial \mathbf{Q}},$$

where $\Phi = \Phi(\mathbf{Q}, \overline{q}^{pl})$ is the yield function, whose definition is discussed in detail below, and $\overline{q}^{pl}$ is a scalar hardening parameter used to provide isotropic hardening. The hardening is calculated from the basic stress-strain data for the material by a work equivalency argument. The plastic work per unit length of flaw is

$$dW^{pl} = \mathbf{Q} \cdot d\mathbf{q}^{pl},$$

and is also given by

$$dW^{pl} = \int \int \sigma_0 d\overline{\varepsilon}^{pl} dz dy,$$

where $\sigma_0(\overline{\varepsilon}^{pl})$ is the uniaxial stress-strain behavior of the material and z and y measure position through the thickness and along the length of the single edge notch specimen. We approximate this second expression by

$$dW^{pl} = \sigma_0 d\overline{\varepsilon}^{pl} (t-a)^2 f,$$

where $\sigma_0(\overline{\varepsilon}^{pl})$ is a representative value of the yield stress (at an equivalent plastic strain of $\overline{\varepsilon}^{pl}$); t is the shell thickness; a is the flaw depth; and f is a constant, introduced to provide a matching to numerical results for the specimen. Parks and White (1982) suggest choosing $f = 0.4$, and this value is used in Abaqus.

The yield surface is defined with respect to the generalized stress variables N_I and M_I as follows. Following Rice (1972), we define

$$X = \frac{\sqrt{3}N_I}{2\sigma_0(t-a)}$$

and

$$Y = \frac{\sqrt{3}}{2\sigma_0(t-a)^2}(M_I + t/2N_I).$$

Then for $X \leq 2Y$ the yield function, Φ, is taken as an envelope to limit analysis results, as proposed by Rice:

$$\Phi_1 = (X-0.3)^2 + 4.41\left(Y - \frac{X}{2}\right)^2 - 0.49 = 0.$$

Otherwise, we use

$$\Phi_2 = \alpha_1 X + \alpha_2 Y + \alpha_3 X^3 Y + \alpha_4 X Y^3 - 1 = 0,$$

with

$$\begin{aligned}
\alpha_1 &= 2.264568665,\\
\alpha_2 &= -2.79974917,\\
\alpha_3 &= -0.99731078,\\
\alpha_4 &= 5.0627466.
\end{aligned}$$

This surface is chosen to blend continuously with Φ_1 at $X = 2Y$ and as a reasonable estimate of the behavior for $X > 0, Y < 2X$. It is, otherwise, arbitrary. Rice (1972) points out that at $X = 1.0, Y = 0.5$, the yield surface will have a vertex. The smooth surface used in Abaqus has been adopted for numerical reasons. This smoothness restricts the possible flow behavior at $X = 1, Y = 0.5$, but we assume that this is not a critical issue.

These surfaces are shown in Figure 3.9.5–3. The figure also indicates a region where the model is not appropriate (because the crack will close). Warning messages are provided if the generalized stress point enters this region at any integration point.

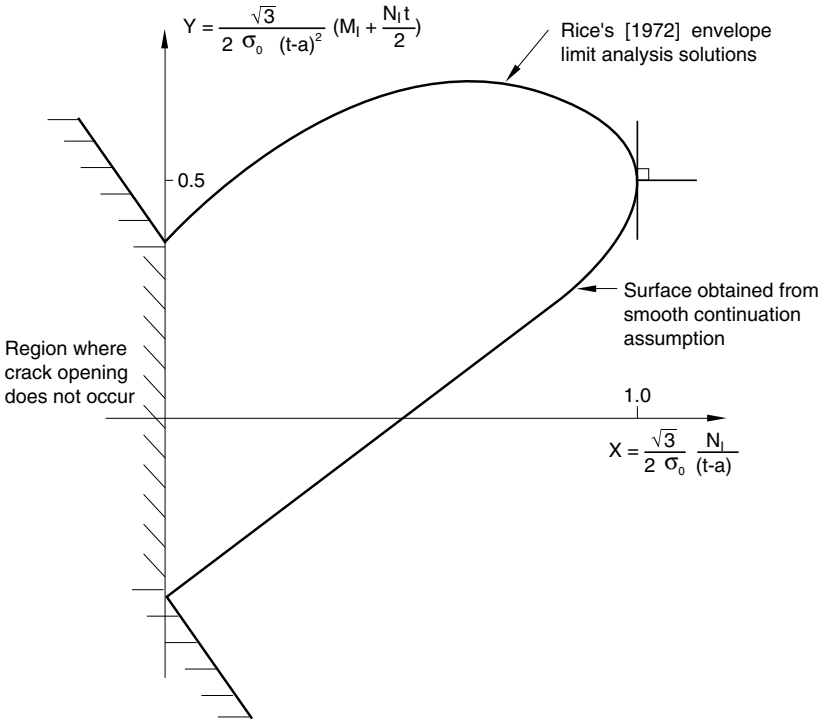


Figure 3.9.5–3 Generalized stress yield surface assumed for line springs.

The plasticity model is integrated by the usual backward Euler method (see "Integration of plasticity models," Section 4.2.2, for details). Once the plastic strain increments are known, from the kinematics of the slip line field proposed by Rice (1972) for an edge cracked strip, the increment of plastic crack-tip opening is given by

$$d\delta_{tip}^{pl} = dq_1^{pl} + (t/2 - a)dq_2^{pl}.$$

The increment in the plastic part of the ***J***-integral is related to the increment of plastic crack-tip opening by

$$dJ^{pl} = m\sigma_0 \, d\delta_{tip}^{pl},$$

where m is given by (Parks and White, 1982),

$$m = \frac{\Phi_{,a}}{\sigma_0\{\Phi_{,1} + (t/2 - a)\Phi_{,2}\}},$$

where $_{,a}$ indicates differentiation with respect to crack depth and $_{,1}$ and $_{,2}$ indicate differentiation by Q_1 and Q_2, respectively. Φ is either Φ_1 or Φ_2 depending on the state of deformation.

The elastic part of the ***J***-integral is obtained from the generalized stresses by calculating the stress intensity factors as described in the previous section (ignoring plasticity effects). The total ***J***-integral is the sum of the plastic and elastic ***J***-integral contributions. Although the method used for computing the elastic contribution is obviously approximate, it is reasonably accurate if J^{pl} dominates J^{el}, which is the case once a significant amount of plasticity develops (Parks and White, 1982).

Reference

- "Line spring elements for modeling part-through cracks in shells," Section 32.9.1 of the Abaqus Analysis User's Guide

3.9.6 FLEXIBLE JOINT ELEMENT

Product: Abaqus/Standard

The JOINTC elements in Abaqus/Standard provide for flexible joints between two nodes. This section defines the kinematic variables used in these elements.

Kinematics

A JOINTC element consists of two nodes, referred to here as nodes 1 and 2. Each node has six degrees of freedom: displacements $\mathbf{u}$ and rotations $\boldsymbol{\phi}$. A local orientation is defined for the element by the user. In a large-displacement analysis that local system rotates with the first node of the element.

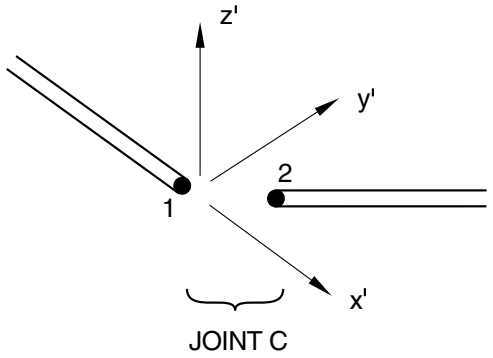


Figure 3.9.6–1 JOINTC geometry.

We define the local system by its unit, orthogonal base vectors, $\mathbf{e}_\alpha$, for $\alpha = 1, 2, 3$. Then at any time in the analysis

$$\mathbf{e}_\alpha = \mathbf{C} \cdot \mathbf{e}^0_\alpha,$$

where $\mathbf{C}(\boldsymbol{\phi}^1)$ is the rotation matrix defined by the rotation at the first node of the element.

The relative displacements in the element are then

$$u_\alpha = (\mathbf{u}^2 - \mathbf{u}^1) \cdot \mathbf{e}_\alpha,$$

with first variations

$$\delta u_\alpha = (\delta\mathbf{u}^2 - \delta\mathbf{u}^1) \cdot \mathbf{e}_\alpha + \delta\boldsymbol{\theta}^1 \cdot \mathbf{e}_\alpha \times (\mathbf{u}^2 - \mathbf{u}^1),$$

where $\delta\boldsymbol{\theta}^1$ is a linearized rotation field (see "Rotation variables," Section 1.3.1), and second variations

$$\begin{aligned} d\delta u_\alpha =& (\delta\mathbf{u}^2 - \delta\mathbf{u}^1)\cdot d\boldsymbol{\theta}^1 \times \mathbf{e}_\alpha + (d\mathbf{u}^2 - d\mathbf{u}^1)\cdot \delta\boldsymbol{\theta}^1 \times \mathbf{e}_\alpha \\ & - d\boldsymbol{\theta}^1 \cdot \delta\boldsymbol{\theta}^1\, u_\alpha + \frac{1}{2}\delta\boldsymbol{\theta}^1 \cdot (\mathbf{u}^2 - \mathbf{u}^1)\, d\boldsymbol{\theta}^1 \cdot \mathbf{e}_\alpha + \frac{1}{2}\delta\boldsymbol{\theta}^1 \cdot \mathbf{e}_\alpha\, d\boldsymbol{\theta}^1 \cdot (\mathbf{u}^2 - \mathbf{u}^1). \end{aligned}$$

The relative rotation about the local $\mathcal{3}$-axis is defined as

$$\psi_3 = \frac{1}{2}(\mathbf{e}_1^2 - \mathbf{e}_1^1)\cdot(\mathbf{e}_2^2 + \mathbf{e}_2^1),$$

with ψ_1 and ψ_2 defined by cyclic permutation of the local direction indices.

These rotation measures define only relative angular rotations for small relative rotations. They are simple to compute, increase monotonically for relative rotations up to 180°, and are taken as suitable for use in the elements for these reasons.

The first variation of ψ_3 is

$$\delta\psi_3 = \frac{1}{2}(\delta\boldsymbol{\theta}^2 \times \mathbf{e}_1^2 - \delta\boldsymbol{\theta}^1 \times \mathbf{e}_1^1)\cdot(\mathbf{e}_2^2 + \mathbf{e}_2^1) + \frac{1}{2}(\mathbf{e}_1^2 - \mathbf{e}_1^1)\cdot(\delta\boldsymbol{\theta}^2 \times \mathbf{e}_2^2 + \delta\boldsymbol{\theta}^1 \times \mathbf{e}_2^1),$$

and its second variation is

$$\begin{aligned} d\delta\psi_3 =& \frac{1}{4}(-2d\boldsymbol{\theta}^2 \cdot \delta\boldsymbol{\theta}^2\, \mathbf{e}_1^2 + d\boldsymbol{\theta}^2 \cdot \mathbf{e}_1^2\, \delta\boldsymbol{\theta}^2 + \delta\boldsymbol{\theta}^2 \cdot \mathbf{e}_1^2\, d\boldsymbol{\theta}^2 \\ & \quad + 2d\boldsymbol{\theta}^1 \cdot \delta\boldsymbol{\theta}^1\, \mathbf{e}_1^1 - d\boldsymbol{\theta}^1 \cdot \mathbf{e}_1^1\, \delta\boldsymbol{\theta}^1 - \delta\boldsymbol{\theta}^1 \cdot \mathbf{e}_1^1\, d\boldsymbol{\theta}^1)\cdot(\mathbf{e}_2^2 + \mathbf{e}_2^1) \\ & + \frac{1}{4}(\mathbf{e}_1^2 - \mathbf{e}_1^1)\cdot(-2d\boldsymbol{\theta}^2 \cdot \delta\boldsymbol{\theta}^2\, \mathbf{e}_2^2 + d\boldsymbol{\theta}^2 \cdot \mathbf{e}_2^2\, \delta\boldsymbol{\theta}^2 + \delta\boldsymbol{\theta}^2 \cdot \mathbf{e}_2^2\, d\boldsymbol{\theta}^2 \\ & \quad - 2d\boldsymbol{\theta}^1 \cdot \delta\boldsymbol{\theta}^1\, \mathbf{e}_2^1 + d\boldsymbol{\theta}^1 \cdot \mathbf{e}_2^1\, \delta\boldsymbol{\theta}^1 + \delta\boldsymbol{\theta}^1 \cdot \mathbf{e}_2^1\, d\boldsymbol{\theta}^1) \\ & + \frac{1}{2}(-\delta\boldsymbol{\theta}^2 \cdot \mathbf{e}_2^2\, d\boldsymbol{\theta}^2 \cdot \mathbf{e}_1^2 + \delta\boldsymbol{\theta}^1 \cdot \mathbf{e}_2^1\, d\boldsymbol{\theta}^1 \cdot \mathbf{e}_1^1 + \delta\boldsymbol{\theta}^1 \cdot d\boldsymbol{\theta}^2(\mathbf{e}_1^2 \cdot \mathbf{e}_2^1 - \mathbf{e}_1^1 \cdot \mathbf{e}_2^2)) \\ & + \frac{1}{2}(-d\boldsymbol{\theta}^2 \cdot \mathbf{e}_2^2\, \delta\boldsymbol{\theta}^2 \cdot \mathbf{e}_1^2 + d\boldsymbol{\theta}^1 \cdot \mathbf{e}_2^1\, \delta\boldsymbol{\theta}^1 \cdot \mathbf{e}_1^1 + d\boldsymbol{\theta}^1 \cdot \delta\boldsymbol{\theta}^2(\mathbf{e}_1^2 \cdot \mathbf{e}_2^1 - \mathbf{e}_1^1 \cdot \mathbf{e}_2^2)). \end{aligned}$$

The relative translational velocities in the element are taken as

$$\dot{u}_\alpha = (\dot{\mathbf{u}}^2 - \dot{\mathbf{u}}^1)\cdot\mathbf{e}_\alpha + \dot{\boldsymbol{\theta}}^1 \cdot \mathbf{e}_\alpha \times (\mathbf{u}^2 - \mathbf{u}^1),$$

and the relative angular velocity about the local $\mathcal{3}$-axis is taken as

$$\dot{\psi}_3 = \frac{1}{2}(\dot{\boldsymbol{\theta}}^2 \times \mathbf{e}_1^2 - \dot{\boldsymbol{\theta}}^1 \times \mathbf{e}_1^1)\cdot(\mathbf{e}_2^2 + \mathbf{e}_2^1) + \frac{1}{2}(\mathbf{e}_1^2 - \mathbf{e}_1^1)\cdot(\dot{\boldsymbol{\theta}}^2 \times \mathbf{e}_2^2 + \dot{\boldsymbol{\theta}}^1 \times \mathbf{e}_2^1).$$

Virtual work

The virtual work contribution of the element is

$$\delta W = F_\alpha \delta u_\alpha + M_\alpha \delta\psi_\alpha.$$

We assume that the behavior of the joint is defined by

$$F_\alpha = F_\alpha(u_\alpha, \dot{u}_\alpha) \quad \text{and} \quad M_\alpha = M_\alpha(\psi_\alpha, \dot{\psi}_\alpha) \quad \text{(no sum on } \alpha\text{)}.$$

The contribution to the operator matrix for the Newton solution is

$$d\delta W = \sum_\alpha \delta u_\alpha \left(\frac{\partial F_\alpha}{\partial u_\alpha} + \frac{\partial F_\alpha}{\partial \dot{u}_\alpha} \frac{\partial \dot{u}}{\partial u} \right) du_\alpha + \delta \psi_\alpha \left(\frac{\partial M_\alpha}{\partial \psi_\alpha} + \frac{\partial M_\alpha}{\partial \dot{\psi}_\alpha} \frac{\partial \dot{u}}{\partial u} \right) d\psi_\alpha + F_\alpha d\delta u_\alpha + M_\alpha d\delta \psi_\alpha,$$

where $\partial \dot{u}/\partial u$ is defined by the dynamic time integration operator.

Reference

- “Flexible joint elements,” Section 32.3 of the Abaqus Analysis User’s Guide

3.9.7 ROTARY INERTIA ELEMENT

Products: Abaqus/Standard Abaqus/Explicit

The MASS and ROTARYI elements allow the inertia of a rigid body to be introduced at a node. In this section the formulation used with these elements is defined.

It is assumed that the node at which the mass and rotary inertia are introduced is the center of mass of the body. We refer to the node as the rigid body reference node, C. Let the local principal axes of inertia of the body be $\mathbf{e}_\alpha$, $\alpha = 1, 2, 3$. Let $\mathbf{r}$ be the vector between C and some point in the rigid body with current coordinates $\mathbf{x}$, so that

$$\mathbf{r} = \mathbf{x} - \mathbf{x}_C = x_\alpha \mathbf{e}_\alpha, \text{ say},$$

where x_α are local coordinates in the rigid body. The mass of the rigid body is the integral of the mass density $\rho(x_\alpha)$ over the body

$$m = \int_V \rho \, dV.$$

Since C is assumed to be the center of mass of the body,

$$\int_V \rho x_\alpha \, dV = 0.$$

Since the $\mathbf{e}_\alpha$ are the principal axes of the body,

$$\int_V \rho x_\alpha x_\beta \, dV = 0 \quad \text{for} \quad \alpha \neq \beta.$$

Let I_{11}, I_{22}, and I_{33} be the second moments of inertia of the body about its principal axes $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$; then

$$I_{11} = \int_V \rho\big((x_2)^2 + (x_3)^2\big) \, dV$$
$$I_{22} = \int_V \rho\big((x_3)^2 + (x_1)^2\big) \, dV$$
$$I_{33} = \int_V \rho\big((x_1)^2 + (x_2)^2\big) \, dV.$$

The rotary inertia tensor is written

$$\mathbf{I} = \sum_{\alpha=1}^{3} I_{\alpha\alpha} \mathbf{e}_\alpha \mathbf{e}_\alpha \, .$$

For a rigid body the velocity of any point in the body is given by

$$\dot{\mathbf{u}} = \dot{\mathbf{u}}_C + \boldsymbol{\omega} \times \mathbf{r},$$

where $\boldsymbol{\omega} = \dot{\boldsymbol{\phi}}$ is the angular velocity of the body. Taking a second time derivative, the acceleration is

$$\ddot{\mathbf{u}} = \ddot{\mathbf{u}}_C + \dot{\boldsymbol{\omega}} \times \mathbf{r} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}).$$

The local or strong form of the equilibrium equations represents the balance of linear momentum and balance of angular momentum; these two equilibrium equations are

$$\begin{aligned} m\ddot{\mathbf{u}}_C &= \bar{\mathbf{f}}\,, \\ \mathbf{I}\cdot\dot{\boldsymbol{\omega}} + \boldsymbol{\omega}\times\mathbf{I}\cdot\boldsymbol{\omega} &= \bar{\mathbf{m}}\,. \end{aligned}$$

The variational or weak form of equilibrium is

$$\delta W_A + \delta W_{ext} = 0\,.$$

The internal or d'Alembert force contribution is

$$\delta W_A = -\int_V \rho\,\delta\mathbf{u}\cdot\ddot{\mathbf{u}}\,dV = -m\ddot{\mathbf{u}}_C\cdot\delta\mathbf{u}_C - (\mathbf{I}\cdot\dot{\boldsymbol{\omega}} + \boldsymbol{\omega}\times\mathbf{I}\cdot\boldsymbol{\omega})\cdot\delta\boldsymbol{\theta},$$

where $\delta\mathbf{u} = \delta\mathbf{u}_C + \delta\boldsymbol{\theta}\times\mathbf{r}$ is the variation of the position of a point in the body. Here $\delta\mathbf{u}_C$ is the variation of the position of the rigid body reference node, and $\delta\boldsymbol{\theta}$ is the variation of the rotation of the rigid body reference node. The external loading contribution is

$$\delta W_{ext} = \bar{\mathbf{f}}\cdot\delta\mathbf{u}_C + \bar{\mathbf{m}}\cdot\delta\boldsymbol{\theta}.$$

Introducing component expressions relative to the principal axes of inertia, the rotational contribution to the weak form becomes

$$\begin{aligned} \delta W_A = -m\,\delta\mathbf{u}_C\cdot\ddot{\mathbf{u}}_C &- \delta\theta^1\big(I_{11}\dot{\omega}^1 + (I_{33}-I_{22})\omega^2\omega^3\big) \\ &- \delta\theta^2\big(I_{22}\dot{\omega}^2 + (I_{11}-I_{33})\omega^3\omega^1\big) \\ &- \delta\theta^3\big(I_{33}\dot{\omega}^3 + (I_{22}-I_{11})\omega^1\omega^2\big). \end{aligned}$$

When the inertia of a rigid body is used with implicit time integration, the Jacobian contribution of δW_A is required: this is

$$\begin{aligned} -d\delta W_A = m\,\delta\mathbf{u}_C\cdot d\ddot{\mathbf{u}}_C + \delta\boldsymbol{\theta}\cdot\big[&\mathbf{I}\cdot d\dot{\boldsymbol{\omega}} + d\boldsymbol{\omega}\times\mathbf{I}\cdot\boldsymbol{\omega} + \boldsymbol{\omega}\times\mathbf{I}\cdot d\boldsymbol{\omega} + d\boldsymbol{\theta}\times\mathbf{I}\cdot\dot{\boldsymbol{\omega}} + \mathbf{I}\cdot(\dot{\boldsymbol{\omega}}\times d\boldsymbol{\theta}) \\ &+ (\boldsymbol{\omega}\cdot\mathbf{I}\cdot\boldsymbol{\omega})d\boldsymbol{\theta} - (\boldsymbol{\omega}\cdot d\boldsymbol{\theta})\mathbf{I}\cdot\boldsymbol{\omega} + \boldsymbol{\omega}\times\mathbf{I}\cdot(\boldsymbol{\omega}\times d\boldsymbol{\theta})\big]. \end{aligned}$$

Reference

- "Rotary inertia," Section 30.2.1 of the Abaqus Analysis User's Guide

3.9.8 DISTRIBUTING COUPLING ELEMENTS

Products: Abaqus/Standard Abaqus/Explicit

The distributing coupling elements provide a means to connect a reference node to a group of coupling nodes in a way that distributes loads according to weight factors that are prescribed individually at each coupling node. The element distributes forces and moments at the reference node as a coupling node–force distribution only. The formulation presented here is also used for the surface-based distributing coupling and in fastener constraints. This section defines this load distribution relationship and the resultant element development for the case when the default continuum coupling method is used.

The reference node has displacement ($\mathbf{u}^R$) and rotation ($\boldsymbol{\phi}^R$) degrees of freedom. The coupling nodes have only displacement ($\mathbf{u}^n$) degrees of freedom active in this element. Each coupling node has a weight factor w^n assigned, which determines the proportion of load carried by the element that is transmitted through the coupling node. Weight factors are dimensionless, and their magnitude is significant only in a relative sense. Hereafter, normalized weights are used:

$$\hat{w}^n = \frac{w^n}{\sum w^n}.$$

Load distribution

Let $\mathbf{F}^R$ and $\mathbf{M}^R$ be the load and moment applied to the reference node. The statically admissible force distribution $\mathbf{F}^n$ among the coupling nodes satisfies

$$\begin{aligned} \sum_n \mathbf{F}^n &= \mathbf{F}^R \\ \sum_n \mathbf{x}^n \times \mathbf{F}^n &= \mathbf{M}^R + \mathbf{x}^R \times \mathbf{F}^R, \end{aligned} \tag{3.9.8–1}$$

where $\mathbf{x}^R$ and $\mathbf{x}^n$ are the positions of the reference and coupling nodes, respectively. For an arbitrary number of coupling nodes there is no unique solution to Equation 3.9.8–1.

The force distribution adopted in Abaqus has the property that the linearized motion of the reference node is compatible with the coupling node group motion in an average sense. This compatibility can be described by considering the momentum of a moving coupling node group in a case where weight factors are considered as masses. In this case the reference node motion is identical to that of a point on a rigid body occupying the position of the reference node, where the center of mass of the rigid body is the center of mass of the coupling nodes and the rigid body moves with the same linear and angular momentum as the coupling node group. Since the element mass is distributed this way, the dynamic behavior of the element also has this property.

$$\mathbf{F}^n \stackrel{\text{def}}{=} \hat{w}^n \left(\mathbf{F}^R + \left(\mathbf{T}^{-1} \cdot \hat{\mathbf{M}}^R \right) \times \mathbf{r}^n \right),$$

where

$$\hat{\mathbf{M}}^R = \mathbf{M}^R + \mathbf{r}^R \times \mathbf{F}^R,$$
$$\mathbf{r}^n = \mathbf{x}^n - \bar{\mathbf{x}},$$
$$\bar{\mathbf{x}} = \frac{\sum_n w^n \mathbf{x}^n}{\sum_n w^n} = \sum_n \hat{w}^n \mathbf{x}^n,$$

and the coupling node arrangement inertia tensor is

$$\mathbf{T} = \sum_n \hat{w}^n \left[(\mathbf{r}^n \cdot \mathbf{r}^n)\, \mathbf{I} - (\mathbf{r}^n \mathbf{r}^n) \right],$$

where $\mathbf{I}$ is the second-order identity tensor. This force distribution is recognized to be equivalent to the classic bolt-pattern force distribution when the weight factors are interpreted as bolt cross-section areas.

Constraint expression

The load distribution results in the following linearized constraint on node motions:

$$\delta \mathbf{x}^R = \sum_n \hat{w}^n \delta \mathbf{x}^n + \left(\mathbf{T}^{-1} \cdot \sum_n \hat{w}^n \left(\mathbf{r}^n \times \delta \mathbf{x}^n \right) \right) \times \mathbf{r}^R$$
$$\delta \boldsymbol{\omega}^R = \mathbf{T}^{-1} \cdot \sum_n \hat{w}^n \left(\mathbf{r}^n \times \delta \mathbf{x}^n \right),$$

where $\mathbf{r}^R = \mathbf{x}^R - \bar{\mathbf{x}}$.

Finite motion

Finite displacement and rotation terms take the form of a constraint on the motion of the reference node as a function of the coupling-node finite incremental motions. A measure of the finite rotation of the coupling node arrangement is developed first and is based on the mid-increment position of the coupling nodes, defined as

$$\check{\mathbf{r}}^n \stackrel{\text{def}}{=} \mathbf{r}_0^n + \frac{1}{2} \Delta \mathbf{x}^n,$$

and the mid-increment inertia tensor is

$$\mathbf{J} = \sum_n \hat{w}^n \left[(\check{\mathbf{r}}^n \cdot \check{\mathbf{r}}^n) \mathbf{I} - (\check{\mathbf{r}}^n \check{\mathbf{r}}^n) \right].$$

The mid-increment "spin" is then

$$\Delta \boldsymbol{\theta}^R = \mathbf{J}^{-1} \cdot \left(\sum_n \hat{w}^n \check{\mathbf{r}}^n \times \Delta \mathbf{x}^n \right).$$

The finite incremental rotation tensor $\Delta\mathbf{R}$ is deduced from the above expression according to the Hughes and Winget (1980) formula,

$$\Delta\mathbf{R} = \left(\mathbf{I} - \frac{1}{2}\widehat{\Delta\boldsymbol{\theta}}^{R}\right)^{-1} \cdot \left(\mathbf{I} + \frac{1}{2}\widehat{\Delta\boldsymbol{\theta}}^{R}\right).$$

This orthogonal tensor yields an incremental finite-rotation vector $\Delta\acute{\boldsymbol{\phi}}^{R}$. From this rotation description comes the constraint expressions for finite displacement and rotation:

$$\begin{aligned}\Delta\mathbf{x}^{R} &= \sum_{n} \hat{w}^{n}\Delta\mathbf{x}^{n} + \Delta\mathbf{R}\cdot\mathbf{r}_{0}^{R} - \mathbf{r}_{0}^{R},\\ \Delta\boldsymbol{\phi}^{R} &= \Delta\acute{\boldsymbol{\phi}}^{R}.\end{aligned}$$

Virtual work contribution

The virtual work expression for the attached structure is augmented with the contribution of the constraint

$$\begin{aligned}\delta\Pi^{*} = \delta\Pi \;+\; &\boldsymbol{\lambda}_{f}\cdot\left[\delta\mathbf{x}^{R} - \sum_{n}\hat{w}^{n}\delta\mathbf{x}^{n} - \delta\mathbf{R}^{R}\cdot\mathbf{r}_{0}^{R}\right] +\\ &\boldsymbol{\lambda}_{m}\cdot\left[\delta\boldsymbol{\omega}^{R} - \delta\acute{\boldsymbol{\phi}}^{R}\right] +\\ &\delta\boldsymbol{\lambda}_{f}\cdot\left[\Delta\mathbf{x}^{R} - \sum_{n}\hat{w}^{n}\Delta\mathbf{x}^{n} - \Delta\mathbf{R}\cdot\mathbf{r}_{0}^{R} + \mathbf{r}_{0}^{R}\right] +\\ &\delta\boldsymbol{\lambda}_{m}\cdot\left[\Delta\boldsymbol{\phi}^{R} - \Delta\acute{\boldsymbol{\phi}}^{R}\right],\end{aligned} \tag{3.9.8–2}$$

where $\delta\Pi^{*}$ is the augmented virtual work expression, $\delta\Pi$ is the virtual work expression for the attached structure, and $\boldsymbol{\lambda}_{f}$ and $\boldsymbol{\lambda}_{m}$ are the respective Lagrange multiplier variables for force and moment.

Initial stress stiffness terms

The initial stress stiffness terms are derived from a suitable approximation of the exact virtual work expression shown in Equation 3.9.8–2. This approximation is based on an assumption of infinitesimal incremental motions, $\Delta\mathbf{r}^{n}$ and $\Delta\boldsymbol{\phi}^{R}$, that implies

$$\begin{aligned}\delta\boldsymbol{\phi}^{R} &= \delta\boldsymbol{\omega}^{R} \quad \text{and}\\ \delta\acute{\boldsymbol{\phi}}^{R} &= \mathbf{T}^{-1}\cdot\sum_{n}\hat{w}^{n}\left(\mathbf{r}^{n}\times\delta\mathbf{x}^{n}\right).\end{aligned}$$

An approximate virtual work expression is obtained:

$$\widetilde{\delta\Pi}^{*} = \delta\Pi + \boldsymbol{\lambda}_f \cdot \left[\delta\mathbf{x}^R - \sum_n \hat{w}^n \delta\mathbf{x}^n - \delta\boldsymbol{\omega}^R \times \mathbf{r}^R\right] + \boldsymbol{\lambda}_m \cdot \left[\delta\boldsymbol{\omega}^R - \mathbf{T}^{-1} \cdot \sum_n \hat{w}^n \left(\mathbf{r}^n \times \delta\mathbf{x}^n\right)\right] +$$

$$\delta\boldsymbol{\lambda}_f \cdot \left[\Delta\mathbf{x}^R - \sum_n \hat{w}^n \Delta\mathbf{x}^n - \Delta\mathbf{R} \cdot \mathbf{r}_0^R + \mathbf{r}_0^R\right] + \delta\boldsymbol{\lambda}_m \cdot \left[\Delta\boldsymbol{\phi}^R - \Delta\acute{\boldsymbol{\phi}}^R\right].$$

This expression yields the following initial stress stiffness terms:

$$\boldsymbol{\lambda}_f \cdot \left[\left(\delta\boldsymbol{\omega}^R \cdot \mathrm{d}\boldsymbol{\omega}^R\right)\mathbf{r}^R - \frac{1}{2}\left(\delta\boldsymbol{\omega}^R \cdot \mathbf{r}^R\right)\mathrm{d}\boldsymbol{\omega}^R - \frac{1}{2}\left(\mathrm{d}\boldsymbol{\omega}^R \cdot \mathbf{r}^R\right)\delta\boldsymbol{\omega}^R\right] +$$

$$\boldsymbol{\lambda}_m \cdot \mathbf{T}^{-1} \cdot \left[\delta\boldsymbol{\omega}^R \sum_n \hat{w}^n \mathbf{r}^n \cdot \mathrm{d}\mathbf{r}^n - \frac{1}{2}\sum_n \hat{w}^n \mathrm{d}\mathbf{r}^n \mathbf{r}^n \cdot \delta\boldsymbol{\omega}^R - \frac{1}{2}\sum_n \hat{w}^n \mathbf{r}^n \mathrm{d}r^n \cdot \delta\boldsymbol{\omega}^R + \mathrm{d}\boldsymbol{\omega}^R \sum_n \hat{w}^n \mathbf{r}^n \cdot \delta\mathbf{r}^n - \frac{1}{2}\sum_n \hat{w}^n \delta\mathbf{r}^n \mathbf{r}^n \cdot \mathrm{d}\boldsymbol{\omega}^R - \frac{1}{2}\sum_n \hat{w}^n \mathbf{r}^n \delta r^n \cdot \mathrm{d}\boldsymbol{\omega}^R\right].$$

Mass

The distributing coupling elements also distribute masses to each coupling node according to the weight distribution. A prescribed element mass of M is distributed to the cloud nodes according to

$$m^n = \hat{w}^n M,$$

where m^n is the cloud node mass. Masses are distributed only to the cloud nodes; no mass is associated with the reference node.

References

- "Distributing coupling elements," Section 32.4 of the Abaqus Analysis User's Guide
- "Coupling constraints," Section 35.3.2 of the Abaqus Analysis User's Guide

4. Mechanical Constitutive Theories

Overview 4.1
Plasticity overview 4.2
Metal plasticity 4.3
Plasticity for non-metals 4.4
Other inelastic models 4.5
Large-strain elasticity 4.6
Mullins effect and permanent set 4.7
Viscoelasticity 4.8
Hysteresis 4.9

4.1 Overview

- “Mechanical constitutive models,” Section 4.1.1

4.1.1 MECHANICAL CONSTITUTIVE MODELS

Products: Abaqus/Standard Abaqus/Explicit

A wide variety of materials is encountered in stress analysis problems, and for any one of these materials a range of constitutive models is available to describe the material's behavior. For example, a component made from a standard structural steel can be modeled as an isotropic, linear elastic, material with no temperature dependence. This simple model would probably suffice for routine design, so long as the component is not in any critical situation. However, if the component might be subjected to a severe overload, it is important to determine how it might deform under that load and if it has sufficient ductility to withstand the overload without catastrophic failure. The first of these questions might be answered by modeling the material as a rate-independent elastic, perfectly plastic material, or—if the ultimate stress in a tension test of a specimen of the material is very much above the initial yield stress—isotropic work hardening might be included in the plasticity model. A nonlinear analysis (with or without consideration of geometric nonlinearity, depending on whether the analyst judges that the structure might buckle or undergo large geometry changes during the event) is then done to determine the response. But the severe overload might be applied suddenly, thus causing rapid straining of the material. In such circumstances the inelastic response of metals usually exhibits rate dependence: the flow stress increases as the strain rate increases. A "viscoplastic" (rate-dependent) material model might, therefore, be required. (Arguing that it is conservative to ignore this effect because it is a strengthening effect is not necessarily acceptable—the strengthening of one part of a structure might cause load to be shed to another part, which proves to be weaker in the event.) So far the analyst can manage with relatively simple (but nonlinear) constitutive models. But if the failure is associated with localization—tearing of a sheet of material or plastic buckling—a more sophisticated material model might be required because such localizations depend on details of the constitutive behavior that are usually ignored because of their complexity (see, for example, Needleman, 1977). Or if the concern is not gross overload, but gradual failure of the component because of creep at high temperature or because of low-cycle fatigue, or perhaps a combination of these effects, then the response of the material during several cycles of loading, in each of which a small amount of inelastic deformation might occur, must be predicted: a circumstance in which we need to model much more of the detail of the material's response.

So far the discussion has considered a conventional structural material. We can broadly classify the materials of interest as those that exhibit almost purely elastic response, possibly with some energy dissipation during rapid loading by viscoelastic response (the elastomers, such as rubber or solid propellant); materials that yield and exhibit considerable ductility beyond yield (such as mild steel and other commonly used metals, ice at low strain rates, and clay); materials that flow by rearrangement of particles that interact generally through some dominantly frictional mechanism (such as sand); and brittle materials (rocks, concrete, ceramics). The constitutive library provided in Abaqus contains a range of linear and nonlinear material models for all of these categories of materials. In general the library has been developed to provide those models that are most usually required for practical applications. There are several distinct models in the library; and for the more commonly encountered materials (metals, in particular), several ways of modeling the material are provided, each suitable to a particular type of analysis application. But the library is far from comprehensive: the range of physical material behavior is far too broad for this ever to be possible. The analyst must review the

material definitions provided in Abaqus in the context of each particular application. If there is no model in the library that is useful for a particular case, Abaqus/Standard contains a user subroutine—**UMAT**—and, similarly, Abaqus/Explicit contains a user subroutine—**VUMAT**. In these routines the user can code a material model (or call other routines that perform that task). This "user subroutine" capability is a powerful resource for the sophisticated analyst who is able to cope with the demands of programming a complex material model.

Theoretical aspects of the material models that are provided in Abaqus are described in this chapter, which is intended as a definition of the details of the material models that are provided: it also provides useful guidance to analysts who might have to code their own models in **UMAT** or **VUMAT**.

From a numerical viewpoint the implementation of a constitutive model involves the integration of the state of the material at an integration point over a time increment during a nonlinear analysis. (The implementation of constitutive models in Abaqus assumes that the material behavior is entirely defined by local effects, so each spatial integration point can be treated independently.) Since Abaqus/Standard is most commonly used with implicit time integration, the implementation must also provide an accurate "material stiffness matrix" for use in forming the Jacobian of the nonlinear equilibrium equations; this is not necessary for Abaqus/Explicit.

The mechanical constitutive models that are provided in Abaqus often consider elastic and inelastic response. The inelastic response is most commonly modeled with plasticity models. Several plasticity models are described in this chapter. Some of the constitutive models in Abaqus also use damage mechanics concepts, the distinction being that in plasticity theory the elasticity is not affected by the inelastic deformation (the Young's modulus of a metal specimen is not changed by loading it beyond yield, until the specimen is very close to failure), while damage models include the degradation of the elasticity caused by severe loading (such as the loss of elastic stiffness suffered by a concrete specimen after it has been subjected to large uniaxial compressive loading).

In the inelastic response models that are provided in Abaqus, the elastic and inelastic responses are distinguished by separating the deformation into recoverable (elastic) and nonrecoverable (inelastic) parts. This separation is based on the assumption that there is an additive relationship between strain rates:

$$\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}}^{el} + \dot{\boldsymbol{\varepsilon}}^{pl}, \tag{4.1.1–1}$$

where $\dot{\boldsymbol{\varepsilon}}$ is the total strain rate, $\dot{\boldsymbol{\varepsilon}}^{el}$ is the rate of change of the elastic strain, and $\dot{\boldsymbol{\varepsilon}}^{pl}$ is the rate of change of inelastic strain.

A more general assumption is that the total deformation, $\mathbf{F}$, is made up of inelastic deformation followed by purely elastic deformation (with the rigid body rotation added in at any stage in the process):

$$\mathbf{F} = \mathbf{F}^{el} \cdot \mathbf{F}^{pl}. \tag{4.1.1–2}$$

In "The additive strain rate decomposition," Section 1.4.4, the circumstances are discussed under which Equation 4.1.1–1 is a legitimate approximation to Equation 4.1.1–2. We conclude that, if

1. the total strain rate measure used in Equation 4.1.1–1 is the rate of deformation:

$$\dot{\boldsymbol{\varepsilon}} = \mathrm{sym}(\mathbf{L}) = \mathrm{sym}\left(\dot{\mathbf{F}} \cdot \mathbf{F}^{-1}\right) = \mathrm{sym}\left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right),$$

where $\mathbf{v}$ is the velocity and $\mathbf{x}$ is the current spatial position of a material point; and

2. the elastic strains are small,

then the approximation is consistent. Abaqus uses the rate of deformation as the strain rate measure in finite-strain problems for this reason. (The distinction between different strain measures matters only when the strains are not negligible compared to unity; that is, in finite-strain problems.) The elastic strains always remain small for many materials of practical interest; for example, the yield stress of a metal is typically three orders of magnitude smaller than its elastic modulus, implying elastic strains of order 10^{-3}. However, some materials (polymers, for example) can undergo large elastic straining and also flow inelastically, in which case the additive strain rate decomposition is no longer a consistent approximation.

Various elastic response models are provided in Abaqus. The simplest of these is linear elasticity:

$$\boldsymbol{\sigma} = \mathbf{D}^{el} : \boldsymbol{\varepsilon}^{el},$$

where $\mathbf{D}^{el}$ is a matrix that may depend on temperature but does not depend on the deformation (except when such dependency is introduced in damage models). This elasticity model is intended to be used for small-strain problems or to model the elasticity in an elastic-plastic model in which the elastic strains are always small.

An extension of the elastic type of behavior is the hypoelastic model:

$$\dot{\boldsymbol{\sigma}} = \mathbf{D}^{el} : \dot{\boldsymbol{\varepsilon}}^{el},$$

where now $\mathbf{D}^{el}$ may depend on the deformation. In this case the elasticity may be nonlinear, but the implementation in Abaqus still assumes that the elastic strains will always be small. In porous and granular media, the elastic properties are strongly dependent on the volume strain; porous elastic behavior is described in "Porous elasticity," Section 4.4.1.

The most general type of nonlinear elastic behavior is the hyperelastic model, in which we assume that there is a strain energy density potential—U—from which the stresses are defined (to within a hydrostatic stress value if the material is fully incompressible) by

$$\boldsymbol{\sigma} = \frac{\partial U}{\partial \boldsymbol{\varepsilon}},$$

where $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ are any work conjugate stress and strain measures. This form of elasticity model is generally used to model elastomers: materials whose long-term response to large deformations is fully recoverable (elastic). The hyperelasticity modeling provided in Abaqus is described in "Large-strain elasticity," Section 4.6. The hyperelasticity models cannot be used with the plastic deformation models in the program, but can be combined with viscoelastic behavior, as described in "Finite-strain viscoelasticity," Section 4.8.2.

The plasticity models offered in Abaqus are discussed in general terms in "Plasticity overview," Section 4.2. Both rate-independent and rate-dependent models, with and without yield surfaces, are offered. Models are included in the program that are intended for applications to metals ("Metal plasticity," Section 4.3) as well as some nonmetallic materials such as soils, polymers, and crushable foams ("Plasticity for non-metals," Section 4.4). The jointed material model ("Constitutive model for jointed materials," Section 4.5.4) and the concrete model ("An inelastic constitutive model for concrete," Section 4.5.1) also include plasticity modeling.

The constitutive routines in Abaqus exist in a library that can be accessed by any of the solid or structural elements. This access is made independently at each "constitutive calculation point." These points are the numerical integration points in the elements. Thus, the constitutive routines are concerned only with a single calculation point. The element provides an estimate of the kinematic solution to the problem at the point under consideration. These kinematic data are passed to the constitutive routines as the deformation gradient—$\mathbf{F}$—or, more typically, as the strain and rotation increments—$\Delta\boldsymbol{\varepsilon}$ and $\Delta\mathbf{R}$. The constitutive routines obtain the state at the point under consideration at the start of the increment from the "material point data base." The state variables include the stress and any state variables used in the constitutive models—plastic strains, for example. The constitutive routines also look up the constitutive definition. Their function then is to update the state to the end of the increment and, if the procedure uses implicit time integration and if Newton's method is being used to solve the equations, to define the material contribution to the Jacobian matrix, $\partial\boldsymbol{\sigma}/\partial\boldsymbol{\varepsilon}$. For material models that are defined in rate form and, therefore, require integration (such as incremental plasticity models), this Jacobian contribution depends on the model and also on the integration method used for the model. Its derivation is, therefore, discussed in some detail in the sections that define such models.

Reference

- "Material library: overview," Section 21.1.1 of the Abaqus Analysis User's Guide

4.2 Plasticity overview

- "Plasticity models: general discussion," Section 4.2.1
- "Integration of plasticity models," Section 4.2.2

4.2.1 PLASTICITY MODELS: GENERAL DISCUSSION

Products: Abaqus/Standard Abaqus/Explicit

Incremental plasticity theory is based on a few fundamental postulates, which means that all of the elastic-plastic response models provided in Abaqus (except the deformation theory model in Abaqus/Standard, which is primarily provided for fracture mechanics applications) have the same general form. The basic equations of the models are defined in their general form in this section.

Plasticity models are written as rate-independent models or as rate-dependent models. A rate-independent model is one in which the constitutive response does not depend on the rate of deformation—the response of many metals at low temperatures relative to their melting temperature and at low strain rates is effectively rate independent. In a rate-dependent model the response does depend on the rate at which the material is strained. Examples of such models are the simple metal "creep" models provided in Abaqus/Standard and the rate-dependent plasticity model that is used to describe the behavior of metals at higher strain rates. Because these models have similar forms, their numerical treatment is based on the same technique.

A basic assumption of elastic-plastic models is that the deformation can be divided into an elastic part and an inelastic (plastic) part. In its most general form this statement is written as

$$\mathbf{F} = \mathbf{F}^{el} \cdot \mathbf{F}^{pl}, \tag{4.2.1–1}$$

where $\mathbf{F}$ is the total deformation gradient, $\mathbf{F}^{el}$ is the fully recoverable part of the deformation at the point under consideration ($[\mathbf{F}^{el}]^{-1}$ is the deformation that would occur if, after the deformation $\mathbf{F}$, inelastic response were somehow prevented but at the same time the stress at the point reduced to zero), and $\mathbf{F}^{pl}$ is defined by $\mathbf{F}^{pl} = [\mathbf{F}^{el}]^{-1} \cdot \mathbf{F}$. The rigid body rotation at the point can be included in the definition of either $\mathbf{F}^{el}$ or $\mathbf{F}^{pl}$ or can be considered separately before or after either part of the decomposition: this makes no difference except in the convenience of the basis for writing the parts of the deformation.

This decomposition can be used directly to formulate the plasticity model. Historically, an additive strain rate decomposition,

$$\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}}^{el} + \dot{\boldsymbol{\varepsilon}}^{pl}, \tag{4.2.1–2}$$

has been used in its place. Here $\dot{\boldsymbol{\varepsilon}}$ is the total (mechanical) strain rate, $\dot{\boldsymbol{\varepsilon}}^{el}$ is the elastic strain rate, and $\dot{\boldsymbol{\varepsilon}}^{pl}$ is the plastic strain rate.

It is shown in "The additive strain rate decomposition," Section 1.4.4, that Equation 4.2.1–2 is a consistent approximation to Equation 4.2.1–1 when the elastic strains are infinitesimal (negligible compared to unity) and when the strain rate measure used in Equation 4.2.1–2 is the rate of deformation:

$$\dot{\boldsymbol{\varepsilon}} = \operatorname{sym}\left[\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right].$$

Equation 4.2.1–2, with the rate of deformation used as the definition of total strain rate, is used in all of the plasticity models that are implemented in Abaqus. Rice's argument implies that the elastic response must always be small in problems in which these models are used. In practice this is the case: plasticity models are provided for metals, soils, polymers, crushable foams, and concrete; and in each of these materials it is very unlikely that the elastic strain would ever be larger than a few percent (and even this would be quite unusual in a metal). Thus, the use of Equation 4.2.1–2 does not appear to be objectionable for the models in question, at least from a formal point of view. However, the user who needs to develop user subroutine **UMAT** or **VUMAT** for a different material model in which the elastic strains and the inelastic strains may both be large should consider using Equation 4.2.1–1 directly.

The elastic part of the response is assumed to be derivable from an elastic strain energy density potential, so the stress is defined by

$$\boldsymbol{\sigma} = \frac{\partial U}{\partial \boldsymbol{\varepsilon}^{el}}, \tag{4.2.1–3}$$

where U is the strain energy density potential. Since we assume that, in the absence of plastic straining, the variation of elastic strain is the same as the variation in the rate of deformation, conjugacy arguments define the stress measure $\boldsymbol{\sigma}$ as the "true" (Cauchy) stress. All stress output in Abaqus is given in this form.

In some materials the elastic response is essentially incompressible. But this is not usually the case for the materials whose inelastic deformation is considered with the models provided in Abaqus, so Equation 4.2.1–3 can be taken to define the stress completely. However, the inelastic response is often assumed to be approximately incompressible (in metals, for example, or in soils undergoing large plastic flows), so the user must be careful to ensure that the elements chosen can accommodate incompressible response without exhibiting "locking" problems when the model is three-dimensional, plane strain, or axisymmetric. This requires the use of hybrid or fully or selectively reduced-integration elements.

For several of the plasticity models provided in Abaqus the elasticity is linear, so the strain energy density potential has a very simple form. For the soils model the volumetric elastic strain is proportional to the logarithm of the equivalent pressure stress. For the concrete model damaged elasticity is used to account for the crack opening after the concrete has cracked: in that case the elasticity model is more complex.

The rate-independent plasticity models in Abaqus and one of the rate-dependent models all have a region of purely elastic response. The yield function, f, defines the limit to this region of purely elastic response and is written so that

$$f(\boldsymbol{\sigma}, \theta, H_\alpha) < 0 \tag{4.2.1–4}$$

for purely elastic response. Here θ is the temperature, and H_α are a set of hardening parameters. The subscript α is introduced simply to indicate that there may be several hardening parameters, H_α: the range of α is not specified until we define a particular plasticity model. The hardening parameters are state variables that are introduced to allow the models to describe some of the complexity of the inelastic response of real materials. In the simplest plasticity model ("perfect plasticity") the yield surface acts as a limit surface and there are no hardening parameters at all: no part of the model evolves during the deformation. Complex plasticity models usually include a large number of hardening parameters. The models provided in Abaqus are generally not

the most complex models and use only a few such parameters (only one is used in the isotropic hardening metal model and in the Cam-clay model; six are used in the simple kinematic hardening model).

In the concrete and the jointed material plasticity models in Abaqus the yield behavior is modeled with several independent inelastic flow systems. For this case Equation 4.2.1–4 is generalized to

$$f_i < 0$$

for purely elastic response in system i, where $f_i(\boldsymbol{\sigma}, H_{i,\alpha})$ is one of the yield functions and $H_{i,\alpha}$ are the hardening parameters for the ith system. For generality in this discussion we assume the model uses such a system of independent yield functions. In the simpler systems with a single yield function i can only take the value 1.

Stress states that cause the yield function to have a positive value cannot occur in rate-independent plasticity models, although this is possible in a rate-dependent model. Thus, in the rate-independent models we have the yield constraints

$$f_i = 0$$

during inelastic flow.

When the material is flowing inelastically the inelastic part of the deformation is defined by the flow rule, which we can write as

$$d\boldsymbol{\varepsilon}^{pl} = \sum_i d\lambda_i \frac{\partial g_i}{\partial \boldsymbol{\sigma}}, \tag{4.2.1–5}$$

where $g_i(\boldsymbol{\sigma}, \theta, H_{i,\alpha})$ is the flow potential for the ith system and $d\lambda_i$ is the rate of change of time, dt, for a rate-dependent model or is a scalar measuring the amount of the plastic flow rate on the ith system, whose value is determined by the requirement to satisfy the consistency condition $f_i = 0$, for plastic flow of a rate-independent model. The summation is over only the actively yielding systems: $d\lambda_i = 0$ for those systems for which $f_i < 0$.

The form in which the flow rule is written above assumes that there is a single flow potential, g_i, in the ith system. More general plasticity models might have several active flow potentials at a point. This is, for instance, the case in the concrete and jointed material models built into Abaqus.

For some rate-independent plasticity models the direction of flow is the same as the direction of the outward normal to the yield surface:

$$\frac{\partial g_i}{\partial \boldsymbol{\sigma}} = c_i \frac{\partial f_i}{\partial \boldsymbol{\sigma}},$$

where c_i is a scalar. Such models are called "associated flow" plasticity models. Associated flow models are useful for materials in which dislocation motion provides the fundamental mechanisms of plastic flow when there are no sudden changes in the direction of the plastic strain rate at a point. They are generally not accurate for materials in which the inelastic deformation is primarily caused by frictional mechanisms. The metal plasticity models in Abaqus (except cast iron) and the Cam-clay soil model use associated flow. The cast iron, granular/polymer, crushable foam, Mohr-Coulomb, Drucker-Prager/Cap, and jointed material

models provide nonassociated flow with respect to volumetric straining and equivalent pressure stress. The concrete model uses associated flow.

The rate form of the flow rule is essential to incremental plasticity theory, because it allows the history dependence of the response to be modeled.

The final ingredient in plasticity models is the set of evolution equations for the hardening parameters. We write these equations as

$$dH_{i,\alpha} = d\lambda_i \, h_{i,\alpha}(\boldsymbol{\sigma}, \theta, H_{i,\beta}), \tag{4.2.1–6}$$

where $h_{i,\alpha}$ is the (rate form) hardening law for $H_{i,\alpha}$. In complex plasticity models—for example, models used to describe the cyclic behavior of metals used for high temperature applications—these evolution laws have complicated forms, since such complexity is required to match the experimentally observed behavior. The plasticity models offered in Abaqus use simple evolution equations: isotropic hardening, Prager-Ziegler kinematic hardening, and the location of the center of the yield surface along the equivalent pressure stress axis in the Cam-clay model. The independence of the yield systems designated by the subscript i is implicit in the assumption in Equation 4.2.1–6 above that the evolution of the $H_{i,\alpha}$ depends only on other hardening parameters, $H_{i,\beta}$, in the same (ith) system.

Equation 4.2.1–1 to Equation 4.2.1–6 define the general structure of all of the plasticity models in Abaqus. Since the flow rule and the hardening evolution rules are written in rate form, they must be integrated. The general technique of integration is discussed in "Integration of plasticity models," Section 4.2.2. The sections immediately following that discussion describe the details of the specific plasticity models that are provided in Abaqus.

Reference

- "Inelastic behavior," Section 23.1.1 of the Abaqus Analysis User's Guide

4.2.2 INTEGRATION OF PLASTICITY MODELS

Products: Abaqus/Standard Abaqus/Explicit

The plasticity models provided in Abaqus have been described in general terms in "Plasticity models: general discussion," Section 4.2.1. The only rate equations are the evolutionary rule for the hardening, the flow rule, and the strain rate decomposition. The simplest operator that provides unconditional stability for integration of rate equations is the backward Euler method: applying this method to the flow rule (Equation 4.2.1–5) gives

$$\Delta\boldsymbol{\varepsilon}^{pl} = \sum_i \Delta\lambda_i \frac{\partial g_i}{\partial \boldsymbol{\sigma}}, \tag{4.2.2–1}$$

and applying it to the hardening evolution equations, Equation 4.2.1–6, gives

$$\Delta H_{i,\alpha} = \Delta\lambda_i h_{i,\alpha}. \tag{4.2.2–2}$$

In these equations and throughout the remainder of this section any quantity not specifically associated with a time point is taken at the end of the increment (at time $t + \Delta t$). The strain rate decomposition, Equation 4.2.1–2, is integrated over a time increment as

$$\Delta\boldsymbol{\varepsilon} = \Delta\boldsymbol{\varepsilon}^{el} + \Delta\boldsymbol{\varepsilon}^{pl},$$

where $\Delta\boldsymbol{\varepsilon}$ is defined by the central difference operator:

$$\Delta\boldsymbol{\varepsilon} = \operatorname{sym}\left[\frac{\partial \Delta\mathbf{x}}{\partial(\mathbf{x}_t + \frac{1}{2}\Delta\mathbf{x})}\right].$$

We integrate the total values of each strain measure as the sum of the value of that strain at the start of the increment, rotated to account for rigid body motion during the increment, and the strain increment. The rotation to account for rigid body motion during the increment is defined approximately using the algorithm of Hughes and Winget (1980). This integration allows the strain rate decomposition to be integrated into

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{el} + \boldsymbol{\varepsilon}^{pl}. \tag{4.2.2–3}$$

From a computational viewpoint the problem is now algebraic: we must solve the integrated equations of the constitutive model for the state at the end of the increment. The set of equations that define the algebraic problem are the strain decomposition, Equation 4.2.2–3; the elasticity, Equation 4.2.1–3; the integrated flow rule, Equation 4.2.2–1; the integrated hardening laws, Equation 4.2.2–2; and for rate independent models, the yield constraints

$$f_i = 0, \tag{4.2.2–4}$$

for active systems (systems in which $f_i < 0$ have $\Delta\lambda_i = 0$).

We assume that the flow surface is sufficiently smooth so that its (second) derivatives with respect to stress and the hardening parameters are well-defined. This is generally true for the models in Abaqus: the exceptions occur at corners or vertices of the surfaces. These special cases are handled individually when they arise.

For some plasticity models the algebraic problem can be solved in closed form. For other models it is possible to reduce the problem to a one variable or a two variable problem that can then be solved to give the entire solution. For example, the Mises yield surface—which is generally used for isotropic metals, together with linear, isotropic elasticity—is a case for which the integrated problem can be solved exactly or in one variable (see "Isotropic elasto-plasticity," Section 4.3.2).

For other rate-independent models with a single yield system the algebraic problem is considered to be a problem in the components of $\Delta\boldsymbol{\varepsilon}^{pl}$. Once these have been found—the elasticity—together with the integrated strain rate decomposition—define the stress. The flow rule then defines $\Delta\lambda$ and the hardening laws define the increments in the hardening variables.

We now derive the equations for the Newton solution of the integrated problem for the case of rate-independent plasticity with a single yield system. The rate-dependent problem with a single yield system is solved in a similar way. For the particular cases of multiple, independent, yield systems (concrete and jointed material) particular techniques are used for this algebraic solution, taking advantage of the simplifications available in those particular models. The concrete model and its integration are described in "An inelastic constitutive model for concrete," Section 4.5.1, and the jointed material model is described in "Constitutive model for jointed materials," Section 4.5.4.

During the solution, the elasticity relationship and the integrated strain rate decomposition are satisfied exactly, so that

$$\mathbf{c}_\sigma = -\mathbf{D}^{el} : \mathbf{c}_\varepsilon, \tag{4.2.2–5}$$

where $\mathbf{c}_\sigma$ is the correction to the stress, $\mathbf{c}_\varepsilon$ is the correction to the plastic strain increments, and

$$\mathbf{D}^{el} = \frac{\partial^2 U}{\partial \boldsymbol{\varepsilon}^{el} \partial \boldsymbol{\varepsilon}^{el}}$$

is the tangent elasticity matrix.

The hardening laws are also satisfied exactly (because the increments of the hardening parameters are defined from these laws) so that

$$c_\alpha = h_\alpha c_\lambda + \Delta\lambda \left(\frac{\partial h_\alpha}{\partial \boldsymbol{\sigma}} : \mathbf{c}_\sigma + \frac{\partial h_\alpha}{\partial H_\beta} c_\beta \right),$$

where c_α is the correction to ΔH_α and c_λ is the correction to $\Delta\lambda$.

This set of equations can be rewritten

$$c_\alpha = \hat{H}_\alpha c_\lambda + \hat{\mathbf{w}}_\alpha : \mathbf{c}_\sigma, \tag{4.2.2–6}$$

where

$$\hat{H}_\alpha = \left[\delta_{\alpha\beta} - \Delta\lambda \frac{\partial h_\beta}{\partial H_\alpha}\right]^{-1} h_\beta$$

and

$$\hat{\mathbf{w}}_\alpha = \Delta\lambda \left[\delta_{\alpha\beta} - \Delta\lambda \frac{\partial h_\beta}{\partial H_\alpha}\right]^{-1} \frac{\partial h_\beta}{\partial \boldsymbol{\sigma}}.$$

The flow rule is not satisfied exactly until the solution has been found, so it gives the Newton equations

$$\mathbf{c}_\varepsilon - c_\lambda \frac{\partial g}{\partial \boldsymbol{\sigma}} - \Delta\lambda \left(\frac{\partial^2 g}{\partial \boldsymbol{\sigma} \partial \boldsymbol{\sigma}} : \mathbf{c}_\sigma + \frac{\partial^2 g}{\partial \boldsymbol{\sigma} \partial H_\alpha} c_\alpha \right) = \Delta\lambda \frac{\partial g}{\partial \boldsymbol{\sigma}} - \Delta\boldsymbol{\varepsilon}^{pl}.$$

Using Equation 4.2.2–5 and Equation 4.2.2–6 allows these equations to be rewritten as

$$\left[\mathbf{I} + \Delta\lambda\, \hat{\mathbf{N}} : \mathbf{D}^{el}\right] : \mathbf{c}_\varepsilon - \hat{\mathbf{n}} c_\lambda = \Delta\lambda \frac{\partial g}{\partial \boldsymbol{\sigma}} - \Delta\boldsymbol{\varepsilon}^{pl}, \tag{4.2.2–7}$$

where

$$\hat{\mathbf{N}} = \frac{\partial^2 g}{\partial \boldsymbol{\sigma} \partial \boldsymbol{\sigma}} + \frac{\partial^2 g}{\partial \boldsymbol{\sigma} \partial H_\alpha} \hat{\mathbf{w}}_\alpha,$$

and

$$\hat{\mathbf{n}} = \frac{\partial g}{\partial \boldsymbol{\sigma}} + \Delta\lambda \frac{\partial^2 g}{\partial \boldsymbol{\sigma} \partial H_\alpha} \hat{H}_\alpha.$$

Likewise, the yield condition is not satisfied exactly during the Newton iteration, so

$$\frac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbf{c}_\sigma + \frac{\partial f}{\partial H_\alpha} c_\alpha = -f.$$

Using Equation 4.2.2–5 and Equation 4.2.2–6 in this equation gives

$$\hat{\mathbf{m}} : \mathbf{D}^{el} : \mathbf{c}_\varepsilon - \frac{\partial f}{\partial H_\alpha} \hat{H}_\alpha c_\lambda = f, \tag{4.2.2–8}$$

where

$$\hat{\mathbf{m}} = \frac{\partial f}{\partial \boldsymbol{\sigma}} + \frac{\partial f}{\partial H_\alpha} \hat{\mathbf{w}}_\alpha.$$

We now eliminate c_λ between Equation 4.2.2–7 and Equation 4.2.2–8. Taking Equation 4.2.2–7 along $\hat{\mathbf{m}} : \mathbf{D}^{el}$ and using Equation 4.2.2–8 gives

$$c_\lambda = \frac{1}{d}\Delta\lambda\,\hat{\mathbf{m}} : \mathbf{D}^{el} : \hat{\mathbf{N}} : \mathbf{D}^{el} : \mathbf{c}_\varepsilon - \frac{1}{d}\hat{\mathbf{m}} : \mathbf{D}^{el} : \left(\Delta\lambda\frac{\partial g}{\partial \boldsymbol{\sigma}} - \Delta\boldsymbol{\varepsilon}^{pl}\right) + \frac{1}{d}f,$$

where

$$d = \hat{\mathbf{m}} : \mathbf{D}^{el} : \hat{\mathbf{n}} - \frac{\partial f}{\partial H_\alpha}\hat{H}_\alpha.$$

Using this equation in Equation 4.2.2–7 then gives

$$\left[\mathbf{I} + \Delta\lambda\,\mathbf{Z} : \hat{\mathbf{N}} : \mathbf{D}^{el}\right] : \mathbf{c}_\varepsilon = \mathbf{Z} : \left(\Delta\lambda\frac{\partial g}{\partial \boldsymbol{\sigma}} - \Delta\boldsymbol{\varepsilon}^{pl}\right) + \frac{1}{d}f\hat{\mathbf{n}},$$

where

$$\mathbf{Z} = \mathbf{I} - \frac{1}{d}\hat{\mathbf{n}}\,\hat{\mathbf{m}} : \mathbf{D}^{el},$$

which is a set of linear equations solved for the $\mathbf{c}_\varepsilon$. The solution is then updated and the Newton loop continued until the flow equation and yield constraint are satisfied.

The solution for rate-dependent plasticity models with a single yield function is developed in the same way, the only differences being the lack of a yield constraint and the identification of $\Delta\lambda$ with time.

Tangent matrix

The tangent matrix for the material, $\partial\boldsymbol{\sigma}/\partial\boldsymbol{\varepsilon}$, is required when Abaqus/Standard is being used for implicit time integration and Newton's method is being used to solve the equilibrium equations. The matrix is obtained directly by taking variations of the integrated equations with respect to all solution parameters, and then solving for the relationship between $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$. The procedure closely follows the derivation used above for the Newton solution: the result is the tangent matrix

$$\boldsymbol{\sigma} = \mathbf{D} : \boldsymbol{\varepsilon},$$

where

$$\mathbf{D} = \left[\mathbf{I} + \Delta\lambda\,\mathbf{D}^{el} : \mathbf{Z} : \hat{\mathbf{N}}\right]^{-1} : \mathbf{D}^{el} : \mathbf{Z}.$$

Reference

- "Inelastic behavior," Section 23.1.1 of the Abaqus Analysis User's Guide

4.3 Metal plasticity

- "Metal plasticity models," Section 4.3.1
- "Isotropic elasto-plasticity," Section 4.3.2
- "Stress potentials for anisotropic metal plasticity," Section 4.3.3
- "Rate-dependent metal plasticity (creep)," Section 4.3.4
- "Models for metals subjected to cyclic loading," Section 4.3.5
- "Porous metal plasticity," Section 4.3.6
- "Cast iron plasticity," Section 4.3.7
- "ORNL constitutive theory," Section 4.3.8
- "Deformation plasticity," Section 4.3.9
- "Heat generation caused by plastic straining," Section 4.3.10

4.3.1 METAL PLASTICITY MODELS

Products: Abaqus/Standard Abaqus/Explicit

Abaqus offers several models for metal plasticity analysis. The main options are a choice between rate-independent and rate-dependent plasticity, a choice between the Mises yield surface for isotropic materials and Hill's yield surface for anisotropic materials, and for rate-independent modeling a choice between isotropic and kinematic hardening. Special plasticity theories are the cast iron model ("Cast iron plasticity," Section 4.3.7), the ORNL model for types 304 and 316 stainless steel in nuclear applications ("ORNL constitutive theory," Section 4.3.8), and deformation plasticity for fracture mechanics applications ("Deformation plasticity," Section 4.3.9).

Rate-independent plasticity is used mostly in modeling the response of metals and some other materials at low temperature (typically below half the melting temperature on an absolute scale) and low strain rates. The rate-independent metal plasticity model uses associated flow.

Two types of rate-dependent models are offered. In the first type a rate-dependent yield strength is introduced in the material model. This is intended for relatively high strain rate applications, such as dynamic events or metal forming process simulations. This type of rate dependence can be introduced in different ways. One way is to use an overstress power law,

$$\dot{\bar{\varepsilon}}^{pl} = D\left(\frac{\bar{\sigma}}{\sigma^0} - 1\right)^n \quad \text{for} \quad \bar{\sigma} \geq \sigma^0,$$

where $\dot{\bar{\varepsilon}}^{pl}$ is the equivalent plastic strain rate; $\bar{\sigma}$ is the yield stress at nonzero plastic strain rate; $\sigma^0(\bar{\varepsilon}^{pl}, \theta, f_i)$ is the static yield stress (which may depend on the plastic strain—$\bar{\varepsilon}^{pl}$—via isotropic hardening, on the temperature—θ—and on other field variables, f_i); and $D(\theta, f_i)$, $n(\theta, f_i)$ are material parameters that can be functions of temperature and, possibly, of other predefined state variables. Another way is to define a yield stress ratio, $\bar{\sigma}/\sigma^0$, as a function of the equivalent plastic strain rate, $\dot{\bar{\varepsilon}}^{pl}$. Both of these options assume that the shapes of the hardening curves at different strain rates are identical. If the shapes of the hardening curves at different strain rates are different, the static and rate-dependent stress-strain relations can be specified directly. The yield stress at a given strain rate is interpolated directly from these relations. Finally, the user can describe general rate-dependent isotropic hardening with user subroutine **UHARD**. See Symonds (1967), Lindholm and Besseny (1969), and Eleiche (1972) for collections of material response measurements or bibliographies of such measurements at high strain dependents.

For high temperature "creep" problems Abaqus/Standard offers some simple built-in creep laws. But for many practical problems the user must write the uniaxial creep behavior into user subroutine **CREEP** because of the complexity of the experimentally measured material response. Creep response under cyclic loading shows significant Bauschinger effects, which cannot be modeled except by introducing sophisticated hardening models. The only capability in Abaqus for such cases is the "ORNL" option. This option uses simple rules to model the Bauschinger effect and is intended primarily as a design evaluation model for the high temperature response of stainless steel. It does not model the material's response in detail. User subroutine **UMAT** must be used if that hardening model is not adequate.

Isotropic hardening means that the yield function is written

$$f(\boldsymbol{\sigma}) = \sigma^0(\varepsilon^{pl}, \theta),$$

where σ^0 is the equivalent (uniaxial) stress, ε^{pl} is the work equivalent plastic strain, defined by

$$\sigma^0 \dot{\varepsilon}^{pl} = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{pl},$$

and θ is temperature.

Isotropic hardening is generally considered to be a suitable model for problems in which the plastic straining goes well beyond the incipient yield state where the Bauschinger effect is noticeable (Rice, 1975). Therefore, this hardening theory is used for such applications as dynamic problems involving finite strains and manufacturing processes—any process involving large plastic strain and in which the plastic strain does not continuously reverse direction sharply.

Some cases, such as low-cycle fatigue situations, involve relatively low amplitude strain cycling. In these cases it becomes important to model the Bauschinger effect. Kinematic hardening is the simplest theory that does this. Abaqus offers a linear kinematic and a nonlinear isotropic/kinematic hardening model for such cases. These models are described in "Models for metals subjected to cyclic loading," Section 4.3.5.

Reference

- "Metal plasticity," Section 23.2 of the Abaqus Analysis User's Guide

4.3.2 ISOTROPIC ELASTO-PLASTICITY

Products: Abaqus/Standard Abaqus/Explicit

This material model is very commonly used for metal plasticity calculations, either as a rate-dependent or as a rate-independent model, and has a particularly simple form. Because of this simplicity the algebraic equations associated with integrating the model are easily developed in terms of a single variable, and the material stiffness matrix can be written explicitly. This results in particularly efficient code. In this section these equations are developed.

For simplicity of notation all quantities not explicitly associated with a time point are assumed to be evaluated at the end of the increment.

The Mises yield function with associated flow means that there is no volumetric plastic strain; since the elastic bulk modulus is quite large, the volume change will be small. Thus, we can define the volume strain as

$$\varepsilon_{vol} = \text{trace}(\boldsymbol{\varepsilon});$$

and, hence, the deviatoric strain is

$$\mathbf{e} = \boldsymbol{\varepsilon} - \frac{1}{3}\varepsilon_{vol}\,\mathbf{I}.$$

Material model definition

The strain rate decomposition is

$$d\boldsymbol{\varepsilon} = d\boldsymbol{\varepsilon}^{el} + d\boldsymbol{\varepsilon}^{pl}.$$

Using the standard definition of corotational measures, this can be written in integrated form as

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{el} + \boldsymbol{\varepsilon}^{pl}. \tag{4.3.2–1}$$

The elasticity is linear and isotropic and, therefore, can be written in terms of two temperature-dependent material parameters. For the purpose of this development it is most appropriate to choose these parameters as the bulk modulus, K, and the shear modulus, G. These are computed readily from the user's input of Young's modulus, E, and Poisson's ratio, ν, as

$$K = \frac{E}{3(1-2\nu)}$$

and

$$G = \frac{E}{2(1+\nu)}\ .$$

The elasticity can be written in volumetric and deviatoric components as follows.
Volumetric:

$$p = -K\varepsilon_{vol}, \tag{4.3.2–2}$$

where

$$p = -\frac{1}{3}\,\text{trace}(\boldsymbol{\sigma})$$

is the equivalent pressure stress.
Deviatoric:

$$\mathbf{S} = 2G\,\mathbf{e}^{el}, \tag{4.3.2–3}$$

where $\mathbf{S}$ is the deviatoric stress,

$$\mathbf{S} = \boldsymbol{\sigma} + p\,\mathbf{I}.$$

The flow rule is

$$d\mathbf{e}^{pl} = d\bar{e}^{pl}\,\mathbf{n}, \tag{4.3.2–4}$$

where

$$\mathbf{n} = \frac{3}{2}\frac{\mathbf{S}}{q},$$

$$q = \sqrt{\frac{3}{2}\mathbf{S} : \mathbf{S}},$$

and $d\bar{e}^{pl}$ is the (scalar) equivalent plastic strain rate.
The plasticity requires that the material satisfy a uniaxial-stress plastic-strain strain-rate relationship. If the material is rate independent, this is the yield condition:

$$q = \sigma^0, \tag{4.3.2–5}$$

where $\sigma^0(\bar{e}^{pl}, \theta)$ is the yield stress and is defined by the user as a function of equivalent plastic strain ($\bar{e}^{pl}$) and temperature (θ).
If the material is rate dependent, the relationship is the uniaxial flow rate definition:

$$\dot{\bar{e}}^{pl} = h(q, \bar{e}^{pl}, \theta),$$

where $\boldsymbol{h}$ is a known function. For example, the rate-dependent material model offers an overstress power law model of the form

$$\dot{\bar{e}}^{pl} = D\left(\frac{q}{\sigma^0} - 1\right)^n,$$

where $D(\theta)$ and $n(\theta)$ are user-defined temperature-dependent material parameters and $\sigma^0(\bar{e}^{pl}, \theta)$ is the static yield stress.

Integrating this relationship by the backward Euler method gives

$$\Delta\bar{e}^{pl} = \Delta t\, h(q, \bar{e}^{pl}, \theta). \tag{4.3.2–6}$$

This equation can be inverted (numerically, if necessary) to give $\boldsymbol{q}$ as a function of $\bar{e}^{pl}$ at the end of the increment.

Thus, both the rate-independent model and the integrated rate-dependent model give the general uniaxial form

$$q = \overline{\sigma}(\bar{e}^{pl}), \tag{4.3.2–7}$$

where $\overline{\sigma} = \sigma^0$ for the rate-independent model and $\overline{\sigma}$ is obtained by inversion of Equation 4.3.2–6 for the rate-dependent model.

Equation 4.3.2–1 to Equation 4.3.2–7 define the material behavior. In any increment when plastic flow is occurring (which is determined by evaluating $\boldsymbol{q}$ based on purely elastic response and finding that its value exceeds σ^0), these equations must be integrated and solved for the state at the end of the increment. As in the general discussion in "Metal plasticity models," Section 4.3.1, the integration is done by applying the backward Euler method to the flow rule (Equation 4.3.2–4), giving

$$\Delta\mathbf{e}^{pl} = \Delta\bar{e}^{pl}\,\mathbf{n}. \tag{4.3.2–8}$$

Combining this with the deviatoric elasticity (Equation 4.3.2–3) and the integrated strain rate decomposition (Equation 4.3.2–1) gives

$$\mathbf{S} = 2G(\mathbf{e}^{el}\big|_t + \Delta\mathbf{e} - \Delta\bar{e}^{pl}\,\mathbf{n}). \tag{4.3.2–9}$$

Using the integrated flow rule (Equation 4.3.2–8), together with the Mises definition of the flow direction, $\mathbf{n}$ (in Equation 4.3.2–4), this becomes

$$(1 + \frac{3G}{q}\Delta\bar{e}^{pl})\mathbf{S} = 2G(\mathbf{e}^{el}\big|_t + \Delta\mathbf{e}).$$

For simplicity of notation we write

$$\hat{\mathbf{e}} = \mathbf{e}^{el}\big|_t + \Delta\mathbf{e},$$

so that this equation is

$$(1 + \frac{3G}{q}\Delta\bar{e}^{pl})\mathbf{S} = 2G\hat{\mathbf{e}}. \tag{4.3.2–10}$$

Taking the inner product of this equation with itself gives

$$q + 3G\Delta\bar{e}^{pl} = 3G\tilde{e}, \tag{4.3.2–11}$$

where

$$\tilde{e} = \sqrt{\frac{2}{3}\hat{\mathbf{e}} : \hat{\mathbf{e}}}\,.$$

The Mises equivalent stress, q, must satisfy the uniaxial form defined in Equation 4.3.2–7, so that from Equation 4.3.2–11,

$$3G(\tilde{e} - \Delta\bar{e}^{pl}) - \overline{\sigma} = 0. \tag{4.3.2–12}$$

This is a nonlinear equation for $\Delta\bar{e}^{pl}$ in the general case when $\overline{\sigma}$ depends on the equivalent plastic strain (that is, when the material is rate-dependent, or when there is nonzero work hardening). (It is linear in Δe^{pl} for rate-independent perfect plasticity.) We solve it by Newton's method:

$$c^{pl} = \frac{3G(\tilde{e} - \Delta\bar{e}^{pl}) - \overline{\sigma}}{3G + H}, \quad \text{where} \quad H = \frac{d\overline{\sigma}}{d\bar{e}^{pl}},$$

$$\Delta\bar{e}^{pl} = \Delta\bar{e}^{pl} + c^{pl},$$

and we iterate until convergence is achieved.

Once $\Delta\bar{e}^{pl}$ is known, the solution is fully defined: using Equation 4.3.2–5,

$$q = \overline{\sigma};$$

and so, from Equation 4.3.2–10,

$$\mathbf{S} = \frac{2G}{1 + \frac{3G}{q}\Delta\bar{e}^{pl}}\hat{\mathbf{e}}.$$

From Equation 4.3.2–4,

$$\mathbf{n} = \frac{3}{2}\frac{\mathbf{S}}{q};$$

and thus, from Equation 4.3.2–6,

$$\Delta \mathbf{e}^{pl} = \Delta \bar{e}^{pl}\, \mathbf{n}.$$

For cases where three direct strain components are provided by the kinematic solution (that is, all but plane stress and uniaxial stress cases), Equation 4.3.2–2 defines

$$p = -K\varepsilon_{vol},$$

so that the solution is then fully defined.

Plane stress

For plane stress ε_{33} is not defined by the kinematics but by the plane stress constraint

$$\sigma_{33} = 0.$$

This additional equation (or equivalently $p = S_{33}$) must be solved along with the yield condition and Equation 4.3.2–9. The predicted third strain component

$$\hat{\varepsilon}_{33} = -\frac{\nu}{1-\nu}(\hat{\varepsilon}_{11} + \hat{\varepsilon}_{22}),$$

where

$$\hat{\varepsilon}_{11} = \varepsilon_{11}^{el}\big|_t + \Delta\varepsilon_{11}$$

and

$$\hat{\varepsilon}_{22} = \varepsilon_{22}^{el}\big|_t + \Delta\varepsilon_{22},$$

serves as an initial guess toward the final value of $\hat{\varepsilon}_{33}$ that enables (with the correct plastic straining) the plane stress condition to be satisfied.

Uniaxial stress

For cases with only one direct strain component defined by the kinematic solution (uniaxial deformation), we require

$$p = S_{22} = S_{33}\ (= -\frac{1}{2}S_{11}),$$

so that

$$\sigma_{11} = \frac{3}{2}S_{11}. \qquad (4.3.2\text{–}13)$$

Material stiffness

For this simple plasticity model the material stiffness matrix can be derived without the need for matrix inversion (as was needed in the general case described in "Integration of plasticity models," Section 4.2.2), as follows.

Taking the variation of Equation 4.3.2–10 with respect to all quantities at the end of the increment gives

$$(1 + \frac{3G}{q}\Delta\bar{e}^{pl})\partial\mathbf{S} + \mathbf{S}\frac{3G}{q}(\partial\bar{e}^{pl} - \frac{\Delta\bar{e}^{pl}}{q}\,\partial q) = 2G\,\partial\hat{\mathbf{e}}. \tag{4.3.2–14}$$

Now, from Equation 4.3.2–5,

$$\partial q = H\partial\bar{e}^{pl},$$

and, from Equation 4.3.2–11,

$$\partial q + 3G\,\partial\bar{e}^{pl} = 3G\,\partial\tilde{e}.$$

Combining these last two results,

$$\partial\bar{e}^{pl} = \frac{1}{1+B}\,\partial\tilde{e},$$

where

$$B = \frac{H}{3G}.$$

From the definition of $\tilde{e}$ (see Equation 4.3.2–11),

$$\partial\tilde{e} = \frac{2}{3\tilde{e}}\hat{\mathbf{e}} : \partial\hat{\mathbf{e}},$$

and, hence,

$$\partial\bar{e}^{pl} = \frac{2}{3}\frac{1}{\tilde{e}(1+B)}\hat{\mathbf{e}} : \partial\hat{\mathbf{e}}.$$

Combining these results with Equation 4.3.2–14 gives

$$\partial\mathbf{S} = \left[Q\,\Im - R\,\mathbf{S}\,\mathbf{S}\right] : \partial\hat{\mathbf{e}},$$

where $Q = \frac{2}{3}q/\tilde{e}$, $\Im$ is the fourth-order unit tensor, and

$$R = \frac{1}{q\tilde{e}} \frac{(1 - \Delta\bar{e}^{pl} H/q)}{(1 + B)}.$$

For all cases where three direct strains are defined by the kinematic solution, the material stiffness is completed by

$$\partial p = -K\,\mathbf{I} : \partial\boldsymbol{\varepsilon},$$

so since

$$\boldsymbol{\sigma} = \mathbf{S} - p\,\mathbf{I}$$

and

$$\partial\hat{\mathbf{e}} = \left(\Im - \frac{\mathbf{1}}{\mathbf{3}}\mathbf{I}\,\mathbf{I}\right) : \partial\boldsymbol{\varepsilon},$$

we have

$$\partial\boldsymbol{\sigma} = \left[Q\,\Im + \left(\boldsymbol{K} - \frac{\mathbf{1}}{\mathbf{3}}\boldsymbol{Q}\right)\mathbf{I}\,\mathbf{I} - \boldsymbol{R}\,\mathbf{S}\,\mathbf{S}\right] : \partial\boldsymbol{\varepsilon}.$$

Plane stress

For the plane stress case the material stiffness matrix is found by imposing

$$\partial\sigma_{33} = 0$$

on the general material stiffness matrix obtained for the plane strain case.

Uniaxial stress

For the uniaxial stress case the material stiffness matrix is available directly from the variation of Equation 4.3.2–13 as

$$\partial\sigma_{11} = [\frac{3}{2}Q - R\,\sigma_{11}^2]\,\partial\varepsilon_{11}.$$

Reference

- "Classical metal plasticity," Section 23.2.1 of the Abaqus Analysis User's Guide

4.3.3 STRESS POTENTIALS FOR ANISOTROPIC METAL PLASTICITY

Products: Abaqus/Standard Abaqus/Explicit

The metal plasticity models in Abaqus use the Mises stress potential for isotropic behavior and the Hill stress potentials for anisotropic behavior. Both of these potentials depend only on the deviatoric stress, so the plastic part of the response is incompressible. This means that, in cases where the plastic flow dominates the response (such as limit load calculations or metal forming problems), except for plane stress problems, the finite elements should be chosen so that they can accommodate the incompressible flow. Usually the reduced integration elements are used for this purpose: in Abaqus/Standard the "hybrid" elements can also be used, at higher cost. The fully integrated first-order continuum elements in Abaqus/Standard use selectively reduced integration, whereby the volumetric strain is calculated at the centroid of the element only. Those elements that are described in "Solid isoparametric quadrilaterals and hexahedra," Section 3.2.4, are also suitable for such problems.

The Mises stress potential is

$$f(\boldsymbol{\sigma}) = q,$$

where

$$q = \sqrt{\frac{3}{2}\mathbf{S} : \mathbf{S}},$$

in which $\mathbf{S}$ is the deviatoric stress:

$$\mathbf{S} = \boldsymbol{\sigma} - \frac{1}{3}\mathrm{trace}(\boldsymbol{\sigma})\mathbf{I} = \boldsymbol{\sigma} - \frac{1}{3}\mathbf{I}\,\mathbf{I} : (\boldsymbol{\sigma}).$$

The potential is a circle in the plane normal to the hydrostatic axis in principal stress space. For this function,

$$\frac{\partial f}{\partial \boldsymbol{\sigma}} = \frac{1}{q}\frac{3}{2}\mathbf{S},$$

and

$$\frac{\partial^2 f}{\partial \boldsymbol{\sigma}\partial \boldsymbol{\sigma}} = \frac{1}{q}\left(\frac{3}{2}\mathfrak{S} - \frac{\mathbf{1}}{\mathbf{2}}\mathbf{I}\,\mathbf{I} - \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\sigma}}\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\sigma}}\right)$$

in which $\mathfrak{S}$ is the fourth-order unit tensor.

Hill's stress function is a simple extension of the Mises function to allow anisotropic behavior. The function is

$$f(\boldsymbol{\sigma}) = \sqrt{F(\sigma_y - \sigma_z)^2 + G(\sigma_z - \sigma_x)^2 + H(\sigma_x - \sigma_y)^2 + 2L{\tau_{yz}}^2 + 2M{\tau_{zx}}^2 + 2N{\tau_{xy}}^2},$$

in terms of rectangular Cartesian stress components, where F, G, H, L, M, N are constants obtained by tests of the material in different orientations. They are defined as

$$F = \frac{\sigma_0^2}{2}\left(\frac{1}{\bar{\sigma}_{22}^2} + \frac{1}{\bar{\sigma}_{33}^2} - \frac{1}{\bar{\sigma}_{11}^2}\right),$$

$$G = \frac{\sigma_0^2}{2}\left(\frac{1}{\bar{\sigma}_{33}^2} + \frac{1}{\bar{\sigma}_{11}^2} - \frac{1}{\bar{\sigma}_{22}^2}\right),$$

$$H = \frac{\sigma_0^2}{2}\left(\frac{1}{\bar{\sigma}_{11}^2} + \frac{1}{\bar{\sigma}_{22}^2} - \frac{1}{\bar{\sigma}_{33}^2}\right),$$

$$L = \frac{3}{2}\left(\frac{\tau_0}{\bar{\tau}_{23}}\right)^2,$$

$$M = \frac{3}{2}\left(\frac{\tau_0}{\bar{\tau}_{13}}\right)^2,$$

$$N = \frac{3}{2}\left(\frac{\tau_0}{\bar{\tau}_{12}}\right)^2,$$

where $\sigma_0, \bar{\sigma}_{11}, \bar{\sigma}_{22}, \bar{\sigma}_{33}, \bar{\tau}_{12}, \bar{\tau}_{23}, \bar{\tau}_{13}$ are specified by the user and $\tau_0 = \sigma_0/\sqrt{3}$. $\bar{\sigma}$ and $\bar{\tau}$ are the values of stress that make the potential equal to σ_0 if only one stress component is nonzero.

For this function

$$\frac{\partial f}{\partial \boldsymbol{\sigma}} = \frac{1}{f}\mathbf{b},$$

where

$$\mathbf{b} = \begin{bmatrix} -G(\sigma_z - \sigma_x) + H(\sigma_x - \sigma_y) \\ F(\sigma_y - \sigma_z) - H(\sigma_x - \sigma_y) \\ -F(\sigma_y - \sigma_z) + G(\sigma_z - \sigma_x) \\ 2N\tau_{xy} \\ 2M\tau_{zx} \\ 2L\tau_{yz} \end{bmatrix}.$$

In addition,

$$\frac{\partial^2 f}{\partial \boldsymbol{\sigma} \partial \boldsymbol{\sigma}} = \frac{1}{f}\left(\frac{\partial \mathbf{b}}{\partial \boldsymbol{\sigma}} - \frac{1}{f^2}\mathbf{b}\,\mathbf{b}\right),$$

where

$$\frac{\partial \mathbf{b}}{\partial \boldsymbol{\sigma}} = \begin{bmatrix} G+H & -H & -G & 0 & 0 & 0 \\ -H & F+H & -F & 0 & 0 & 0 \\ -G & -F & F+G & 0 & 0 & 0 \\ 0 & 0 & 0 & 2N & 0 & 0 \\ 0 & 0 & 0 & 0 & 2M & 0 \\ 0 & 0 & 0 & 0 & 0 & 2L \end{bmatrix}.$$

Reference

- "Anisotropic yield/creep," Section 23.2.6 of the Abaqus Analysis User's Guide

4.3.4 RATE-DEPENDENT METAL PLASTICITY (CREEP)

Product: Abaqus/Standard

The rate-dependent plasticity (creep) models provided in Abaqus/Standard are used to model inelastic straining of materials that are rate sensitive. High-temperature "creep" in structures is one important class of examples of the application of such a material model. Because such problems generally involve relatively small amounts of inelastic straining (otherwise the structure is not a suitable design), the explicit, forward Euler method is often satisfactory as an integrator for the flow rule. This method is only conditionally stable, but the stability limit is usually sufficiently large compared to the time history of interest in such cases that the explicit method is very economical. Cormeau (1975) has developed formulae for the stability limit for most common cases of stress induced creep, and these results are used to monitor stability. For this explicit approach the integration is trivial. Combining the integrated flow rule

$$\Delta\boldsymbol{\varepsilon}^{pl} = \Delta t \frac{\partial G_{cr}}{\partial \boldsymbol{\sigma}}|_t$$

with the integrated strain rate decomposition and the (linear) elasticity gives

$$\boldsymbol{\sigma}|_{t+\Delta t} = \mathbf{D}^{el} : \left(\boldsymbol{\varepsilon}_{t+\Delta t} - \boldsymbol{\varepsilon}^{pl}|_t - \Delta t \left. \frac{\partial G_{cr}}{\partial \boldsymbol{\sigma}} \right|_t \right). \tag{4.3.4–1}$$

All of the terms on the right-hand side of this set of equations are known when the constitutive integration is done, so these equations define $\boldsymbol{\sigma}_{t+\Delta t}$ explicitly.

There also exist many problems involving rate-dependent plastic response in which the characteristic relaxation times for the material under the stress states to which it is subjected are very short compared to the time period of interest in the analysis, so the conditional stability of the explicit operator will only allow very short time increments. For such cases it can be more economical to use the backward Euler method because of its unconditional stability. Abaqus always uses the implicit method for high strain rate applications to avoid time increment restrictions being introduced by considerations of stability in the integration of the constitutive model. Abaqus will also use the implicit method in all geometrically nonlinear problems and in problems for which rate-independent plasticity is active simultaneously.

The backward Euler method is implicit; and because the plastic strain rate is usually a strong function of stress, some care must be taken to develop an effective algorithm to solve the nonlinear algebraic equations that result from the use of this operator. The problem has been posed formally in "Integration of plasticity models," Section 4.2.2. The main difficulty is to obtain a reasonable starting guess for $\Delta\boldsymbol{\varepsilon}^{pl}$. For this we proceed as follows.

For simplicity, we consider rate-dependent behavior only and the particular form of flow rule defined by

$$\dot{\boldsymbol{\varepsilon}}^{pl} = \dot{\bar{\varepsilon}}^{sw} \frac{1}{3}\mathbf{I} + \dot{\bar{\varepsilon}}^{cr}\mathbf{n},$$

where $\dot{\bar{\varepsilon}}^{sw}$ is the "equivalent swelling strain rate," $\dot{\bar{\varepsilon}}^{cr}$ is the "equivalent creep strain rate," and $\mathbf{n}$ is the gradient of the deviatoric stress potential,

$$\mathbf{n} = \frac{\partial \tilde{q}}{\partial \boldsymbol{\sigma}},$$

where $\tilde{q}$ is the Mises or Hill stress potential (defined in "Stress potentials for anisotropic metal plasticity," Section 4.3.3).

The "equivalent strain rates" are part of the stress potential for the plastic response and, therefore, are assumed to have evolution laws of the form

$$\dot{\overline{\varepsilon}}^{sw} = h^s(p, \tilde{q}, \overline{\varepsilon}^{sw}, \overline{\varepsilon}^{cr}, \theta, \ldots)$$

and

$$\dot{\overline{\varepsilon}}^{cr} = h^c(p, \tilde{q}, \overline{\varepsilon}^{sw}, \overline{\varepsilon}^{cr}, \theta, \ldots).$$

Backward Euler integration of the flow equation gives

$$\Delta \boldsymbol{\varepsilon}^{pl} = \Delta \overline{\varepsilon}^{sw} \frac{1}{3}\mathbf{I} + \Delta \overline{\varepsilon}^{cr} \mathbf{n},$$

where $\mathbf{n}$ is understood to be evaluated at time $t + \Delta t$, and

$$\Delta \overline{\varepsilon}^{sw} = \Delta t \, h^s(p, \tilde{q}, \overline{\varepsilon}^{sw}, \overline{\varepsilon}^{cr}, \theta, \ldots) \tag{4.3.4–2}$$

and

$$\Delta \overline{\varepsilon}^{cr} = \Delta t \, h^c(p, \tilde{q}, \overline{\varepsilon}^{sw}, \overline{\varepsilon}^{cr}, \theta, \ldots). \tag{4.3.4–3}$$

$\Delta \overline{\varepsilon}^{cr}$ and $\Delta \overline{\varepsilon}^{sw}$ are usually defined in user subroutine **CREEP**.

The solution to the algebraic problem is obtained by first finding reasonable initial guesses for $\Delta \overline{\varepsilon}^{cr}$ and $\Delta \overline{\varepsilon}^{sw}$ and then solving the full problem.

The Mises and Hill equivalent stress definitions ($\tilde{q}$) both have the property that

$$\mathbf{n} : \boldsymbol{\sigma} = \tilde{q}.$$

We also have the simple relationship

$$\mathbf{I} : \boldsymbol{\sigma} = -3p.$$

The initial estimates for $\Delta \overline{\varepsilon}^{cr}$ and $\Delta \overline{\varepsilon}^{sw}$ are obtained by projecting the problem onto $\mathbf{n}^{el}$ and $\mathbf{I}$, where $\mathbf{n}^{el}$ is $\partial \tilde{q}/\partial \boldsymbol{\sigma}$ defined at $\boldsymbol{\sigma}^{el}$, the stress state that would arise at the end of the increment if there were no inelastic deformation during the increment. The projections are

$$\tilde{q} = \tilde{q}^{el} - G\Delta \overline{\varepsilon}^{cr} - B\Delta \overline{\varepsilon}^{sw}, \tag{4.3.4–4}$$

and

$$p = p^{el} - B\Delta\overline{\varepsilon}^{cr} - K\Delta\overline{\varepsilon}^{sw}, \tag{4.3.4–5}$$

where

$$\tilde{q}^{el} = \mathbf{n} : \boldsymbol{\sigma}^{el},$$

$$p^{el} = -\frac{1}{3}\mathbf{I} : \boldsymbol{\sigma}^{el},$$

$$G = \mathbf{n} : \mathbf{D}^{el} : \mathbf{n},$$

$$K = \frac{1}{9}\mathbf{I} : \mathbf{D}^{el} : \mathbf{I},$$

and

$$B = \frac{1}{3}\mathbf{n}^{el} : \mathbf{D}^{el} : \mathbf{I}.$$

Equation 4.3.4–2 to Equation 4.3.4–5 are a set of nonlinear equations that can be solved for $\Delta\overline{\varepsilon}^{cr}$ and $\Delta\overline{\varepsilon}^{sw}$. We solve these equations by Newton's method and then use this solution as the starting estimate for solving the complete problem. When the Mises stress potential is used and the problem is not plane stress, this starting estimate is the solution to the complete problem because the Mises stress potential is a circle in the deviatoric plane.

User subroutine CREEP

Abaqus/Standard provides a very general capability for implementing viscoplastic models such as creep and swelling in which the strain rate potential can be written as a function of equivalent pressure stress, p; the Mises or Hill's equivalent deviatoric stress, $\tilde{q}$; and any number of solution-dependent state variables. The purpose of this section is to provide an overview of the operations that need to be performed in user subroutine **CREEP**. To illustrate the main ideas, the creep law is assumed to be of a strain hardening type and of the form

$$\dot{\overline{\varepsilon}}^{cr} = B\tilde{q}^n f(\overline{\varepsilon}^{cr}),$$

where B and n are material constants, and f is a nonlinear function of its argument. In user subroutine **CREEP** the user needs to define the creep strain increment based on the above creep law. Given that the creep law is in rate form, an integration scheme is needed to convert it to an incremental form defining the creep strain increment. This conversion can be accomplished by using either an explicit (forward Euler) or an implicit (backward Euler) integration scheme. In the explicit scheme the creep strain rate

during any time increment is defined in terms of (known) quantities at the beginning of the increment. Thus, an explicit integration would lead to the following incremental form of the creep law:

$$\Delta\overline{\varepsilon}^{cr} = B\tilde{q}_t^n f(\overline{\varepsilon}_t^{cr})\Delta t.$$

In an implicit scheme the creep strain rate during any time increment is defined in terms of (unknown) quantities at the end of the increment. Thus, an implicit scheme would lead to the following incremental form of the creep law:

$$\Delta\overline{\varepsilon}^{cr} = B\tilde{q}_{t+\Delta t}^n f(\overline{\varepsilon}_{t+\Delta t}^{cr})\Delta t = h^c(\tilde{q}_{t+\Delta t}, \overline{\varepsilon}_{t+\Delta t}^{cr}),$$

where h^c is a nonlinear function of its arguments. In the incremental forms above the subscripts t and $t+\Delta t$ refer to the values of the corresponding quantities at the beginning and at the end of the increment, respectively.

Recognizing that $\overline{\varepsilon}_{t+\Delta t}^{cr} = \overline{\varepsilon}_t^{cr} + \Delta\overline{\varepsilon}^{cr}$ and also that $\tilde{q}_{t+\Delta t}$ is a function of $\overline{\varepsilon}_{t+\Delta t}^{cr}$, the implicit integration scheme leads to a nonlinear equation in the creep strain increment $\Delta\overline{\varepsilon}^{cr}$ that is solved by Abaqus iteratively at each material point (these are local material point iterations and are not the same as the global equilibrium iterations). The user subroutine gets called at each integration point during each local iteration. The Newton-Raphson scheme used to solve the above nonlinear equation iteratively is given by

$$c\Delta\overline{\varepsilon}_{i+1}^{cr} = -[\frac{\partial h^c}{\partial\Delta\overline{\varepsilon}^{cr}}]_i^{-1} h^c(\tilde{q}_{t+\Delta t}, \overline{\varepsilon}_{t+\Delta t}^{cr})_i,$$

$$\Delta\overline{\varepsilon}_{i+1}^{cr} = \Delta\overline{\varepsilon}_i^{cr} + c\Delta\overline{\varepsilon}^{cr},$$

where the subscripts i and $i+1$ are iteration counters, and $c\Delta\overline{\varepsilon}^{cr}$ represents the correction to the creep strain increment. As illustrated by the two equations above, the Jacobian of the implicit scheme requires the partial derivative of the function h^c with respect to $\Delta\overline{\varepsilon}^{cr}$. For the example considered here the Jacobian may be further expressed as

$$\frac{\partial h^c}{\partial\Delta\overline{\varepsilon}^{cr}} = \frac{\partial h^c}{\partial\tilde{q}_{t+\Delta t}}\frac{\partial\tilde{q}_{t+\Delta t}}{\partial\Delta\overline{\varepsilon}^{cr}} + \frac{\partial h^c}{\partial\overline{\varepsilon}_{t+\Delta t}^{cr}}\frac{\partial\overline{\varepsilon}_{t+\Delta t}^{cr}}{\partial\Delta\overline{\varepsilon}^{cr}}.$$

There are additional terms if the function h^c also depends on the hydrostatic pressure. In an implicit scheme the user also needs to define the appropriate derivatives that enter the Jacobian of the nonlinear equation for $\Delta\overline{\varepsilon}^{cr}$. In the example above the user needs to define the quantities $\frac{\partial h^c}{\partial\tilde{q}_{t+\Delta t}}$ and $\frac{\partial h^c}{\partial\overline{\varepsilon}_{t+\Delta t}^{cr}}$. The incremental creep strain and the Jacobian contributions require the values of the equivalent Mises or Hill's stress (and the pressure stress, if relevant) and the equivalent creep strain at both the beginning and the end of the increment, which are available in the user subroutine.

On the other hand, in the explicit scheme the creep strain increment is defined in terms of quantities known at the beginning of the increment and, hence, no local iterations are needed. However, as discussed in "Rate-dependent plasticity: creep and swelling," Section 23.2.4 of the Abaqus Analysis User's Guide, the explicit scheme has limitations related to stability.

Irrespective of the integration scheme used to integrate the rate form of the creep equation, user subroutine **CREEP** is called at each material point once at the beginning and once at the end of each increment. These calls are for the purpose of getting the creep strain increment based on the creep strain rate at the beginning and at the end of the increment, respectively. The difference between these two creep strain increment values measures the accuracy of the integration scheme and must be less than the value specified on the relevant analysis step option for the maximum difference.

Reference

- "Rate-dependent plasticity: creep and swelling," Section 23.2.4 of the Abaqus Analysis User's Guide

4.3.5 MODELS FOR METALS SUBJECTED TO CYCLIC LOADING

Products: Abaqus/Standard Abaqus/Explicit

The kinematic hardening models in Abaqus are intended to simulate the behavior of metals that are subjected to cyclic loading. These models are typically applied to studies of low-cycle fatigue and ratchetting. The basic concept of these models is that the yield surface shifts in stress space so that straining in one direction reduces the yield stress in the opposite direction, thus simulating the Bauschinger effect and anisotropy induced by work hardening.

Two kinematic hardening models are available in Abaqus. The simplest model provides linear kinematic hardening and is, thus, mainly used for low-cycle fatigue evaluations. This model yields physically reasonable results if the uniaxial behavior is linearized in the plastic range (a constant work-hardening slope). This is usually best accomplished by guessing the strain levels that will be attained in the problem and linearizing the actual material behavior accordingly. It is important to recognize this restriction on the theory's ability to provide reasonable results and to provide material data accordingly. This model is available with the Mises or Hill yield surface.

The combined isotropic/kinematic hardening model is an extension of the linear model. It provides a more accurate approximation to the stress-strain relation than the linear model. It also models other phenomena—such as ratchetting, relaxation of the mean stress, and cyclic hardening—that are typical of materials subjected to cyclic loading. In addition, it is possible to superpose several kinematic hardening models, which in general will produce more accurate results when the range of changes in strain is significant and will allow more accurate modeling of the ratchetting effect. This model is available only with the Mises yield surface.

This section first describes those aspects of the formulation that are common to both models; the specific formulation of each model is presented subsequently.

Strain rate decomposition

The total strain rate $\dot{\varepsilon}$ is written in terms of the elastic and plastic strain rates as

$$\dot{\varepsilon} = \dot{\varepsilon}^{el} + \dot{\varepsilon}^{pl}. \tag{4.3.5–1}$$

Elastic behavior

The elastic behavior can be modeled only as linear elastic

$$\boldsymbol{\sigma} = \mathbf{D}^{el} : \boldsymbol{\varepsilon}, \tag{4.3.5–2}$$

where $\mathbf{D}^{el}$ represents the fourth-order elasticity tensor and $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ are the second-order stress and strain tensors, respectively.

Plastic behavior

The models are pressure-independent plasticity models. For both models the yield surface is defined by the function

$$f(\boldsymbol{\sigma} - \boldsymbol{\alpha}) = \sigma^0, \tag{4.3.5–3}$$

where $f(\boldsymbol{\sigma} - \boldsymbol{\alpha})$ is the equivalent Mises stress or Hill's potential with respect to the backstress or "kinematic shift" $\boldsymbol{\alpha}$, and σ^0 is the size of the yield surface. For example, the equivalent Mises stress is defined as

$$f(\boldsymbol{\sigma} - \boldsymbol{\alpha}) = \sqrt{\frac{3}{2}(\mathbf{S} - \boldsymbol{\alpha}^{dev}) : (\mathbf{S} - \boldsymbol{\alpha}^{dev})}, \tag{4.3.5–4}$$

where $\boldsymbol{\alpha}^{dev}$ is the deviatoric part of the backstress and $\mathbf{S}$ is the deviatoric stress tensor.

These models assume associated plastic flow:

$$\dot{\boldsymbol{\varepsilon}}^{pl} = \frac{\partial f(\boldsymbol{\sigma} - \boldsymbol{\alpha})}{\partial \boldsymbol{\sigma}} \dot{\bar{\varepsilon}}^{pl}, \tag{4.3.5–5}$$

where $\dot{\boldsymbol{\varepsilon}}^{pl}$ represents the rate of plastic flow and $\dot{\bar{\varepsilon}}^{pl}$ is the equivalent plastic strain rate, $\dot{\bar{\varepsilon}}^{pl} = \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{pl} : \dot{\boldsymbol{\varepsilon}}^{pl}}$.

Linear kinematic hardening model

This model is the simpler of the two kinematic hardening models available in Abaqus. The size of the yield surface, $\sigma^0(\theta)$, can be a function of temperature only for this model. The evolution of $\boldsymbol{\alpha}$ is defined by Ziegler's hardening rule, generalized to the nonisothermal case as

$$\dot{\boldsymbol{\alpha}} = C\dot{\bar{\varepsilon}}^{pl}\frac{1}{\sigma^0}(\boldsymbol{\sigma} - \boldsymbol{\alpha}) + \frac{1}{C}\boldsymbol{\alpha}\dot{C}, \tag{4.3.5–6}$$

where $C(\theta)$ is the hardening parameter ($C(\theta)$ is the work-hardening slope of the isothermal uniaxial stress-strain response, $d\sigma/d\bar{\varepsilon}^{pl}$, taken at different temperatures) and $\dot{C}$ is the rate of change of C with respect to time. This form of evolution law for $\boldsymbol{\alpha}$ defines the rate of $\boldsymbol{\alpha}$ due to plastic straining to be in the direction of the current radius vector from the center of the yield surface, $\boldsymbol{\sigma} - \boldsymbol{\alpha}$, and the rate due to temperature changes to be toward the origin of stress space. The inclusion of the $\dot{C}$ term in the evolution of $\boldsymbol{\alpha}$ ensures that the material response is temperature history independent and, consequently, can be characterized with isothermal data. Rice (1975) writes this concept quite generally as

$$\dot{\boldsymbol{\alpha}} = \dot{\mu}(\boldsymbol{\sigma} - \boldsymbol{\alpha}) + h\,\boldsymbol{\alpha}\,\dot{\theta}. \tag{4.3.5–7}$$

The particular identification of $\dot{\mu} = C\dot{\bar{\varepsilon}}^{pl}/\sigma^0$ and $h = (dC/d\theta)/C$ in Equation 4.3.5–6 above is assumed, so the material behavior is defined by the isothermal, uniaxial work hardening data, $C(\theta)$ only.

Nonlinear isotropic/kinematic hardening model

This model is based on the work of Lemaitre and Chaboche (1990). The size of the yield surface, $\sigma^0(\bar{\varepsilon}^{pl}, \theta, f_i)$, is defined as a function of equivalent plastic strain, $\bar{\varepsilon}^{pl}$; temperature, θ; and field variables, f_i. This dependency can be provided directly, can be coded in user subroutine **UHARD**, or can be modeled with a simple exponential law for materials that either cyclically harden or soften as

$$\sigma^0 = \sigma|_0 + Q_\infty(1 - e^{-b\bar{\varepsilon}^{pl}}), \tag{4.3.5–8}$$

where $\sigma|_0(\theta, f_i)$ is the yield surface size at zero plastic strain, and $Q_\infty(\theta, f_i)$ and $b(\theta, f_i)$ are additional material parameters that must be calibrated from cyclic test data.

The overall backstress is composed of multiple backstress components, where the evolution of the backstress components of the model is defined as

$$\dot{\boldsymbol{\alpha}}_k = C_k\,\dot{\bar{\varepsilon}}^{pl}\frac{1}{\sigma^0}(\boldsymbol{\sigma} - \boldsymbol{\alpha}) - \gamma_k\,\boldsymbol{\alpha}_k\,\dot{\bar{\varepsilon}}^{pl} + \frac{1}{C_k}\boldsymbol{\alpha}_k\,\dot{C}_k \tag{4.3.5–9}$$

and the overall backstress is computed from the relation

$$\boldsymbol{\alpha} = \sum_{k=1}^{N}\boldsymbol{\alpha}_k, \tag{4.3.5–10}$$

where N is the number of backstresses, C_k and γ_k are material parameters, and $\dot{C}_k$ is the rate of change of C_k with respect to temperature and field variables. This equation is the basic Ziegler law, generalized to account for temperature and field-variable dependency of C_k and to which a "recall" term, $\gamma_k\,\boldsymbol{\alpha}_k\,\dot{\bar{\varepsilon}}^{pl}$, has been added. The recall term introduces nonlinearity in the evolution law. The rate of change of γ_k with respect to temperature and field variables is not accounted for in the evolution law for $\boldsymbol{\alpha}_k$. Consequently, if γ_k vary with temperature, the material response predicted by the model will be temperature history dependent. However, it can be shown that this dependency on temperature history is small and fades away with increasing plastic deformation. The condition for complete temperature history independent behavior is to use constant values for γ_k.

The evolution of the backstress and the isotropic hardening are illustrated in Figure 4.3.5–1 for unidirectional loading and in Figure 4.3.5–2 for multiaxial loading.

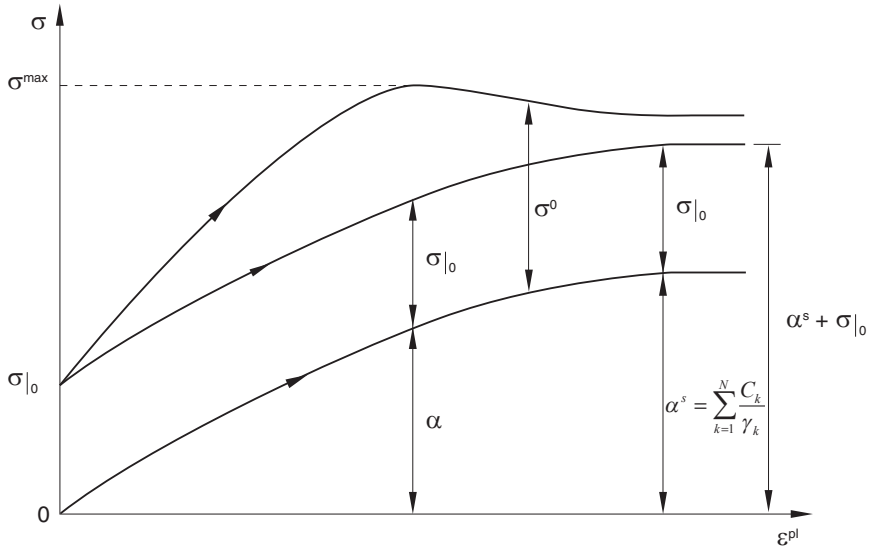


Figure 4.3.5–1 One-dimensional representation of the nonlinear model.

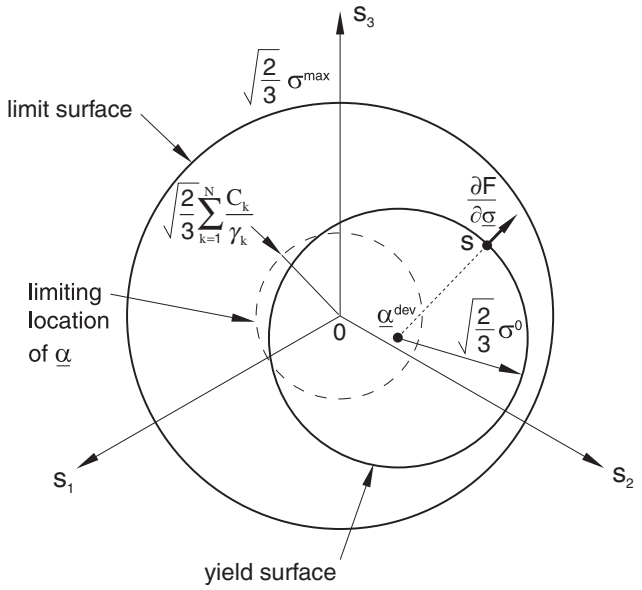


Figure 4.3.5–2 Three-dimensional representation of the nonlinear model.

The center of the yield surface is contained within a cylinder of radius $\sqrt{2/3}\sum_k^N C_k/\gamma_k$, which follows directly from Equation 4.3.5–10. Therefore, the yield surface is contained within the limiting surface of radius $\sqrt{2/3}\,\sigma^{max}$, as shown in the figures. This model can be degenerated into the linear kinematic model described above by setting $\sigma^0 = \sigma|_0$ and $\gamma_k = 0$, for $k = 1, ..., N$.

The physical behavior that can be captured by this model, as well as its limitations, is described in detail in the Abaqus Analysis User's Guide.

Reference

- "Models for metals subjected to cyclic loading," Section 23.2.2 of the Abaqus Analysis User's Guide

4.3.6 POROUS METAL PLASTICITY

Products: Abaqus/Standard Abaqus/Explicit

The porous metal plasticity model is intended for use with *mildly voided metals*. Even though the material that contains the voids (also known as the matrix material) is assumed to be plastically incompressible, the plastic behavior of the bulk material is pressure-dependent due to the presence of voids. The model is described in the following paragraphs, followed by a brief description of the material point calculations.

Yield condition

For a metal containing a dilute concentration of voids, based on a rigid-plastic upper bound solution for spherically symmetric deformations of a single spherical void, Gurson (1977) proposed a yield condition of the form

$$\phi = \left(\frac{q}{\sigma_y}\right)^2 + 2q_1 f \cosh\left(-\frac{3}{2}\frac{q_2 p}{\sigma_y}\right) - (1 + q_3 f^2) = 0,$$

where

$$\mathbf{S} = p\mathbf{I} + \boldsymbol{\sigma}$$

is the deviatoric part of the macroscopic Cauchy stress tensor $\boldsymbol{\sigma}$;

$$q = \sqrt{\frac{3}{2}\,\mathbf{S} : \mathbf{S}}$$

is the Mises stress;

$$p = -\frac{1}{3}\boldsymbol{\sigma} : \mathbf{I}$$

is the hydrostatic pressure; f is the volume fraction of the voids in the material; and $\sigma_y(\bar{\varepsilon}_m^{pl})$ is the yield stress of the fully dense matrix material as a function of $\bar{\varepsilon}_m^{pl}$, the equivalent plastic strain in the matrix. Tvergaard (1981) introduced the constants q_1, q_2, and $q_3 = q_1^2$ (as coefficients of the void volume fraction and pressure terms) to make the predictions of the Gurson model agree with numerical studies of "ordered" voided materials in plane strain tensile fields; one can recover the original Gurson model by setting $q_1 = q_2 = q_3 = 1$.

It should be noted that $f = 0$ implies that the material is fully dense, and the Gurson yield condition reduces to that of von Mises; $f = 1$ implies that the material is fully voided and has no stress carrying capacity. This is illustrated in Figure 4.3.6–1, where the yield surfaces for different levels of porosity are shown in the p–q plane. Figure 4.3.6–2 compares the behavior of a porous metal (which has an initial void volume fraction of f_0) in tension and compression against the perfectly plastic matrix material; the initial yield stress of the porous metal is σ_{y_0}.

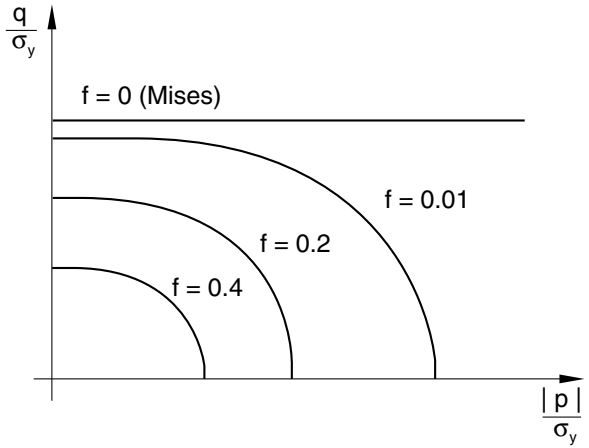


Figure 4.3.6–1 Schematic of the yield surface in the p–q plane.

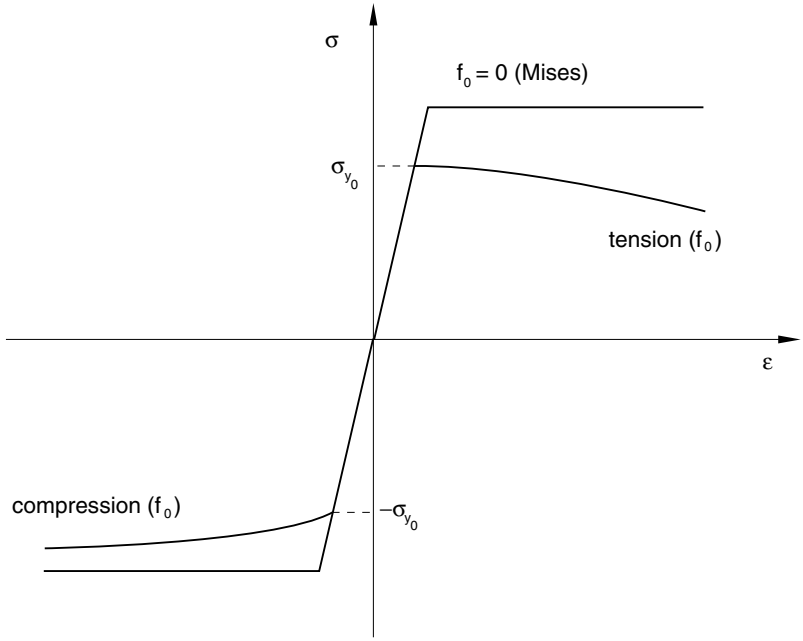


Figure 4.3.6–2 Schematic of uniaxial behavior of a porous metal.

In compression the porous material "hardens" due to closing of the voids, and in tension it "softens" due to growth and nucleation of the voids.

Flow rule

The plastic strains are derived from the yield potential; the presence of the first invariant of the stress tensor in the yield condition results in nondeviatoric plastic strains:

$$\dot{\boldsymbol{\varepsilon}}^{pl} = \dot{\lambda}\frac{\partial \phi}{\partial \boldsymbol{\sigma}} = \dot{\lambda}\left(-\frac{1}{3}\frac{\partial \phi}{\partial p}\mathbf{I} + \frac{3}{2q}\frac{\partial \phi}{\partial q}\mathbf{S}\right),$$

where $\dot{\lambda}$ is the nonnegative plastic flow multiplier.

Evolution of the plastic strains and f

The hardening of the (fully dense) matrix material is described through $\sigma_y = \sigma_y(\bar{\varepsilon}_m^{pl})$. The evolution of $\bar{\varepsilon}_m^{pl}$ is assumed to be governed by the equivalent plastic work expression; i.e.,

$$(1-f)\sigma_y \dot{\bar{\varepsilon}}_m^{pl} = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{pl}.$$

The change in volume fraction of the voids is due partly to the growth of existing voids and partly to the nucleation of voids. Growth of existing voids is based on the law of conservation of mass and is expressed as

$$\dot{f}_{\text{gr}} = (1-f)\,\dot{\boldsymbol{\varepsilon}}^{pl} : \mathbf{I}.$$

Nucleation of voids can occur due to micro-cracking and/or decohesion of the particle-matrix interface. Abaqus assumes that the nucleation of new voids is plastic strain controlled (see Chu and Needleman, 1980), so that

$$\dot{f}_{\text{nucl}} = A\,\dot{\bar{\varepsilon}}_m^{pl},$$

where

$$A = \frac{f_N}{s_N\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{\bar{\varepsilon}_m^{pl} - \varepsilon_N}{s_N}\right)^2\right].$$

The normal distribution of the nucleation strain has a mean value ε_N, a standard deviation s_N, and nucleates voids with volume fraction f_N. The total rate of change of f is given as

$$\dot{f} = \dot{f}_{\text{gr}} + \dot{f}_{\text{nucl}}.$$

Voids are nucleated only in tension; Abaqus will not consider the nucleation term at a material point if the stress state is compressive.

The nucleation function A/f_N, which is assumed to have a normal distribution, is shown in Figure 4.3.6–3 for different values of the parameter s_N. Figure 4.3.6–4 shows the extent of softening in a uniaxial tension test of a porous material for different values of f_N.

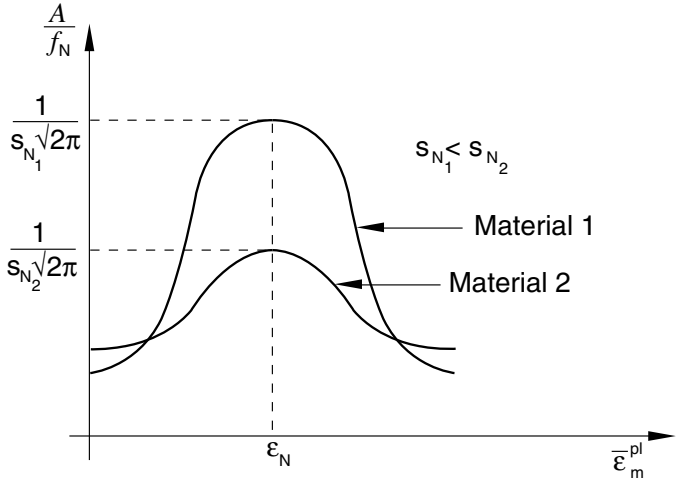


Figure 4.3.6–3 Nucleation function A/f_N.

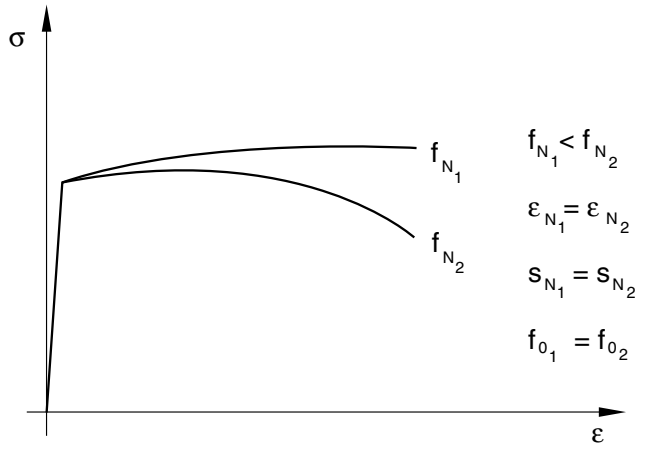


Figure 4.3.6–4 Softening (in uniaxial tension) as a function of f_N.

Integration of the elastoplastic equations

The integration of the elastoplastic equations for the porous plasticity model is carried out using the backward Euler scheme proposed by Aravas (1987). This method is briefly discussed in the following paragraphs; the user can refer to the paper for further details.

During the constitutive calculations in an increment, the stress and state variables are known at time t (beginning of the increment). Given a total incremental strain $\Delta\boldsymbol{\varepsilon}$, the stress and state variables need to be updated at $t + \Delta t$ (end of the increment) so that they satisfy the yield condition, flow rules, and evolution equation of the state variables. To do this, consider the elasticity equations

$$\boldsymbol{\sigma} = \mathbf{D}^{el} : \boldsymbol{\varepsilon}^{el} = \mathbf{D}^{el} : (\boldsymbol{\varepsilon}^{el}|_t + \Delta\boldsymbol{\varepsilon}^{el}) = \boldsymbol{\sigma}^{el} - \mathbf{D}^{el} : \Delta\boldsymbol{\varepsilon}^{pl}, \tag{4.3.6–1}$$

where

$$\boldsymbol{\sigma}^{el} = \mathbf{D}^{el} : (\boldsymbol{\varepsilon}^{el}|_t + \Delta\boldsymbol{\varepsilon})$$

is the elastic predictor and

$$\mathbf{D}^{el} = 2G\,\mathfrak{S} + (\boldsymbol{K} - \frac{\mathbf{2}}{\mathbf{3}}\boldsymbol{G})\,\mathbf{I}\,\mathbf{I}$$

is the linear isotropic elasticity tensor with $\boldsymbol{G}$ and $\boldsymbol{K}$ being the shear and bulk modulus, respectively, and $\mathfrak{S}$ and $\mathbf{I}$ being the fourth- and second-order identity tensors, respectively. Also, in the above, the additive decomposition of strain is used to write the total incremental strain as the sum of the elastic and plastic parts. All of the stress and state variables are evaluated at $t + \Delta t$ unless indicated otherwise.

The yield condition, the flow rule, and the evolution of the state variables are rewritten as

$$\phi(\boldsymbol{\sigma}, H^\alpha) = 0,$$

$$\Delta\boldsymbol{\varepsilon}^{pl} = \frac{1}{3}\Delta\varepsilon_p\,\mathbf{I} + \Delta\varepsilon_q\,\mathbf{n}, \tag{4.3.6–2}$$

and

$$\Delta H^\alpha = \bar{H}^\alpha(\Delta\boldsymbol{\varepsilon}^{pl}, \boldsymbol{\sigma}, H^\beta), \tag{4.3.6–3}$$

where

$$\mathbf{n} = \frac{3}{2q}\mathbf{S},$$
$$\Delta\varepsilon_p = -\Delta\lambda\frac{\partial\phi}{\partial p},$$
$$\Delta\varepsilon_q = \Delta\lambda\frac{\partial\phi}{\partial q}.$$

In the above H^α, $\alpha = 1, 2$ are the state variables $\bar{\varepsilon}^{pl}_m$ and $\boldsymbol{f}$, respectively. The plastic multiplier $\Delta\lambda$ is eliminated from the last two equations to give

$$\Delta\varepsilon_p \frac{\partial \phi}{\partial q} + \Delta\varepsilon_q \frac{\partial \phi}{\partial p} = 0.$$

Equation 4.3.6–2 is used in the elasticity Equation 4.3.6–1 to yield

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^{el} - K\Delta\varepsilon_p \mathbf{I} - 2G\Delta\varepsilon_q \mathbf{n}. \tag{4.3.6–4}$$

$\mathbf{S}^{el}$ and $\mathbf{n}$ are coaxial; therefore, $\mathbf{n}$ is determined as

$$\mathbf{n} = \frac{3}{2q^{el}}\mathbf{S}^{el}. \tag{4.3.6–5}$$

Once $\mathbf{n}$ is known, it is easily seen that consistent determination of the scalars $\Delta\varepsilon_p$ and $\Delta\varepsilon_q$ completes the solution. Therefore, the problem of integrating the pressure-dependent elastoplastic constitutive equations is reduced to solving the following two nonlinear equations for the scalars $\Delta\varepsilon_p$ and $\Delta\varepsilon_q$:

$$\phi(p, q, H^\alpha) = 0, \tag{4.3.6–6}$$

$$\Delta\varepsilon_p \frac{\partial \phi}{\partial q} + \Delta\varepsilon_q \frac{\partial \phi}{\partial p} = 0. \tag{4.3.6–7}$$

In the above equations $\boldsymbol{p}$, $\boldsymbol{q}$, and H^α are defined by

$$p = p^{el} + K\Delta\varepsilon_p, \tag{4.3.6–8}$$

$$q = q^{el} - 3G\Delta\varepsilon_q, \tag{4.3.6–9}$$

$$\Delta H^\alpha = h^\alpha(\Delta\varepsilon_p, \Delta\varepsilon_q, p, q, H^\beta), \tag{4.3.6–10}$$

where Equation 4.3.6–8 and Equation 4.3.6–9 are obtained by projecting Equation 4.3.6–4 onto $\mathbf{I}$ and $\mathbf{n}$, respectively, and Equation 4.3.6–10 is an alternate form of Equation 4.3.6–3. Solving the above system of equations for the unknowns p, q, $\Delta\varepsilon_p$, $\Delta\varepsilon_q$, and H^α completes the integration algorithm for the porous plasticity model.

Equation 4.3.6–6 and Equation 4.3.6–7 are solved for $\Delta\varepsilon_p$ and $\Delta\varepsilon_q$ using Newton's method; $\boldsymbol{p}$ and $\boldsymbol{q}$ are updated using Equation 4.3.6–8 and Equation 4.3.6–9; the state variables are updated using Equation 4.3.6–10.

Computing the linearization moduli

In the implicit finite element method of solving large-deformation problems, the discretized equilibrium equations result in a set of nonlinear equations for the nodal unknowns at the end of the increment. Abaqus/Standard uses Newton's method to solve these equations, which requires the computation of *linearization moduli*

$$\mathcal{J} = \frac{\partial \Delta \boldsymbol{\sigma}}{\partial \Delta \boldsymbol{\varepsilon}} = \left(\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}} \right)_{t+\Delta t}.$$

To compute $\mathcal{J}$ (also known as the Jacobian), we start with the elasticity equations (Equation 4.3.6–4), which can be rewritten as

$$\boldsymbol{\sigma} = 2G(\mathbf{e}^{el}|_t + \Delta \mathbf{e} - \Delta \varepsilon_q \, \mathbf{n}) + K(\varepsilon^{el}_{kk}|_t + \Delta \varepsilon_{kk} - \Delta \varepsilon_p) \, \mathbf{I},$$

where $\mathbf{e} = \boldsymbol{\varepsilon} - (1/3)\varepsilon_{kk} \, \mathbf{I}$ is the deviatoric part of $\boldsymbol{\varepsilon}$. From the above equation you find that

$$\partial \boldsymbol{\sigma} = 2G(\partial \mathbf{e} - \partial \Delta \varepsilon_q \, \mathbf{n} - \Delta \varepsilon_q \, \partial \mathbf{n}) + K(\partial \varepsilon_{kk} - \partial \Delta \varepsilon_p) \mathbf{I}. \tag{4.3.6–11}$$

Once again Equation 4.3.6–6 and Equation 4.3.6–7 are used in computing the variations $\partial \Delta \varepsilon_p$ and $\partial \Delta \varepsilon_q$. After some lengthy algebraic calculations, a set of linear equations is obtained that can be solved for $\partial \Delta \varepsilon_p$ and $\partial \Delta \varepsilon_q$. These derivatives are substituted into Equation 4.3.6–11 to obtain the linearization moduli. In general this linearization moduli is not symmetric. Further details of the derivation of the Jacobian can be found in Aravas (1987).

Reference

- "Porous metal plasticity," Section 23.2.9 of the Abaqus Analysis User's Guide

4.3.7 CAST IRON PLASTICITY

Products: Abaqus/Standard Abaqus/Explicit

The cast iron plasticity constitutive model is intended for modeling the elastoplastic behavior of gray cast iron. In tension gray cast iron is more brittle than most metals. This brittleness is attributed to the microstructure of the material, which consists of a distribution of graphite flakes in a steel matrix. In tension the graphite flakes act as stress concentrators, leading to an overall decrease in mechanical properties (such as yield strength). In compression, on the other hand, the graphite flakes serve to transmit stresses, and the overall response is governed by the response of the steel matrix alone. The above differences manifest themselves in the following macroscopic properties: (i) different yield strengths in tension and compression, with the yield stress in compression being a factor of three or more higher than the yield stress in tension; (ii) inelastic volume change in tension, but little or no inelastic volume change in compression; and (iii) different hardening behavior in tension and compression. It is commonly accepted (Hjelm, 1992, 1994) that a Mises-type yield condition along with an associated flow rule models the material response sufficiently accurately under compressive loading conditions. This assumption is not true for tensile loading conditions: a pressure-dependent yield surface with nonassociated flow is required to model the brittle behavior in tension. The model is described in detail in the remainder of this section.

Strain rate decomposition

An additive strain rate decomposition is assumed:

$$\dot{\varepsilon} = \dot{\varepsilon}^{el} + \dot{\varepsilon}^{pl},$$

where $\dot{\varepsilon}$ is the total strain rate, $\dot{\varepsilon}^{el}$ is the elastic part of the strain rate, and $\dot{\varepsilon}^{pl}$ is the inelastic (plastic) part of the strain rate.

Elastic behavior

In compression the elastic behavior of gray cast iron is similar to that of many steels. It shows a well-defined elastic stiffness. In uniaxial tension the slope of the stress/strain curve decreases continuously, and it is difficult to estimate the elastic modulus from experimental results.

The model in Abaqus assumes that the elastic behavior of gray cast iron can be represented by linear isotropic elasticity, with the same stiffness in tension and compression.

Yield condition

The model makes use of a composite yield surface to describe the different behavior in tension and compression. In tension yielding is assumed to be governed by the maximum principal stress, while in compression yielding is assumed to be pressure-independent and governed by the deviatoric stresses alone. In principal stress space the composite yield surface consists of the Rankine cube in tension and the Mises cylinder in compression.

The material is assumed to be isotropic; hence, the yield surface can be expressed as a function of three invariant measures of the stress tensor: the equivalent pressure stress,

$$p = -\frac{1}{3}\boldsymbol{\sigma} : \mathbf{I};$$

the Mises equivalent stress,

$$q = \sqrt{\frac{3}{2}\,\mathbf{S} : \mathbf{S}};$$

and the third invariant of the deviatoric stress,

$$r = (\frac{9}{2}\mathbf{S} \cdot \mathbf{S} : \mathbf{S})^{\frac{1}{3}},$$

where $\mathbf{S}$ is the stress deviator, defined as

$$\mathbf{S} = p\mathbf{I} + \boldsymbol{\sigma},$$

$\boldsymbol{\sigma}$ is the Cauchy stress tensor, and $\mathbf{I}$ is the second-order identity tensor. It is convenient to combine the invariants q and r to define a nondimensional quantity, Θ, where

$$\cos(3\Theta) = (\frac{r}{q})^3.$$

In principal stress space the variable Θ identifies the meridional plane for a given stress state.

On any given meridional plane the yield surface consists of two distinct line segments given by

$$F_t = R_R(\Theta)q - p - \sigma_t = 0$$

and

$$F_c = q - \sigma_c = 0.$$

In the expressions above $R_R(\Theta) = (2/3)\cos\Theta$; $\sigma_t(\bar{\varepsilon}_t^{pl}, \theta, f^\alpha)$ is the yield stress in uniaxial tension, which may depend on the equivalent plastic strain in uniaxial tension $\bar{\varepsilon}_t^{pl}$, temperature θ, and field variables f^α ($\alpha = 1, 2, ..$); and $\sigma_c(\bar{\varepsilon}_c^{pl}, \theta, f^\alpha)$ is the yield stress in uniaxial compression, which may depend on the equivalent plastic strain in uniaxial compression $\bar{\varepsilon}_c^{pl}$, temperature θ, and field variables f^α ($\alpha = 1, 2, ..$). The composite yield surface is illustrated in the meridional plane in Figure 4.3.7–1, in the deviatoric plane in Figure 4.3.7–2, and in the principal stress space in Figure 4.3.7–3.

Flow rule

For the purposes of discussing the flow and hardening behavior, it is useful to divide a meridional plane into two regions as shown in Figure 4.3.7–4. The region to the left of the uniaxial compression line

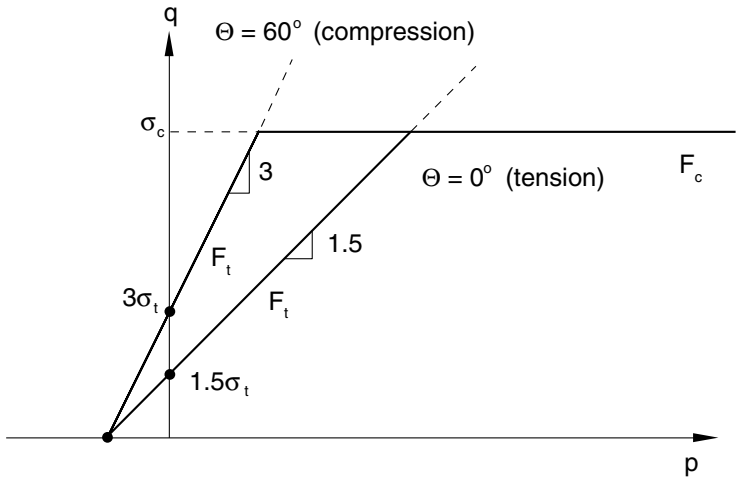


Figure 4.3.7–1 Schematic of the yield surface in the meridional plane.

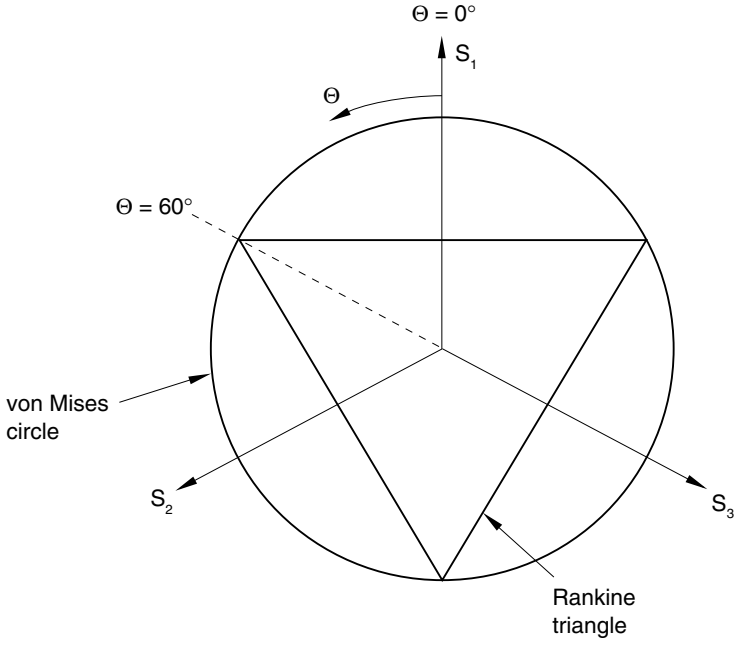


Figure 4.3.7–2 Schematic of the yield surface in the deviatoric plane.

(labeled UC) is referred to as the "tensile region," and the region to the right of the uniaxial compression line is referred to as the "compressive region."

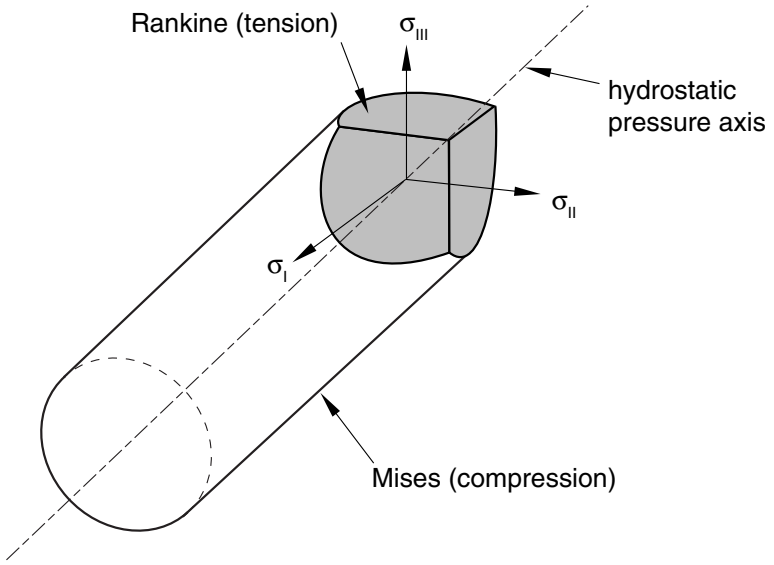


Figure 4.3.7–3 Schematic of the yield surface in the principal stress space.

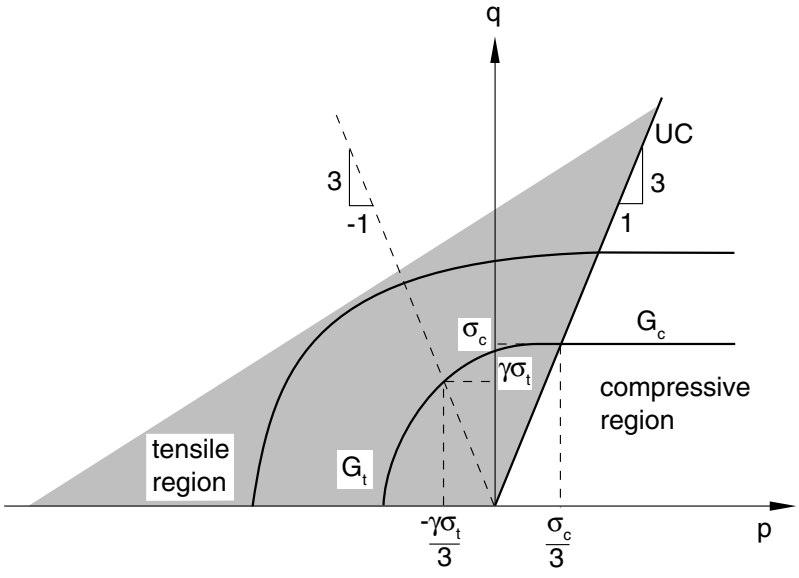


Figure 4.3.7–4 Schematic of the flow potentials in the meridional plane.

The plastic strains are defined to be normal to a family of self-similar flow potentials parametrized by the value of the potential G:

$$\gamma \dot{\boldsymbol{\varepsilon}}^{pl} = \dot{\lambda} \frac{\partial G}{\partial \boldsymbol{\sigma}},$$

where $\dot{\lambda}$ is the nonnegative plastic multiplier. The flow potential, $G(p, q, \theta, f^{\alpha})$, is assumed to be independent of the third stress invariant (independent of Θ). It consists of the Mises cylinder in compression with an ellipsoidal "cap" in tension. The transition between the two surfaces is smooth. The projection of the flow potential onto the deviatoric plane is a circle. On the meridional plane the potential G can take one of two values, G_t and G_c, defined by the relations

$$\frac{(p - G_t)^2}{a^2} + q^2 = 9{G_t}^2 \quad \text{when} \quad p < \frac{\sigma_c}{3},$$

$$q = 3G_c \quad \text{when} \quad p \geq \frac{\sigma_c}{3}.$$

G_t is the value of the potential in the tensile region, and G_c is the value of the potential in the compressive region. The potential in the tensile region is of the form of an ellipse, where a is the ratio of the horizontal (p) to the vertical (q) axis of the ellipse. The shape of the ellipse is controlled by a, which is chosen such that it passes through the two points $(-\gamma\sigma_t/3, \gamma\sigma_t)$ and $(\sigma_c/3, \sigma_c)$, where γ is a material parameter that controls plastic dilatation. The above requirement determines a^2 to be equal to $(\sigma_c + \gamma\sigma_t)/9(\sigma_c - \gamma\sigma_t)$. The value of the potential, G, depends on the current stress point (p, q). In the compressive region the flow potential consists of the Mises straight line. A consequence of the above choice of the flow potential is that plastic flow results in inelastic volume change in the tensile region and no inelastic volume change in the compressive region. Figure 4.3.7–4 illustrates two potentials in the p–q plane.

It can be shown that for uniaxial tension, $\gamma = 2r_{ct}(1 + \nu_{pl})/3$, where $r_{ct} = \sigma_c/\sigma_t$ and ν_{pl} (referred to as the "plastic Poisson's ratio") is equal to the absolute value of the ratio of the transverse to the longitudinal plastic strain under uniaxial tension. The quantities σ_c, σ_t, and ν_{pl} are material properties that must be provided by the user. The plastic Poisson's ratio, ν_{pl}, is expected to be less than 0.5 since experiments suggest that there is a permanent increase in the volume of gray cast iron when it is loaded in uniaxial tension beyond yield. In addition, ν_{pl} must be greater than -1 for the potential to be well-defined. For the gray cast iron described in the work of Coffin (1950), the value of ν_{pl} is approximately 0.04. This estimate is based on the reported value of the permanent volume change close to the ultimate tensile stress. The value of ν_{pl} may change with plastic deformation. However, the model in Abaqus assumes that it is constant with respect to plastic deformation. It can depend on temperature and field variables.

Since the flow potential is different from the yield surface (nonassociated flow), the material Jacobian matrix is unsymmetric.

Hardening

Abaqus assumes that the graphite flakes do not influence the hardening behavior in the compressive region, where the matrix behavior (Mises plasticity) characterizes the overall response. In the tensile

region the plastic strain consists of both a volumetric part (corresponding to opening up of the graphite flakes) and a deviatoric part (corresponding to inelastic shearing of the matrix). In the limiting case of pure hydrostatic tension it can be expected that the matrix material does not respond plastically; therefore, the entire plastic strain is volumetric, corresponding to the opening up of the graphite flakes. Thus, in the tensile region, we expect that the deviatoric part of the plastic strain decreases as the loading path approaches hydrostatic tension, eventually becoming zero at hydrostatic tension, while the volumetric part of the plastic strain decreases as the loading path approaches uniaxial compression and is zero for uniaxial compression and confining pressures higher than $\sigma_c/3$.

The hardening of the model is controlled by the evolution of $\sigma_t = \sigma_t(\bar{\varepsilon}_t^{pl}, \theta, f^\alpha)$ and $\sigma_c = \sigma_c(\bar{\varepsilon}_c^{pl}, \theta, f^\alpha)$. To develop the hardening model, we need to find expressions for the equivalent plastic strains, $\bar{\varepsilon}_t^{pl}$ and $\bar{\varepsilon}_c^{pl}$, as functions of the plastic strain tensor $\boldsymbol{\varepsilon}^{pl}$. To simplify the discussion, we assume that the temperature θ and the field variables f^α ($\alpha = 1, 2, ..$) are constants, so that the yield stresses are taken to be functions of the corresponding equivalent strains only. Furthermore, it is convenient to decompose the plastic strain rate into volumetric and deviatoric components:

$$\dot{\varepsilon}_{vol}^{pl} = \dot{\boldsymbol{\varepsilon}}^{pl} : \mathbf{I},$$

$$\dot{e}^{pl} = \frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{pl} : \mathbf{n},$$

where $\mathbf{n} = (3/2q)\mathbf{S}$ is the flow direction in the deviatoric plane. To find an expression relating $\bar{\varepsilon}_c^{pl}$ to $\boldsymbol{\varepsilon}^{pl}$, we use the equivalent plastic work expression

$$\sigma_c \dot{\bar{\varepsilon}}_c^{pl} = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{pl}.$$

Using the definition of the flow potential, it then follows (given that $\dot{\varepsilon}_{vol}^{pl} = 0$ and $q = \sigma_c$ in compression) that

$$\dot{\bar{\varepsilon}}_c^{pl} = \dot{e}^{pl}.$$

This expression is used to update $\sigma_c(\bar{\varepsilon}_c^{pl})$ for all loading conditions, including tensile loading conditions. Next, we find an expression relating $\bar{\varepsilon}_t^{pl}$ to $\boldsymbol{\varepsilon}^{pl}$. Using the equivalent plastic work expression

$$\sigma_t \dot{\bar{\varepsilon}}_t^{pl} = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{pl}$$

and the definition of the flow potential, it follows that

$$\dot{\bar{\varepsilon}}_t^{pl} = \frac{1}{\sigma_t}(-p\dot{\varepsilon}_{vol}^{pl} + q\dot{e}^{pl}).$$

This expression is used to update the yield stress $\sigma_t(\bar{\varepsilon}_t^{pl})$ for all loading conditions, including compressive loading conditions.

Under predominantly compressive loading conditions, $\dot{\varepsilon}^{pl}_{vol} = 0$, $\dot{\bar{\varepsilon}}^{pl}_{c} = \dot{e}^{pl}$, and $\dot{\bar{\varepsilon}}^{pl}_{t} = (\sigma_c/\sigma_t)\dot{e}^{pl}$. Thus, both σ_c and σ_t are updated based on the deviatoric plastic strain, e^{pl}, alone. This behavior is depicted in Figure 4.3.7–5.

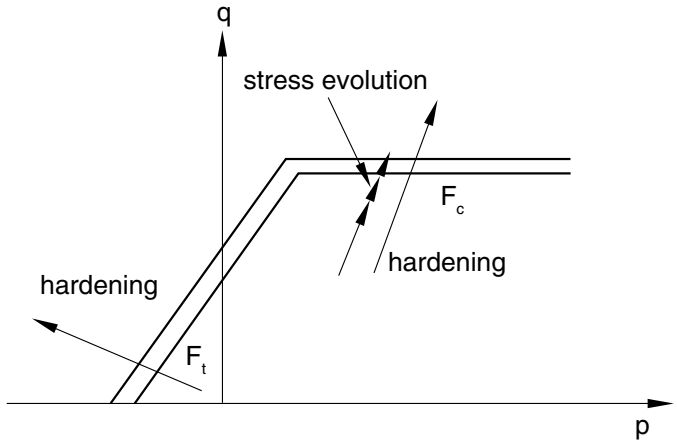


Figure 4.3.7–5 Hardening during compression loading.

At the other extreme, under hydrostatic tension (see Figure 4.3.7–6) $\dot{e}^{pl} = 0$.

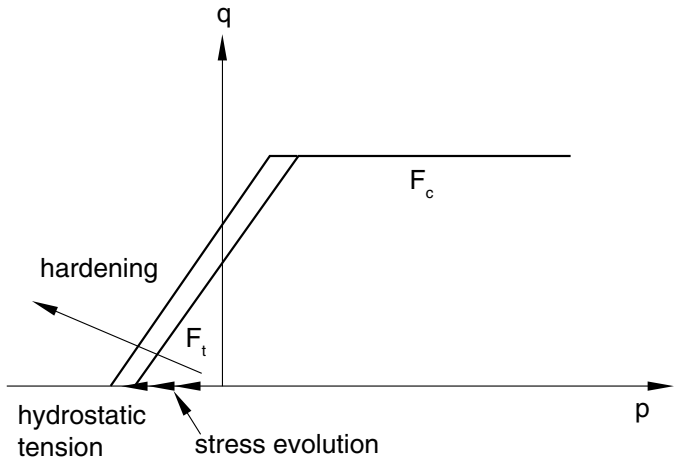


Figure 4.3.7–6 Hardening due to hydrostatic tension ($p < 0$ and $q = 0$).

Hence, $\dot{\bar{\varepsilon}}_c^{pl} = 0$, and $\dot{\bar{\varepsilon}}_t^{pl} = \dot{\varepsilon}_{vol}^{pl}$. Consequently, σ_c does not evolve, but σ_t evolves based on $\dot{\varepsilon}_{vol}^{pl}$. For all other loading paths in the tensile region, both $\dot{\varepsilon}_{vol}^{pl}$ and $\dot{e}^{pl}$ are nonzero. In such cases σ_c is updated based on the deviatoric plastic strain, e^{pl}, alone, while σ_t is updated based on both ε_{vol}^{pl} and e^{pl} (see Figure 4.3.7–5). Given the micromechanics of deformation of gray cast iron, the behavior seems reasonable, although there appears to be no experimental evidence confirming the above hardening behavior.

Since the matrix behavior of gray cast iron is similar to that of steel, it is reasonable to expect that under reversal of loading a combined kinematic and "quasi-isotropic" hardening scheme may be more appropriate to model, for example, the Bauschinger effect. Also, under compressive loading following tensile loading, we expect that some of the voids that had opened up at the graphite flakes may close. Thus, a cap in the yield surface may be appropriate under high confining pressures. However, only limited information is available in the literature. The model in Abaqus does not address these issues related to loading reversals and, therefore, should be used only for essentially monotonic loading conditions.

Reference

- "Cast iron plasticity," Section 23.2.10 of the Abaqus Analysis User's Guide

4.3.8 ORNL CONSTITUTIVE THEORY

Product: Abaqus/Standard

This section describes the constitutive theory for Types 304 and 316 stainless steel in Abaqus/Standard, as set forth in nuclear standard NE F9–5T(1981). The constitutive theory is uncoupled into a rate-independent plasticity response and a rate-dependent creep response, each of which is governed by a separate constitutive law.

The plasticity theory uses a Mises yield surface that can expand isotropically and translate kinematically in stress space. Nuclear standard NE F9–5T provides for some coupling between the plasticity and creep responses by allowing prior creep strain to expand and translate the subsequent yield surface in stress space. For Types 304 and 316 stainless steel, however, prior plasticity does not change the subsequent creep response.

A set of auxiliary creep and load reversal detection rules, using modified strain hardening creep theory, overcomes the inconsistencies usually encountered with standard strain hardening theories under a stress reversal. In particular, creep theories based on strain hardening assumptions predict creep rates that are too small under conditions of stress reversal, so that the amount of creep that occurs under cyclic loading conditions will generally be underestimated.

ORNL plasticity theory of stainless steel

The plasticity theory for stainless steels, as set forth in nuclear standard NE F9–5T, employs a von-Mises yield surface with kinematic hardening. Ziegler's hardening rule, generalized to the nonisothermal case, is used in Abaqus.

Normally, when combined isotropic and kinematic hardening is considered, the center of the yield surface is assumed to translate linearly with plastic strain according to a Prager or Ziegler kinematic hardening rule. Incorporation of isotropic hardening into the constitutive formulation then changes the form of the stress-strain relation but leaves the Prager or Ziegler kinematic shift rule for determining the motion of the center of the yield surface intact. In the ORNL plasticity formulation the form of the stress-strain law is left intact (a bilinear representation in one dimension), and modifications to the kinematic shift rule are made to accommodate isotropic hardening.

The Mises yield surface is used:

$$f(\mathbf{S}-\boldsymbol{\alpha}) = \sqrt{\tfrac{3}{2}(\mathbf{S}-\boldsymbol{\alpha}):(\mathbf{S}-\boldsymbol{\alpha})} = \sigma^0(\bar{\varepsilon}^{pl}, \bar{\varepsilon}^H, \theta), \tag{4.3.8–1}$$

where $\mathbf{S}$ is the deviatoric stress

$$\mathbf{S} = \boldsymbol{\sigma} - \frac{1}{3}\mathbf{I}\mathbf{I}:\boldsymbol{\sigma}$$

and $\boldsymbol{\alpha}$ is the kinematic shift (the position of the center of the yield surface in stress space) that is taken to be purely deviatoric, so

$$\mathbf{I} : \boldsymbol{\alpha} = 0.$$

The quantity $\sigma^0(\overline{\varepsilon}^{pl}, \overline{\varepsilon}^{H}, \theta)$ is the equivalent uniaxial stress and denotes the distance in stress space from the center of the yield surface to any point on its surface. In the general case this yield surface radius is assumed to depend on the equivalent plastic strain, $\overline{\varepsilon}^{pl}$; the ORNL defined equivalent creep strain, $\overline{\varepsilon}^{H}$; and temperature, θ.

The model uses associated plastic flow, which means that the plastic strain rate is defined by

$$d\boldsymbol{\varepsilon}^{pl} = d\lambda \frac{\partial f}{\partial \boldsymbol{\sigma}}, \tag{4.3.8–2}$$

where $d\lambda$ is a scalar that can be identified as the work equivalent plastic strain rate, $d\overline{\varepsilon}^{pl}$, defined by

$$\sigma^0 \, d\overline{\varepsilon}^{pl} = (\boldsymbol{\sigma} - \boldsymbol{\alpha}) : d\boldsymbol{\varepsilon}^{pl}.$$

This is so because

$$\frac{\partial f}{\partial \boldsymbol{\sigma}} = \frac{3}{2}\frac{1}{\sigma^0}(\mathbf{S} - \boldsymbol{\alpha})$$

for the Mises yield surface chosen and, thus,

$$\sigma^0 \, d\overline{\varepsilon}^{pl} = d\lambda \frac{1}{\sigma^0}(\boldsymbol{\sigma} - \boldsymbol{\alpha}) : \frac{\partial f}{\partial \boldsymbol{\sigma}} = \sigma^0 \, d\lambda.$$

For continued satisfaction of the yield condition (Equation 4.3.8–1) during active plastic straining, the consistency condition is

$$\frac{\partial f}{\partial \boldsymbol{\sigma}} : (d\boldsymbol{\sigma} - d\boldsymbol{\alpha}) - \frac{\partial \sigma^0}{\partial \overline{\varepsilon}^{pl}} d\overline{\varepsilon}^{pl} - \frac{\partial \sigma^0}{\partial \theta} d\theta = 0. \tag{4.3.8–3}$$

For the evolution of $\boldsymbol{\alpha}$ we use the modified form of Ziegler's hardening rule,

$$d\boldsymbol{\alpha} = C\frac{1}{\sigma^0}(\boldsymbol{\sigma} - \boldsymbol{\alpha}) d\overline{\varepsilon}^{pl} - \frac{1}{\sigma^0}(\boldsymbol{\sigma} - \boldsymbol{\alpha})\frac{\partial \sigma^0}{\partial \overline{\varepsilon}^{pl}} d\overline{\varepsilon}^{pl} + \boldsymbol{\alpha}\frac{1}{C}\frac{dC}{d\theta} d\theta. \tag{4.3.8–4}$$

The second term in this equation accounts for isotropic hardening, so when this equation is combined with the consistency condition (Equation 4.3.8–3), the terms containing $\partial\sigma^0/\partial\overline{\varepsilon}^{pl}$ cancel. This produces a uniaxial stress-strain relation that is bilinear.

The strain rate decomposition during active plastic loading is

$$d\boldsymbol{\varepsilon} = d\boldsymbol{\varepsilon}^{el} + d\boldsymbol{\varepsilon}^{pl}$$

(we assume there is no change in the creep strain during active plastic loading). Here $d\boldsymbol{\varepsilon}^{el}$ is the elastic strain increment, and $d\boldsymbol{\varepsilon}$ is the total strain increment.

We use linear elasticity with temperature-dependent moduli, so

$$\boldsymbol{\sigma} = \mathbf{D}^{el} : \boldsymbol{\varepsilon}^{el},$$

where $\mathbf{D}^{el}(\theta)$ is the elasticity matrix. This gives the rate form

$$d\boldsymbol{\sigma} = \mathbf{D}^{el} : d\boldsymbol{\varepsilon}^{el} + \mathbf{g}^{el}\, d\theta,$$

where

$$g^{el} \stackrel{\text{def}}{=} \frac{\partial \mathbf{D}^{el}}{\partial \theta} : \boldsymbol{\varepsilon}^{el}.$$

Combining this with the flow rule (Equation 4.3.8–2) allows the incremental stress-strain relation to be written in the form

$$d\boldsymbol{\sigma} = \mathbf{D}^{el} \left(d\boldsymbol{\varepsilon} - d\overline{\varepsilon}^{pl} \frac{\partial f}{\partial \boldsymbol{\sigma}} \right) + \mathbf{g}^{el}\, d\theta. \tag{4.3.8–5}$$

Introducing the evolution equation for $\boldsymbol{\alpha}$ (Equation 4.3.8–4) into the consistency condition (Equation 4.3.8–3) provides

$$\frac{\partial f}{\partial \boldsymbol{\sigma}} : d\boldsymbol{\sigma} = C\, d\overline{\varepsilon}^{pl} + \frac{1}{C} \frac{dC}{d\theta} \frac{\partial f}{\partial \boldsymbol{\sigma}} : \boldsymbol{\alpha}\, d\theta + \frac{\partial \sigma^0}{\partial \theta}\, d\theta.$$

Projecting Equation 4.3.8–5 onto $\partial f / \partial \boldsymbol{\sigma}$ gives

$$\frac{\partial f}{\partial \boldsymbol{\sigma}} : d\boldsymbol{\sigma} = \frac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbf{D}^{el} : d\boldsymbol{\varepsilon}^{pl} - \frac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbf{D}^{el} : \frac{\partial f}{\partial \boldsymbol{\sigma}}\, d\overline{\varepsilon}^{pl} + \frac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbf{g}^{el}\, d\theta.$$

Combining these two definitions then provides $d\overline{\varepsilon}^{pl}$ from the total strain rate and temperature change rate as

$$d\overline{\varepsilon}^{pl} = \frac{1}{d} \mathbf{n} : \mathbf{D}^{el} : d\boldsymbol{\varepsilon} + \frac{1}{d} \left(\mathbf{n} : \mathbf{g}^{el} - \mathbf{n} : \boldsymbol{\alpha} \frac{1}{C} \frac{dC}{d\theta} - \frac{d\sigma^0}{d\theta} \right) d\theta,$$

where

$$\mathbf{n} \stackrel{\text{def}}{=} \frac{\partial f}{\partial \boldsymbol{\sigma}}$$

and

$$d \stackrel{\text{def}}{=} \mathbf{n} : \mathbf{D}^{el} : \mathbf{n} + C.$$

The incremental stress-strain relationship during active plastic deformation is now obtained directly from Equation 4.3.8–5 as

$$d\boldsymbol{\sigma} = \left[\mathbf{D}^{el} - \frac{1}{d}\mathbf{D}^{el} : \mathbf{n}\,\mathbf{n} : \mathbf{D}^{el}\right] : d\boldsymbol{\varepsilon} + \left\{\mathbf{g}^{el} - \frac{1}{d}\mathbf{D}^{el} : \mathbf{n}\left(\mathbf{n} : \mathbf{g}^{el} - \mathbf{n} : \boldsymbol{\alpha}\frac{1}{C}\frac{dC}{d\theta} - \frac{d\sigma^0}{d\theta}\right)\right\} d\theta. \tag{4.3.8–6}$$

Reduction of Equation 4.3.8–3 and Equation 4.3.8–4 to one dimension, with $\partial\sigma^0/\partial\overline{\varepsilon}^{pl} = 0$ at large strain values where $\overline{\varepsilon}^{pl} \rightarrow \infty$, shows that $C \overset{\text{def}}{=} d\sigma_{11}/d\varepsilon_{11}^{pl}$ is the slope of the plastic portion of the uniaxial stress versus plastic strain relation.

The incremental stress-strain relation (Equation 4.3.8–6) has the same form as that obtained with pure kinematic hardening. As mentioned previously, this is due to the cancellation of the terms $\partial\sigma^0/\partial\overline{\varepsilon}^{pl}$ in the expressions for the evolution of $\boldsymbol{\alpha}$ and the consistency relation. The bilinear stress-strain law and the translation of the yield surface center are depicted in Figure 4.3.8–1.

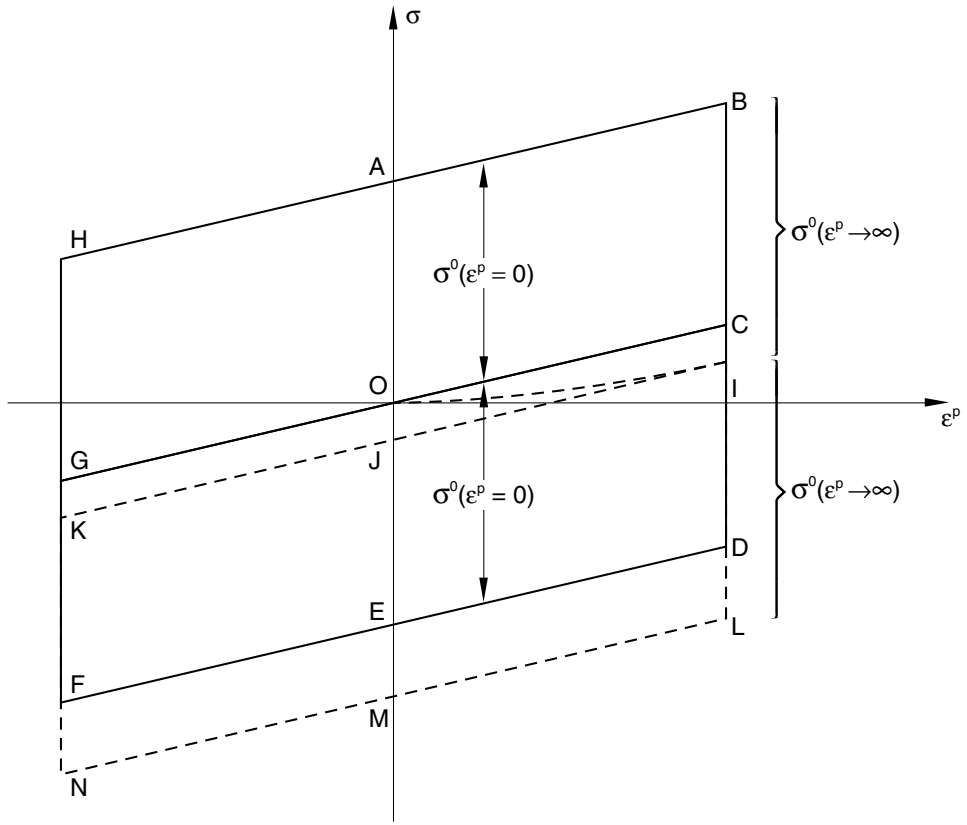


Figure 4.3.8–1 Bilinear stress-strain behavior and movement of the yield surface center with the ORNL plasticity theory.

In pure kinematic hardening under isothermal conditions the yield center follows the path OCOGOC for two stress reversals under fully reversed strain controlled conditions, while the stress-strain curve follows the path OABCDEFGHAB. In the ORNL theory the translation of the center of the yield surface is governed by Equation 4.3.8–6. The term $d\sigma^0/\overline{\varepsilon}^{pl}$ attains its largest value when $\overline{\varepsilon}^{pl} = 0$ and tends to zero under continued cumulative straining as $\overline{\varepsilon}^{pl} \to \infty$.

The yield surface center, thus, follows the path OI under monotonic loading. If point I is sufficiently distant from the origin, point O, so that $\partial\sigma^0/\partial\overline{\varepsilon}^{pl} \to 0$ at point I, then as $\overline{\varepsilon}^{pl}$ continues to grow under continued stress reversals, the continued growth of $\boldsymbol{\alpha}$ is governed by pure kinematic hardening, with $\partial\sigma^0/\partial\overline{\varepsilon}^{pl} = 0$. The yield surface center then translates along the path IJKJI. Unlike pure kinematic hardening the ORNL theory, which allows incremental changes in σ^0 due to cumulative plastic strain, produces a stress-strain curve that is asymmetric about the stress-strain origin.

The expansion of the yield surface is approximated by a step change in the value of σ^0. Figure 4.3.8–2 shows the value of σ^0 appropriate to the virgin (initial) stress-strain curve, $\sigma_0^0(\theta)$, and the value of σ^0 appropriate to the 10th cycle curve, $\sigma_{10}^0(\theta)$. The value of σ^0 is assumed to undergo a step change according to the relations

$$\begin{aligned} \sigma^0 &= \sigma_{10}^0(\theta) \quad \text{if } \boldsymbol{\alpha} : d\boldsymbol{\varepsilon}^{pl} \le 0 \text{ or if } \overline{\varepsilon}^H > 0.002, \\ \sigma^0 &= \sigma_0^0(\theta) \quad \text{if } \boldsymbol{\alpha} : d\boldsymbol{\varepsilon}^{pl} > 0 \text{ and } \overline{\varepsilon}^H \le 0.002. \end{aligned} \tag{4.3.8–7}$$

These conditions state that a step change in the size of the yield surface occurs when the value of the ORNL equivalent creep strain, $\overline{\varepsilon}^H$, reaches 0.2% or when yielding first occurs after stress reversal. Nuclear standard NE F9–5T recommends that the yield surface center remain fixed during the step change in σ^0. In this case the term involving $\partial\sigma^0/\partial\overline{\varepsilon}^{pl}$ in Equation 4.3.8–6 remains identically equal to zero and the yield surface center translates according to Ziegler's kinematic hardening rule. The nuclear standard recommends that the constant C in Equation 4.3.8–6 remain fixed on changing the value of σ^0 from the virgin to the 10th cycle values. Since C is the slope of the stress versus plastic strain in a uniaxial tensile test, this implies that the plastic tangent modulus of the 10th cycle stress-strain curve is the same as the plastic tangent modulus of the virgin stress-strain curve. This requirement is also necessary so that the 10th cycle stress-strain curve under fully reversed strain controlled loading is symmetric about the stress-strain origin.

ORNL creep theory for stainless steel

The flow rule for ORNL creep theory can be written in the form

$$\dot{\boldsymbol{\varepsilon}}^{cr} = \frac{3}{2}\frac{\mathbf{S}}{\overline{\sigma}}\dot{\overline{\varepsilon}}, \tag{4.3.8–8}$$

where $\dot{\overline{\varepsilon}}^{cr}(\overline{\sigma}, \overline{\varepsilon}^{cr}, \theta)$ is the uniaxial equivalent creep strain rate and $\overline{\sigma}$ is the Mises equivalent stress,

$$\overline{\sigma} \overset{\text{def}}{=} \sqrt{\tfrac{3}{2}\mathbf{S} : \mathbf{S}}.$$

Point 1 has coordinates (ε_1,σ_1) where $\sigma_1=E\varepsilon_1$.

Point 2 has coordinates (ε_2,σ_2).

$$C = \frac{\sigma_2-\sigma_1}{(\varepsilon_2-\sigma_2/E)} = \text{Slope of stress versus plastic strain}$$

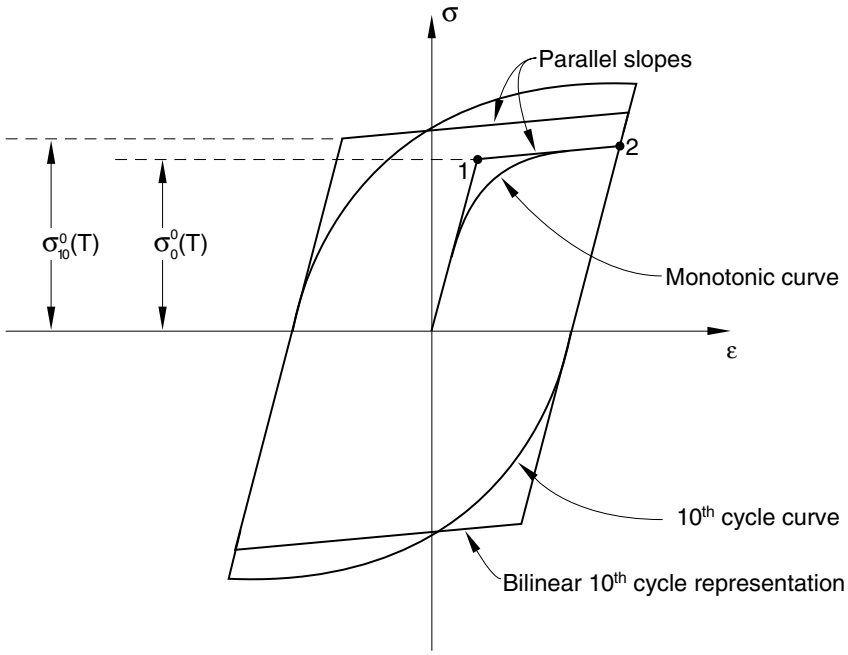


Figure 4.3.8–2 The virgin and 10th cycle ORNL stress-strain curves are assumed to have equal slopes in the plastic portion of the bilinear representation.

The functional dependence of $\dot{\bar{\varepsilon}}^{cr}$ on stress, creep strain, and temperature can be defined either in a user subroutine or by data lines. If the data line option is chosen, Abaqus assumes that the effective creep strain rate can be written as

$$\dot{\bar{\varepsilon}}^{cr} = A\bar{\sigma}^n t^m.$$

Under constant stress this gives

$$\bar{\varepsilon}^{cr} = \frac{1}{m+1} A\bar{\sigma}^n t^{m+1}.$$

Elimination of t between these equations gives the strain hardening form of the creep law,

$$\dot{\overline{\varepsilon}}^{cr} = \left[A\overline{\sigma}^{n}\left((m+1)\overline{\varepsilon}^{cr}\right)^{m}\right]^{1/m+1},$$

which is now assumed to be valid even when the stress changes with time.

The ORNL creep theory now replaces the total effective creep strain, $\overline{\varepsilon}^{cr}$, in this equation by an effective creep strain, $\overline{\varepsilon}^{H}$, determined according to the following algorithm.

At any time during the creep response the equivalent total creep strain used to define the equivalent creep strain rate is defined as the distance from an origin that is either $\boldsymbol{\varepsilon}^{+}$ or $\boldsymbol{\varepsilon}^{-}$. L indicates at any time which origin is active. If the origin is $\boldsymbol{\varepsilon}^{+}$, we set $L = 0$, while if the origin is $\boldsymbol{\varepsilon}^{-}$, $L = 1$. The quantity $\tilde{\varepsilon}$ defines the distance in total creep strain space between the current origin and the previous origin. The effective creep strain, $\overline{\varepsilon}^{H}$, is then determined by the following steps:

0. Initially set $\boldsymbol{\varepsilon}^{+} = \boldsymbol{\varepsilon}^{-} = \mathbf{0}$, set $\tilde{\varepsilon} = 0$ and set $L = 0$.
1. Set the current origin flag $M = L$.
2. Set $X = 10^{-6}$.
3. Compute

$$S^{+} = (\boldsymbol{\varepsilon}^{cr} - \boldsymbol{\varepsilon}^{+}) : \mathbf{S}, \quad S^{-} = (\boldsymbol{\varepsilon}^{cr} - \boldsymbol{\varepsilon}^{-}) : \mathbf{S},$$

$$G^{+} = \left(\tfrac{2}{3}(\boldsymbol{\varepsilon}^{cr} - \boldsymbol{\varepsilon}^{+}) : (\boldsymbol{\varepsilon}^{cr} - \boldsymbol{\varepsilon}^{+})\right)^{\frac{1}{2}}, \quad G^{-} = \left(\tfrac{2}{3}(\boldsymbol{\varepsilon}^{cr} - \boldsymbol{\varepsilon}^{-}) : (\boldsymbol{\varepsilon}^{cr} - \boldsymbol{\varepsilon}^{-})\right)^{\frac{1}{2}}.$$

If the current origin is $\boldsymbol{\varepsilon}^{+}$, the effective creep strain, G^{+}, is decreasing if $S^{+} < 0$. Similarly, if the current origin is $\boldsymbol{\varepsilon}^{-}$, the effective creep strain, G^{-}, is decreasing if $S^{-} < 0$. The rules for choosing a new origin follow in Steps 4 through 14, based on the concept of a decreasing effective creep strain (a load reversal).

4. If $G^{+} < X$ and $G^{-} < X$, set $N = 1$ and go to Step 13. Otherwise, set $N = 0$ and continue to Step 5.
5. If $S^{+} \geq 0$ and $S^{-} < 0$, set $N = 1$ and go to Step 13. Otherwise, set $N = 0$ and continue to Step 6.
6. If $S^{+} > 0$ and $S^{-} \leq 0$, set $N = 1$ and go to Step 13. Otherwise, set $N = 0$ and continue to Step 7.
7. If $S^{+} < 0$ and $S^{-} \geq 0$, set $N = 2$ and go to Step 13. Otherwise, set $N = 0$ and continue to Step 8.
8. If $S^{+} \leq 0$ and $S^{-} \geq 0$, set $N = 2$ and go to Step 13. Otherwise, set $N = 0$ and continue to Step 9.
9. If $S^{+} > 0$ and $S^{-} > 0$ and $G^{-} > (G^{+} + X)$, set $N = 2$ and go to Step 13. Otherwise, set $N = 0$ and continue to Step 10.
10. If $S^{+} > 0$ and $S^{-} > 0$ and $G^{-} \leq (G^{+} + X)$, set $N = 1$ and go to Step 13. Otherwise set $N = 0$ and continue to Step 11.
11. If $S^{+} < 0$ and $S^{-} \leq 0$ and $G > (G^{+} + X)$, set $N = 1$ and go to Step 13. Otherwise, set $N = 0$ and continue to Step 12.
12. If $S^{+} < 0$ and $S^{-} < 0$ and $G^{-} \leq (G^{+} + X)$, set $N = 1$ and go to Step 13. Otherwise, set $N = 0$ and continue to Step 13.

13. If $N = 0$, go to Step 18. (In this case the load did not reverse during the current creep increment, and no updating of the origins is required.) If $N = 1$ or 2, continue to Step 14.
14. If $N = 1$, set $L = 0$. (In this case the new origin is $\boldsymbol{\varepsilon}^+$ and the origin flag is set equal to zero.)
    If $N = 2$, set $L = 1$. (In this case the new origin is $\boldsymbol{\varepsilon}^-$ and the origin flag is set equal to one.)
    If $L = M$, go to Step 18. (In this case the new origin is the same as the current origin. No load reversal has, therefore, taken place; and updating of the origins is not required.)
    If $L \neq M$, continue to Step 15. (In this case a load reversal has taken place.)
15. If $M = 0$, go to Step 16. (Current origin is $\boldsymbol{\varepsilon}^+$ and load reversal has taken place.)
    If $M = 1$, go to Step 17. (Current origin is $\boldsymbol{\varepsilon}^-$ and load reversal has taken place.)
16. If $G^- > \tilde{\varepsilon}$, leave $\boldsymbol{\varepsilon}^+$ at its current value, set $\boldsymbol{\varepsilon}^- = \boldsymbol{\varepsilon}^{cr}$, set $\tilde{\varepsilon} = G^+$, set $L = 1$, and go to Step 18. (New origin is now $\boldsymbol{\varepsilon}^-$.)
    If $G^+ \leq \tilde{\varepsilon}$, leave $\boldsymbol{\varepsilon}^-$, $\boldsymbol{\varepsilon}^+$, and $\tilde{\varepsilon}$ at their current values; set $L = 1$; and go to Step 18. (New origin is now $\boldsymbol{\varepsilon}^+$.)
17. If $G^- > \tilde{\varepsilon}$, leave $\boldsymbol{\varepsilon}^-$ at its current value, set $\boldsymbol{\varepsilon}^+ = \boldsymbol{\varepsilon}^{cr}$, set $\tilde{\varepsilon} = G^+$, set $L = 0$, and continue to Step 18. (New origin is now $\boldsymbol{\varepsilon}^+$.)
    If $G^- \leq \tilde{\varepsilon}$, leave $\boldsymbol{\varepsilon}^-$, $\boldsymbol{\varepsilon}^+$, and $\tilde{\varepsilon}$ at their current values; set $L = 1$; and continue to Step 18. New origin is now $\boldsymbol{\varepsilon}^+$.)
18. If $L = 0$, set $\overline{\varepsilon}^H = \left[2/3(\boldsymbol{\varepsilon}^{cr} - \boldsymbol{\varepsilon}+) : (\boldsymbol{\varepsilon}^{cr} - \boldsymbol{\varepsilon}^+)\right]^{\frac{1}{2}}$.
    If $L = 1$, set $\overline{\varepsilon}^H = \left[2/3(\boldsymbol{\varepsilon}^{cr} - \boldsymbol{\varepsilon}^-) : (\boldsymbol{\varepsilon}^{cr} - \boldsymbol{\varepsilon}^-)\right]^{\frac{1}{2}}$.
19. Go to Step 1 to determine $\overline{\varepsilon}^H$ for the next creep increment.

In addition to the preceding algorithm for the determination of the effective creep strain, $\overline{\varepsilon}^H$, in the strain hardening creep formulation, the ORNL procedures also allow for a translation of the center of the yield surface during creep. Nuclear standard NE F9–5T recommends that the center of the yield surface be shifted during creep according to the relation

$$\dot{\boldsymbol{\alpha}} = H\dot{\boldsymbol{\varepsilon}}^{cr},$$

where

$$H = C \quad \text{if } \sqrt{\tfrac{3}{2}\boldsymbol{\alpha} : \boldsymbol{\alpha}} \leq A\sigma^0$$
$$H = 0 \quad \text{if } \sqrt{\tfrac{3}{2}\boldsymbol{\alpha} : \boldsymbol{\alpha}} > A\sigma^0,$$

the constant C being the slope of the uniaxial stress versus plastic strain curve and σ^0 the current effective yield stress, and A is the saturation rate for the kinematic shift caused by creep strain.

Reference

- "ORNL – Oak Ridge National Laboratory constitutive model," Section 23.2.12 of the Abaqus Analysis User's Guide

4.3.9 DEFORMATION PLASTICITY

Product: Abaqus/Standard

Deformation plasticity theory is provided in Abaqus/Standard to allow fully plastic analysis of ductile metals, usually under small-displacement conditions, for fracture mechanics applications. The model is based on the Ramberg-Osgood relationship. In this section the detailed constitutive model is defined. The procedure for obtaining fully plastic solutions generally consists of incremental loading until the response is fully plastic.

The model is termed deformation plasticity because the stress is defined by the total mechanical strain with no history dependence. There is no "unloading" criterion (to allow recovery of the initial elastic stiffness immediately after a strain reversal), so that the model is only useful as a plasticity model in cases of continuous flow. It is, in fact, a nonlinear elastic model; but at a limit state when all of a specimen or structure is responding plastically, this model is a useful equivalent representation of the plastic response because it has such a simple form.

One-dimensional model

The basic one-dimensional model is

$$E\,\varepsilon = \sigma + \alpha \left(\frac{|\sigma|}{\sigma_0} \right)^{n-1} \sigma, \tag{4.3.9–1}$$

where σ is the stress, ε is the mechanical strain, $\boldsymbol{E}$ is Young's modulus (defined as the slope of the stress-strain curve at zero stress), α is the "yield" offset (in the sense that, when $\sigma = \sigma_0$, $\varepsilon = (1+\alpha)\sigma_0/E$), and $\boldsymbol{n}$ is the hardening exponent for the "plastic" (nonlinear) term: $n > 1$.

The material behavior described by this model is nonlinear at all stress levels, but for commonly used values of the hardening exponent ($n \sim 5$ or more) the nonlinearity becomes significant only at stress magnitudes approaching or exceeding σ_0.

Multiaxial generalization

A linear "elastic" relation is used to generalize the first term of Equation 4.3.9–1; the nonlinear term is generalized to multiaxial stress states through the use of the Mises stress potential and associated flow law, giving the multiaxial model

$$E\,\boldsymbol{\varepsilon} = (1+\nu)\,\mathbf{S} - (1-2\nu)p\,\mathbf{I} + \frac{3}{2}\alpha \left(\frac{q}{\sigma_0} \right)^{n-1} \mathbf{S}, \tag{4.3.9–2}$$

where

$\boldsymbol{\varepsilon}$	is the strain tensor,
$\boldsymbol{\sigma}$	is the stress tensor,
$p = -\frac{1}{3}\boldsymbol{\sigma} : \mathbf{I}$	is the equivalent hydrostatic stress,

$q = \sqrt{\frac{3}{2}\mathbf{S} : \mathbf{S}}$ is the Mises equivalent stress,
$\mathbf{S} = \boldsymbol{\sigma} + p\,\mathbf{I}$ is the stress deviator, and
ν is Poisson's ratio.

The linear part of the behavior may be compressible or incompressible depending on the value of Poisson's ratio, but the nonlinear part of the behavior is incompressible (because the flow is normal to the Mises stress potential). Since the model will generally be used for cases when the deformation is dominated by plastic flow, the use of selectively reduced integration elements or "hybrid" (mixed formulation) elements is recommended with this material model (except in plane stress).

Strain energy density

The model is often used to obtain fully plastic solutions for fracture mechanics when the *J*-integral is needed. For evaluation of the *J*-integral the strain energy density is required. This is

$$W = \int \boldsymbol{\sigma} : d\boldsymbol{\varepsilon}.$$

From Equation 4.3.9–2 this may be obtained as

$$W = \frac{1+\nu}{3E}\,q^2 + \frac{3}{2}\,\frac{1-2\nu}{E}\,p^2 + \frac{n}{n+1}\,\frac{\alpha}{E\sigma_0^{n-1}}\,q^{n+1}.$$

Stress solution

During an analysis at each integration point the latest estimate of the kinematic solution is provided to the constitutive routines, which must provide the corresponding stress tensor calculated for the material model being used. Since this material model is nonlinear, we solve for the stresses by using the methods described below.

The uniaxial case (the only nonzero stress component is one direct stress)

In this case

$$E\,\varepsilon = \sigma + \alpha \left(\frac{q}{\sigma_0}\right)^{n-1} \sigma, \tag{4.3.9–3}$$

where now $q = |\sigma|$.

We solve Equation 4.3.9–3 for σ using Newton's method. Writing c_σ as the correction to σ, the Newton equations for Equation 4.3.9–3 are

$$\left[1 + n\alpha \left(\frac{q}{\sigma_0}\right)^{n-1}\right] c_\sigma = E\,\varepsilon - \sigma - \alpha \left(\frac{q}{\sigma_0}\right)^{n-1} \sigma,$$

$$\sigma = \sigma + c_\sigma .$$

As an initial guess we use $\sigma = E|\varepsilon|$ if $E|\varepsilon| \le \sigma_0$ and $\sigma = \pm[E|\varepsilon|\sigma_0^{n-1}/\alpha]^{1/n}$ (with the sign chosen as the same sign as ε) if $E|\varepsilon| > \sigma_0$.

In this case the material stiffness matrix is

$$\frac{\partial \sigma}{\partial \varepsilon} = \frac{E}{1 + n\alpha\,(q/\sigma_0)^{n-1}}.$$

All strain components defined by the kinematic solution

When all strain components are defined kinematically (that is, all cases except uniaxial and plane stress), projecting Equation 4.3.9–2 onto $\mathbf{I}$ gives

$$p = -\frac{E}{3(1-2\nu)}\,\mathbf{I} : \boldsymbol{\varepsilon},$$

a linear relationship for the volumetric behavior. Defining the deviatoric strain as

$$\mathbf{e} = \boldsymbol{\varepsilon} - \frac{1}{3}\,\mathbf{I}\,\mathbf{I} : \boldsymbol{\varepsilon}$$

and using Equation 4.3.9–2, we find

$$E\,\mathbf{e} = \left(1 + \nu + \frac{3}{2}\alpha\left(\frac{q}{\sigma_0}\right)^{n-1}\right)\mathbf{S}. \tag{4.3.9–4}$$

Defining the equivalent deviatoric strain as

$$\bar{e} = \sqrt{\frac{2}{3}\mathbf{e} : \mathbf{e}} \tag{4.3.9–5}$$

and using Equation 4.3.9–4, we obtain the scalar equation

$$E\,\bar{e} = \frac{2}{3}(1+\nu)q + \alpha\left(\frac{q}{\sigma_0}\right)^{n-1} q. \tag{4.3.9–6}$$

This equation is solved for q using Newton's method:

$$\left[\frac{2}{3}(1+\nu) + n\,\alpha\left(\frac{q}{\sigma_0}\right)^{n-1}\right] c_q = E\,\bar{e} - \frac{2}{3}(1+\nu)q - \alpha\left(\frac{q}{\sigma_0}\right)^{n-1} q,$$

$$q = q + c_q .$$

As for the uniaxial case we will use the starting guesses:

$$q = \frac{3}{2}\frac{E}{1+\nu}\,\bar{e} \qquad \text{if} \qquad \frac{3}{2}\frac{E}{1+\nu}\,\bar{e} \le \sigma_0,$$

and

$$q = \left(\frac{\sigma_0^{n-1} E\,\bar{e}}{\alpha}\right)^{1/n} \qquad \text{if} \qquad \frac{3}{2}\frac{E}{1+\nu}\,\bar{e} > \sigma_0.$$

Equation 4.3.9–6 can also be used to define

$$1 + \nu + \frac{3}{2}\alpha\left(\frac{q}{\sigma_0}\right)^{n-1} = \frac{3}{2}E\frac{\bar{e}}{q},$$

so that Equation 4.3.9–4 becomes

$$\mathbf{S} = \frac{2}{3}\,\frac{q}{\bar{e}}\,\mathbf{e}. \tag{4.3.9–7}$$

Thus, once $\boldsymbol{q}$ is known, $\mathbf{S}$ is defined; and, hence, $\boldsymbol{\sigma}$ is known as

$$\boldsymbol{\sigma} = \mathbf{S} - p\,\mathbf{I}.$$

The material stiffness is defined as follows. From Equation 4.3.9–7 we have

$$\partial\mathbf{S} = \frac{2}{3}\frac{q}{\bar{e}}\left[\partial\mathbf{e} + \left(\frac{\partial q}{q} - \frac{\partial\bar{e}}{\bar{e}}\right)\mathbf{e}\right]; \tag{4.3.9–8}$$

and Equation 4.3.9–6 gives

$$\partial q = \frac{E}{\frac{2}{3}(1+\nu) + n\alpha\,(q/\sigma_0)^{n-1}}\,\partial\bar{e};$$

and from Equation 4.3.9–5,

$$\partial\bar{e} = \frac{2}{3}\,\frac{\mathbf{e}}{\bar{e}} : \partial\mathbf{e}.$$

Using these results, Equation 4.3.9–8 becomes

$$\partial\mathbf{S} = \frac{2}{3}\frac{q}{\bar{e}}\left[\Im - \frac{2}{3\bar{e}}\left\{\frac{1}{\bar{e}} - \frac{1}{\bar{e} + (n-1)\alpha\,(q/\sigma_0)^{n-1}\,(q/E)}\right\}\mathbf{e}\,\mathbf{e}\right] : \partial\mathbf{e}.$$

Now

$$\partial\boldsymbol{\sigma} = \partial\mathbf{S} - \partial p\,\mathbf{I},$$

$$\partial p = -\frac{E}{3(1-2\nu)}\mathbf{I} : \partial\boldsymbol{\varepsilon},$$

and

$$\partial \mathbf{e} = \left(\Im - \frac{1}{3}\mathbf{I}\,\mathbf{I}\right) : \partial\boldsymbol{\varepsilon}.$$

Combining these results, we obtain the material stiffness as

$$\partial\boldsymbol{\sigma} = \left[\frac{2}{3}\frac{q}{\bar{e}}\left(\Im - \frac{2}{3\bar{e}}\left\{\frac{1}{\bar{e}} - \frac{1}{\bar{e} + (n-1)\alpha\,(q/\sigma_0)^{n-1}\,(q/E)}\right\}\mathbf{e}\,\mathbf{e}\right) + \frac{1}{3}\left(\frac{E}{1-2\nu} - \frac{2}{3}\frac{q}{\bar{e}}\right)\mathbf{I}\,\mathbf{I}\right] : \partial\boldsymbol{\varepsilon}.$$

Plane stress

For this case we obtain a solution for $\boldsymbol{\sigma}$ assuming that the material is fully incompressible. We then use this solution as a starting guess for a Newton loop to find $\boldsymbol{\sigma}$. Finally, we compute the corresponding material stiffness matrix.

Incompressible approximation:

In this case $\nu = 1/2$ and $\varepsilon_{33} = -(\varepsilon_{11} + \varepsilon_{22})$, so that $\bar{e}$ is known and Equation 4.3.9–6 can be solved as before for $\boldsymbol{q}$. Equation 4.3.9–7 then defines $\mathbf{S}$, and the plane stress constraint requires that $\sigma_{33} = 0$, so

$$p = S_{33} = -\frac{2}{3}\frac{q}{\bar{e}}(\varepsilon_{11} + \varepsilon_{22});$$

and, hence, we obtain the initial estimate of the solution

$$\sigma_{11} = \frac{2}{3}\frac{q}{\bar{e}}(2\varepsilon_{11} + \varepsilon_{22});$$

$$\sigma_{22} = \frac{2}{3}\frac{q}{\bar{e}}(\varepsilon_{11} + 2\varepsilon_{22}),$$

and

$$\sigma_{12} = \frac{q}{3\bar{e}}\gamma_{12}.$$

Newton solution for the actual stresses:

Equation 4.3.9–2 defines the stresses, where, for this case,

$$p = -\frac{1}{3}(\sigma_{11} + \sigma_{22}),$$

$$\mathbf{S} = \boldsymbol{\sigma} + p\,\mathbf{I},$$

and

$$q = \sqrt{\frac{3}{2}\mathbf{S} : \mathbf{S}}.$$

Therefore, Equation 4.3.9–2 becomes

$$E\,\boldsymbol{\varepsilon} = \left(1 + \nu + \frac{3}{2}\alpha\left(\frac{q}{\sigma_0}\right)^{n-1}\right)\boldsymbol{\sigma} + 3\left(\nu + \frac{1}{2}\alpha\left(\frac{q}{\sigma_0}\right)^{n-1}\right)p\,\mathbf{I}. \tag{4.3.9–9}$$

These equations are solved using Newton's method:

$$\left[\left(1 + \nu + \frac{3}{2}\alpha\left(\frac{q}{\sigma_0}\right)^{n-1}\right)\Im + \frac{9}{4}\alpha(n-1)\left(\frac{q}{\sigma_0}\right)^{n-1}\frac{\mathbf{S}}{q}\frac{\mathbf{S}}{q} - \left(\nu + \frac{1}{2}\alpha\left(\frac{q}{\sigma_0}\right)^{n-1}\right)\mathbf{I}\mathbf{I}\right] : \mathbf{c}_\sigma$$

$$= E\,\boldsymbol{\varepsilon} - \left(1 + \nu + \frac{3}{2}\alpha\left(\frac{q}{\sigma_0}\right)^{n-1}\right)\boldsymbol{\sigma} - 3\left(\nu + \frac{1}{2}\alpha\left(\frac{q}{\sigma_0}\right)^{n-1}\right)p\,\mathbf{I},$$

$$\boldsymbol{\sigma} = \boldsymbol{\sigma} + \mathbf{c}_\sigma.$$

The material stiffness is directly available from Equation 4.3.9–9 as

$$E\left[\left(1 + \nu + \frac{3}{2}\alpha\left(\frac{q}{\sigma_0}\right)^{n-1}\right)\Im + \frac{9}{4}\alpha(n-1)\left(\frac{q}{\sigma_0}\right)^{n-1}\frac{\mathbf{S}}{q}\frac{\mathbf{S}}{q} - \left(\nu + \frac{1}{2}\alpha\left(\frac{q}{\sigma_0}\right)^{n-1}\right)\mathbf{I}\mathbf{I}\right]^{-1}.$$

Reference

- "Deformation plasticity," Section 23.2.13 of the Abaqus Analysis User's Guide

4.3.10 HEAT GENERATION CAUSED BY PLASTIC STRAINING

Products: Abaqus/Standard Abaqus/Explicit

Abaqus allows the introduction of the inelastic heat fraction, η, which defines heat generation caused by mechanical dissipation associated with plastic straining. This term can be introduced as a source of coupling for thermal-mechanical analysis. Such coupling might be important in a simulation in which extensive inelastic deformation is occurring fairly rapidly in a material whose mechanical properties are temperature dependent. If the process is very slow, the heat generated by the plastic deformation has time to dissipate; and uncoupled, isothermal, analysis is sufficient to model the process. If the process is extremely rapid, the heat has no time to diffuse; and uncoupled, adiabatic, analysis (in which each integration point is treated as if it is thermally insulated from its neighbors) is sufficient. Fully coupled thermal-stress analysis is required for cases that lie far enough from both extremes. This section defines the heat generation term caused by inelastic straining and describes how this term contributes to the overall Newton solution scheme.

The model assumes that plastic straining gives rise to a heat flux per unit volume of

$$r^{pl} = \eta \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{pl},$$

where r^{pl} is the heat flux that is added into the thermal energy balance; η is the inelastic heat fraction, which is assumed to be a constant; $\boldsymbol{\sigma}$ is the stress; and $\dot{\boldsymbol{\varepsilon}}^{pl}$ is the rate of plastic straining. For all the plasticity models in Abaqus, the plastic strain increment is written from the flow potential as

$$\dot{\boldsymbol{\varepsilon}}^{pl} = \dot{\varepsilon}^{pl}\mathbf{n},$$

where $\mathbf{n}$ is the flow direction (we assume $\mathbf{n} = \mathbf{n}(\boldsymbol{\sigma}, \varepsilon^{pl}, \theta)$, where θ is the temperature) and ε^{pl} is a scalar measure of plastic straining that is used as a hardening parameter in the yield surface and flow potential definitions in some of the plasticity models. In this case we consider isotropic hardening theories only: Abaqus provides only thermo-mechanical coupling for such models.

Abaqus generally uses a backward Euler scheme to integrate the plastic strain, so r^{pl} at the end of the increment is approximated as

$$r^{pl} = \frac{1}{2\Delta t}\eta \Delta \varepsilon^{pl} \mathbf{n} : (\boldsymbol{\sigma} + \boldsymbol{\sigma}_t),$$

where all quantities are evaluated at the end of the increment (at time $t + \Delta t$) except $\boldsymbol{\sigma}_t$. This notation is adopted throughout the remainder of this section. This term is used as the contribution to the thermal energy balance equation.

When Newton's method is used to solve the nonlinear equations, the coupling term gives rise to three contributions to the Jacobian matrix for the Newton method:

$$\frac{\partial r^{pl}}{\partial \theta}, \quad \frac{\partial r^{pl}}{\partial \boldsymbol{\varepsilon}}$$

from the thermal energy balance equation, and

$$\frac{\partial \boldsymbol{\sigma}}{\partial \theta}$$

from the mechanical equilibrium equation. The general form for these terms is now derived.

The mechanical constitutive model has the following general form. The elasticity defines the stresses by

$$\boldsymbol{\sigma} = \frac{\partial W}{\partial \boldsymbol{\varepsilon}^{el}}, \tag{4.3.10–1}$$

where $W = W(\boldsymbol{\varepsilon}^{el}, \theta)$ is the strain energy density potential and $\boldsymbol{\varepsilon}^{el}$ is the mechanical elastic strain. We implicitly assume that the elasticity is not fully incompressible, although the derivation is not significantly different if this is not the case, since the pressure stress will do no work in a fully incompressible material and so makes no contribution to the terms under discussion.

We assume that there is an additive strain rate decomposition that can be integrated to give

$$\boldsymbol{\varepsilon}^{el} = \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{th} - \boldsymbol{\varepsilon}^{pl}, \tag{4.3.10–2}$$

where $\boldsymbol{\varepsilon}$ is the total strain and $\boldsymbol{\varepsilon}^{th}$ is the strain caused by thermal expansion. In the constitutive models in Abaqus $\boldsymbol{\varepsilon}^{th} = \boldsymbol{\varepsilon}^{th}(\theta)$ only. This form of decomposition of the deformation depends on $\boldsymbol{\varepsilon}$ being measured as the integrated rate of deformation and on the elastic and thermal strains being small: this is true for the standard plasticity models provided in the program.

The plastic flow definition is integrated by the backward Euler method to give

$$\Delta \boldsymbol{\varepsilon}^{pl} = \Delta \varepsilon^{pl} \mathbf{n}. \tag{4.3.10–3}$$

Finally, assuming there is a single active yield surface or a single active flow surface, rate-independent models introduce a yield surface constraint, while rate-dependent models provide an integrated flow rate constraint, both of which are incorporated in the general form

$$f(\boldsymbol{\sigma}) = \overline{\sigma}, \tag{4.3.10–4}$$

where $f(\boldsymbol{\sigma})$ is a scalar stress function (for example, the Mises or Hill stress function for simple metal plasticity models) and $\overline{\sigma} = \sigma^0(\varepsilon^{pl}, \theta)$ is the yield stress for a rate-independent model, while $\overline{\sigma} = B\sigma^0$ for a rate-dependent model, where

$$B = B(\frac{\Delta \varepsilon^{pl}}{\Delta t}, \theta)$$

defines the rate effect from the average plastic strain rate over the increment. For example, by default, the rate-dependent plasticity model defines

$$\dot{\varepsilon}^{pl} = D\left(\frac{\tilde{q}}{\sigma^0} - 1\right)^n,$$

where $\tilde{q}$ is the Mises or Hill equivalent stress, and $D(\theta)$ and $n(\theta)$ are material parameters. Using the average plastic strain rate over the increment in this expression defines

$$B = 1 + \left(\frac{\Delta \varepsilon^{pl}}{D \Delta t} \right)^{1/n} .$$

Equation 4.3.10–1 to Equation 4.3.10–4 are a general definition of all of the standard isotropic hardening plasticity models integrated by the backward Euler method.

We now take variations of these equations with respect to all quantities at the end of the increment:

$$\partial \boldsymbol{\sigma} = \frac{\partial^2 W}{\partial \boldsymbol{\varepsilon}^{el} \partial \theta} \, \partial \theta + \frac{\partial^2 W}{\partial \boldsymbol{\varepsilon}^{el} \partial \boldsymbol{\varepsilon}^{el}} : \left(\partial \boldsymbol{\varepsilon} - \frac{\partial \boldsymbol{\varepsilon}^{th}}{\partial \theta} \, \partial \theta - \partial \boldsymbol{\varepsilon}^{pl} \right),$$

$$\partial \boldsymbol{\varepsilon}^{pl} = \partial \varepsilon^{pl} \, \mathbf{n} + \Delta \varepsilon^{pl} \left(\frac{\partial \mathbf{n}}{\partial \boldsymbol{\sigma}} : \partial \boldsymbol{\sigma} + \frac{\partial \mathbf{n}}{\partial \varepsilon^{pl}} \, \partial \varepsilon^{pl} + \frac{\partial \mathbf{n}}{\partial \theta} \, \partial \theta \right),$$

and

$$\mathbf{m} : \partial \boldsymbol{\sigma} = \frac{\partial \overline{\sigma}}{\partial \varepsilon^{pl}} \, \partial \varepsilon^{pl} + \frac{\partial \overline{\sigma}}{\partial \theta} \, \partial \theta,$$

where

$$\mathbf{m} = \frac{\partial f}{\partial \boldsymbol{\sigma}}.$$

For simplicity of notation we now define

$$\mathbf{D}^{el} = \frac{\partial^2 W}{\partial \boldsymbol{\varepsilon}^{el} \partial \boldsymbol{\varepsilon}^{el}},$$

$$\hat{\mathbf{n}} = \mathbf{n} + \Delta \varepsilon^{pl} \frac{\partial \mathbf{n}}{\partial \varepsilon^{pl}},$$

$$d = \mathbf{m} : \mathbf{D}^{el} : \hat{\mathbf{n}} + \frac{\partial \overline{\sigma}}{\partial \varepsilon^{pl}},$$

$$\mathbf{Z} = \mathbf{I} - \frac{1}{d} \hat{\mathbf{n}} \, \mathbf{m} : \mathbf{D}^{el},$$

$$\mathbf{H} = \mathbf{I} + \Delta \varepsilon^{pl} \mathbf{D}^{el} : \mathbf{Z} : \frac{\partial \mathbf{n}}{\partial \boldsymbol{\sigma}},$$

$$\mathbf{g} = \frac{\partial^2 W}{\partial \boldsymbol{\varepsilon}^{el} \partial \theta} - \mathbf{D}^{el} : \left(\frac{\partial \boldsymbol{\varepsilon}^{th}}{\partial \theta} + \Delta \varepsilon^{pl} \frac{\partial \mathbf{n}}{\partial \theta} \right),$$

$$\tilde{\mathbf{n}} = \mathbf{n} + (\boldsymbol{\sigma} + \boldsymbol{\sigma}_t) : \frac{\partial \mathbf{n}}{\partial \boldsymbol{\sigma}},$$

$$\tilde{g} = \mathbf{m} : \mathbf{g} - \frac{\partial \overline{\sigma}}{\partial \theta},$$

$$\mathbf{h} = \mathbf{H}^{-1} : \left\{ \mathbf{g} - \frac{\tilde{g}}{d} \mathbf{D}^{el} : \mathbf{n} \right\},$$

and

$$\mathbf{D} = \mathbf{H}^{-1} : \mathbf{D}^{el} : \mathbf{Z}.$$

These expressions allow us to write

$$\partial \boldsymbol{\sigma} = \mathbf{D} : \partial \boldsymbol{\varepsilon} + \mathbf{h}\, \partial \theta,$$

and

$$\begin{aligned} \partial r^{pl} = & \frac{1}{2\Delta t} \eta \left\{ \Delta \varepsilon^{pl} \tilde{\mathbf{n}} : \mathbf{D} + \frac{1}{d} (\boldsymbol{\sigma} + \boldsymbol{\sigma}_t) : \hat{\mathbf{n}}\, \mathbf{m} : \mathbf{D}^{el} : \left[\mathbf{I} - \Delta \varepsilon^{pl} \frac{\partial \mathbf{n}}{\partial \boldsymbol{\sigma}} : \mathbf{D} \right] \right\} : \partial \boldsymbol{\varepsilon} \\ & + \frac{1}{2\Delta t} \eta \left\{ \frac{1}{d} (\boldsymbol{\sigma} + \boldsymbol{\sigma}_t) : \hat{\mathbf{n}} \left(\tilde{g} - \Delta \varepsilon^{pl} \mathbf{m} : \mathbf{D}^{el} : \frac{\partial \mathbf{n}}{\partial \boldsymbol{\sigma}} : \mathbf{h} \right) \right. \\ & \left. + \Delta \varepsilon^{pl} (\boldsymbol{\sigma} + \boldsymbol{\sigma}_t) : \frac{\partial \mathbf{n}}{\partial \theta} + \Delta \varepsilon^{pl} \left(\mathbf{n} + (\boldsymbol{\sigma} + \boldsymbol{\sigma}_t) : \frac{\partial \mathbf{n}}{\partial \boldsymbol{\sigma}} \right) : \mathbf{h} \right\} \partial \theta. \end{aligned}$$

References

- “Fully coupled thermal-stress analysis,” Section 6.5.3 of the Abaqus Analysis User’s Guide
- “Fully coupled thermal-electrical-structural analysis,” Section 6.7.4 of the Abaqus Analysis User’s Guide
- “Adiabatic analysis,” Section 6.5.4 of the Abaqus Analysis User’s Guide

4.4 Plasticity for non-metals

- "Porous elasticity," Section 4.4.1
- "Models for granular or polymer behavior," Section 4.4.2
- "Critical state models," Section 4.4.3
- "Drucker-Prager/Cap model for geological materials," Section 4.4.4
- "Mohr-Coulomb model," Section 4.4.5
- "Models for crushable foams," Section 4.4.6

4.4.1 POROUS ELASTICITY

Product: Abaqus/Standard

The porous elasticity model in Abaqus/Standard is designed to be used in conjunction with plasticity models that allow plastic volume changes as described in "Models for granular or polymer behavior," Section 4.4.2, to "Models for crushable foams," Section 4.4.6. Use of the porous elasticity model without one of these plasticity options is not recommended. The model is based on the experimental observation that in porous materials during elastic (recoverable) straining, the change in the void ratio—e—and the change in the logarithm of the equivalent pressure stress—p—defined as

$$p = -\frac{1}{3}\text{trace}\,\boldsymbol{\sigma} = -\frac{1}{3}\boldsymbol{\sigma} : \mathbf{I},$$

are linearly related, so that in rate form,

$$de^{el} = -\kappa\, d(\ln(p)),$$

where κ is a material parameter. In this form the material has zero tensile strength. If the tensile strength p_t^{el} is nonzero, the equivalent relation is

$$de^{el} = -\kappa\, d\left(\ln(p + p_t^{el})\right), \tag{4.4.1–1}$$

which includes the special case of zero tensile strength ($p_t^{el} = 0$). It can be shown that, if the compressibility of the solid material is neglected, the volume change of a material sample is

$$J = \frac{1+e}{1+e_0}, \tag{4.4.1–2}$$

where e_0 is the initial void ratio. If we define the elastic void ratio from the elastic volume change according to the relationship

$$J^{el} = \frac{1+e^{el}}{1+e_0} \tag{4.4.1–3}$$

and then integrate the linear relation, the volumetric elasticity relationship is

$$\frac{\kappa}{1+e_0}\ln\left(\frac{p+p_t^{el}}{p_0+p_t^{el}}\right) = 1 - J^{el}, \tag{4.4.1–4}$$

where p_0 is the initial pressure stress, prescribed by initial conditions. Note that for a zero tensile strength material it is required that $p_0 > 0$. This equation can be inverted to yield

$$p = -p_t^{el} + \left(p_0 + p_t^{el}\right) \exp\left[\frac{1+e_0}{\kappa}(1 - \exp \varepsilon_{vol}^{el})\right], \tag{4.4.1–5}$$

as illustrated in Figure 4.4.1–1.

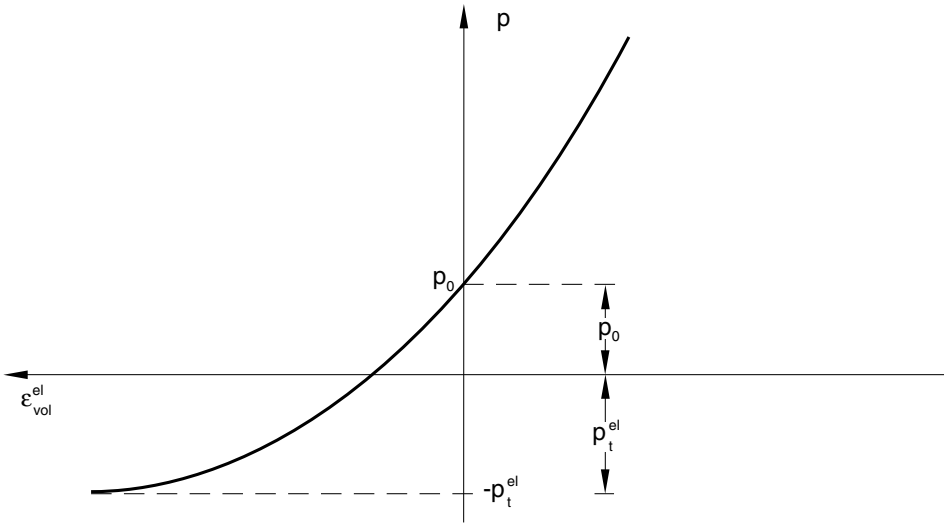


Figure 4.4.1–1 Porous elastic volumetric behavior.

The deviatoric elastic behavior is defined either by choosing a constant shear modulus, G, so that the deviatoric elastic stiffness is independent of the equivalent pressure stress or choosing a constant Poisson's ratio, ν, so that the deviatoric elastic stiffness increases as the equivalent pressure stress increases. If a constant shear modulus is given, the deviatoric relationship is

$$\mathbf{S} = 2G\mathbf{e}^{el},$$

whereas when Poisson's ratio is given, the relationship has the rate form

$$d\mathbf{S} = 2\hat{G}\, d\mathbf{e}^{el},$$

where

$$\hat{G} = \frac{3(1-2\nu)(1+e_0)}{2(1+\nu)\kappa}(p + p_t^{el}) \exp(\varepsilon_{vol}^{el}) \tag{4.4.1–6}$$

for a material with nonzero tensile strength. In these equations $\mathbf{S}$ is the deviatoric stress:

$$\mathbf{S} = \boldsymbol{\sigma} + p\,\mathbf{I},$$

and $\mathbf{e}^{el}$ is the deviatoric part of the elastic strain:

$$\mathbf{e}^{el} = \boldsymbol{\varepsilon}^{el} - \frac{1}{3}\varepsilon^{el}_{vol}\,\mathbf{I},$$

where ε^{el}_{vol} is the volumetric part of the elastic strain.

Reference

- "Elastic behavior of porous materials," Section 22.3.1 of the Abaqus Analysis User's Guide

4.4.2 MODELS FOR GRANULAR OR POLYMER BEHAVIOR

Products: Abaqus/Standard Abaqus/Explicit

The behavior of granular and polymeric materials is complex. However, under essentially monotonic loading conditions rather simple constitutive models provide useful design information. These constitutive models are essentially pressure-dependent plasticity models that have historically been popular in the geotechnical engineering field. However, more recently they have also been found to be useful for the modeling of some polymeric and composite materials that exhibit significantly different yield behavior in tension and compression.

The models described here are extensions of the original Drucker-Prager model (Drucker and Prager, 1952). In the context of geotechnical materials the extensions of interest include the use of curved yield surfaces in the meridional plane, the use of noncircular yield surfaces in the deviatoric stress plane, and the use of nonassociated flow laws. In the context of polymeric and composite materials, the extensions of interest are mainly the use of nonassociated flow laws and the inclusion of rate-dependent effects. In both contexts the models have been extended to include creep.

Available yield criteria

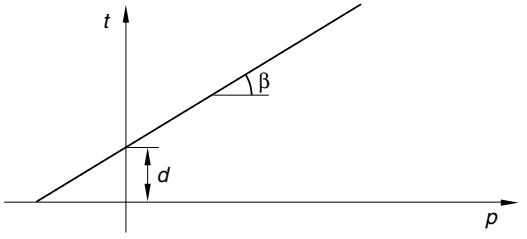
Three yield criteria are provided in this set of models. They offer differently shaped yield surfaces in the meridional plane (p–q plane): a linear form, a hyperbolic form, and a general exponent form (see Figure 4.4.2–1). The stress invariants used in the formulation are defined in "Conventions," Section 1.2.2 of the Abaqus Analysis User's Guide. The choice of model depends largely on the material, the experimental data available for calibration of the model parameters, and on the range of pressure stress values likely to be encountered.

The linear model (available in Abaqus/Standard and Abaqus/Explicit) provides a noncircular section in the deviatoric (Π) plane, associated inelastic flow in the deviatoric plane, and separate dilation and friction angles. The smoothed surface used in the deviatoric plane differs from a true Mohr-Coulomb surface that exhibits vertices. This has restrictive implications, especially with respect to flow localization studies for granular materials, but this may not be of major significance in many routine design applications. Input data parameters define the shape of the yield and flow surfaces in the deviatoric plane as well as the friction and dilation angles, so that a range of simple theories is provided; for example, the original Drucker-Prager model (Drucker and Prager, 1952) is available within this model.

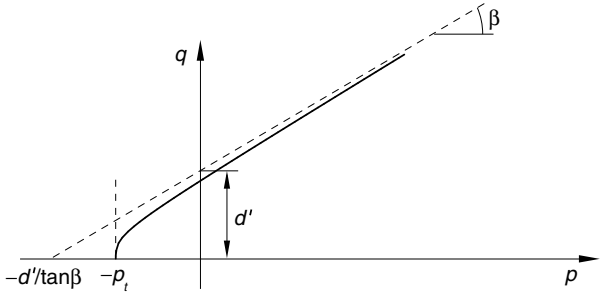
The hyperbolic and general exponent models (available in Abaqus/Standard only) use a von Mises (circular) section in the deviatoric stress plane with associated plastic flow. A hyperbolic flow potential is used in the meridional plane, which—in general—means nonassociated flow.

Hardening, rate dependence, and creep

Perfect plasticity as well as isotropic hardening are offered with these models. Isotropic hardening is generally considered to be a suitable model for problems in which the plastic straining goes well beyond



a) Linear Drucker-Prager: $F = t - p \tan \beta - d = 0$



b) Hyperbolic: $F = \sqrt{(d'|_0 - p_t|_0 \tan \beta)^2 + q^2} - p \tan \beta - d' = 0$



c) Exponent form: $F = aq^b - p - p_t = 0$

Figure 4.4.2–1 Yield criteria in the meridional plane.

the incipient yield state where the Bauschinger effect is noticeable (Rice, 1975). This hardening theory is, therefore, used for processes involving large plastic strain and in which the plastic strain rate does not continuously reverse direction sharply; that is, the models are intended for problems involving essentially monotonic loading, as distinct from cyclic loading.

The isotropic hardening models can be used for rate-dependent as well as rate-independent behavior. The rate-dependent version is intended for relatively high strain rate applications.

Isotropic hardening means that the yield function is written as

$$f(\boldsymbol{\sigma}) = \bar{\sigma}(\bar{\varepsilon}^{pl}, \dot{\bar{\varepsilon}}^{pl}, \theta, f^{\alpha}),$$

where $\boldsymbol{f}$ is an isotropic function of a symmetric second-order tensor and $\bar{\sigma}$ is the equivalent yield stress given by

$$\begin{aligned}
\bar{\sigma} &= \sigma_c(\bar{\varepsilon}^{pl}, \dot{\bar{\varepsilon}}^{pl}, \theta, f^{\alpha}) \quad \text{if hardening is defined by the uniaxial compression yield stress, } \sigma_c, \\
&= \sigma_t(\bar{\varepsilon}^{pl}, \dot{\bar{\varepsilon}}^{pl}, \theta, f^{\alpha}) \quad \text{if hardening is defined by the uniaxial tension yield stress, } \sigma_t, \\
&= d(\bar{\varepsilon}^{pl}, \dot{\bar{\varepsilon}}^{pl}, \theta, f^{\alpha}) \quad \text{if hardening is defined by the pure shear (cohesion) yield stress,}
\end{aligned}$$

where τ is the shear stress; $\boldsymbol{K}$ is a material parameter; $\bar{\varepsilon}^{pl}$ is the equivalent plastic strain; $\dot{\bar{\varepsilon}}^{pl}$ is the equivalent plastic strain rate; θ is temperature; and $f^{\alpha}, \alpha = 1, 2...$ are other predefined field variables.

The equivalent plastic strain rate, $\dot{\bar{\varepsilon}}^{pl}$, is defined for the linear Drucker-Prager model as

$$\begin{aligned}
\dot{\bar{\varepsilon}}^{pl} &= |\dot{\epsilon}_{11}^{pl}| \quad \text{if hardening is defined in uniaxial compression,} \\
&= \dot{\epsilon}_{11}^{pl} \quad \text{if hardening is defined in uniaxial tension,} \\
&= \frac{\dot{\gamma}^{pl}}{\sqrt{3}} \quad \text{if hardening is defined in pure shear,}
\end{aligned}$$

where $\dot{\gamma}^{pl}$ is the engineering shear plastic strain rate and is defined for the hyperbolic and exponential Drucker-Prager models by the plastic work expression

$$\bar{\sigma}\ \dot{\bar{\varepsilon}}^{pl} = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{pl}. \tag{4.4.2–1}$$

The functional dependence $\bar{\sigma}(\bar{\varepsilon}^{pl}, \dot{\bar{\varepsilon}}^{pl}, \theta, f^{\alpha})$ can include hardening as well as rate-dependent effects. If the shapes of the stress-strain curves are different at different strain rates, the test data are entered as tables of yield stress values versus equivalent plastic strain at different equivalent plastic strain rates: one table per strain rate. The yield stress at a given strain and strain rate is interpolated directly from these tables.

Alternatively, when it can be assumed that the shapes of the hardening curves at different strain rates are similar, the hardening and rate dependence are specified separately. In this case we assume that the rate dependence can be written in a separable form:

$$\bar{\sigma} = R\,\sigma^0,$$

where $\sigma^0(\bar{\varepsilon}^{pl}, \theta, f^{\alpha})$ is the static yield stress for the Drucker-Prager hardening model and $R(\dot{\bar{\varepsilon}}^{pl}, \theta, f^{\alpha})$ scales this value at nonzero strain rate ($R = 1.0$ at $\dot{\bar{\varepsilon}}^{pl} = 0.0$). The rate-dependent yield ratio $\boldsymbol{R}$ is defined either in a tabular form or using the standard power law form

$$R = 1 + \left(\frac{\dot{\bar{\varepsilon}}^{pl}}{D} \right)^{\frac{1}{n}},$$

where $D(\theta, f^{\alpha})$ and $n(\theta, f^{\alpha})$ are material parameters.

Creep models are most suitable for applications that exhibit time-dependent inelastic deformation at low deformation rates. Such inelastic deformation, which can coexist with rate-independent plastic deformation, is described later in this section. However, the existence of creep in an Abaqus material definition precludes the use of rate dependence as described above.

Strain rate decomposition

An additive strain rate decomposition is assumed:

$$d\boldsymbol{\varepsilon} = d\boldsymbol{\varepsilon}^{el} + d\boldsymbol{\varepsilon}^{pl} + d\boldsymbol{\varepsilon}^{cr}, \tag{4.4.2–2}$$

where $d\boldsymbol{\varepsilon}$ is the total strain rate, $d\boldsymbol{\varepsilon}^{el}$ is the elastic strain rate, $d\boldsymbol{\varepsilon}^{pl}$ is the inelastic (plastic) strain rate, and $d\boldsymbol{\varepsilon}^{cr}$ is the inelastic creep strain rate. The term $d\boldsymbol{\varepsilon}^{pl}$ is omitted if the stress point is inside the yield surface, and the term $d\boldsymbol{\varepsilon}^{cr}$ is omitted if creep has not been defined or is not active.

Elastic behavior

The elastic behavior can be modeled as linear or with the porous elasticity model including tensile strength described in "Porous elasticity," Section 4.4.1. If creep has been defined, the elastic behavior must be modeled as linear.

Linear Drucker-Prager model

In this model we define a deviatoric stress measure

$$t = \frac{q}{2} \left[1 + \frac{1}{K} - \left(1 - \frac{1}{K} \right) \left(\frac{r}{q} \right)^3 \right],$$

where $K(\theta, f_{\alpha})$ is a material parameter. To ensure convexity of the yield surface, $0.778 \leq K \leq 1.0$. This measure of deviatoric stress is used because it allows the matching of different stress values in tension and compression in the deviatoric plane, thereby providing flexibility in fitting experimental results when the material exhibits different yield values in triaxial tension and compression tests. This function is sketched in Figure 4.4.2–2. It only provides a coarse match to Mohr-Coulomb behavior (where the yield is indepe7ndent of the intermediate principal stress). Since $(r/q)^3 = 1$ in uniaxial tension, $t = q/K$ in this case; since $(r/q)^3 = -1$ in uniaxial compression, $t = q$ in that case. When $\boldsymbol{K}$ = 1, the dependence on the third deviatoric stress invariant is removed; the Mises circle is recovered in the deviatoric plane: $t = q$.

With this expression for the deviatoric stress measure, the yield surface is defined as

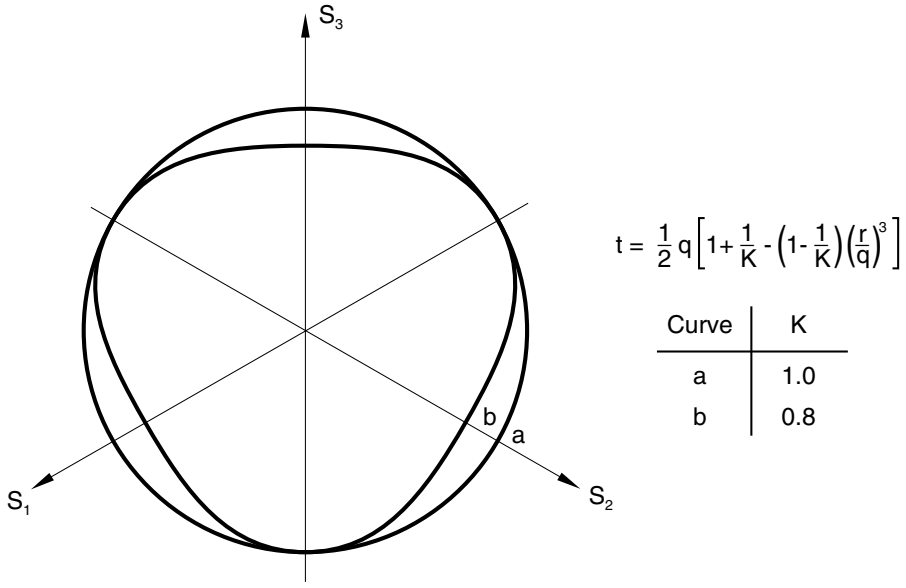


Figure 4.4.2–2 Typical yield surfaces for the linear model in the deviatoric plane.

$$F = t - p\tan\beta - d = 0, \tag{4.4.2–3}$$

where

$$\begin{aligned} d &= (1 - \frac{1}{3}\tan\beta)\sigma_c \quad \text{if hardening is defined by the uniaxial compression yield stress, } \sigma_c; \\ &= (\frac{1}{K} + \frac{1}{3}\tan\beta)\sigma_t \quad \text{if hardening is defined by the uniaxial tension yield stress, } \sigma_t, \\ &= d \quad \text{if hardening is defined by the shear (cohesion) yield stress, } d; \end{aligned}$$

and $\beta(\theta, f^\alpha)$ is the friction angle of the material in the meridional stress plane.

In the case of hardening defined in uniaxial compression, the linear yield criterion precludes friction angles $\beta > 71.5°$ ($\tan\beta > 3$). This is not seen as a limitation since it is unlikely this will be the case for real materials.

The hardening parameter $d(\bar{\sigma})$ measures the cohesion of the material and represents isotropic hardening, as illustrated in Figure 4.4.2–3. The formulation treats β as constant with respect to stress, although it is straightforward to extend the theory to provide for the functional dependence of β on quantities such as $\boldsymbol{p}$.

A method for converting Mohr-Coulomb data (ϕ, the angle of Coulomb friction, and $\boldsymbol{c}$, the cohesion) to appropriate values of β and $\boldsymbol{d}$ is described in the Abaqus Analysis User's Guide.

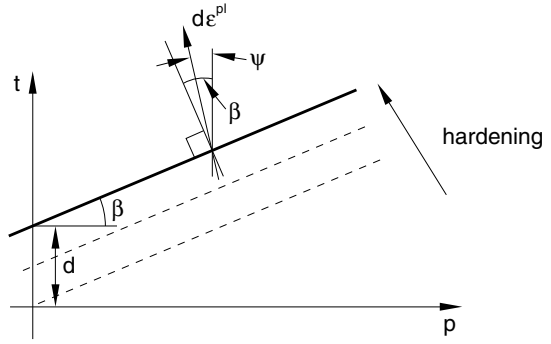


Figure 4.4.2–3 Schematic of hardening and flow for the linear model in the ***p–t*** plane.

Flow rule

Potential flow in the linear model is assumed, so that

$$d\boldsymbol{\varepsilon}^{pl} = \frac{d\bar{\varepsilon}^{pl}}{c}\frac{\partial G}{\partial \boldsymbol{\sigma}}, \tag{4.4.2–4}$$

where

$$\begin{aligned} c =& (1 - \frac{1}{3}\tan\psi) \quad \text{if hardening is defined in uniaxial compression,} \\ =& (\frac{1}{K} + \frac{1}{3}\tan\psi) \quad \text{if hardening is defined in uniaxial tension,} \\ =& \frac{1}{2}(1 + \frac{1}{K}) \quad \text{if hardening is defined in pure shear (cohesion),} \end{aligned}$$

and

$$\begin{aligned} d\bar{\varepsilon}^{pl} =& |d\epsilon_{11}^{pl}| \quad \text{in the uniaxial compression case,} \\ =& d\epsilon_{11}^{pl} \quad \text{in the uniaxial tension case,} \\ =& \frac{d\gamma^{pl}}{\sqrt{3}} \quad \text{in the pure shear case, where } \gamma^{pl} \text{ is the engineering shear plastic strain.} \end{aligned}$$

G is the flow potential, chosen in this model as

$$G = t - p\tan\psi, \tag{4.4.2–5}$$

where $\psi(\theta, f^{\alpha})$ is the dilation angle in the ***p–t*** plane. A geometrical interpretation of ψ is shown in the ***t–p*** diagram of Figure 4.4.2–3. In the case of hardening defined in uniaxial compression, this flow rule

definition precludes dilation angles $\psi > 71.5°$ ($\tan\psi > 3$). This is not seen as a limitation since it is unlikely this will be the case for real materials.

Comparison of Equation 4.4.2–3 and Equation 4.4.2–5 shows that the flow is associated in the deviatoric plane, because the yield surface and the flow potential both have the same functional dependence on $\boldsymbol{t}$. However, the dilation angle, ψ, and the material friction angle, β, may be different, so the model may not be associated in the $\boldsymbol{p}$–$\boldsymbol{t}$ plane. For $\psi = 0$ the material is nondilational; and if $\psi = \beta$, the model is fully associated—the model is then of the type first introduced by Drucker and Prager (1952). For $\psi = \beta$ and $K = 1$ the original Drucker-Prager model is recovered.

Hyperbolic and general exponent models

The hyperbolic and general exponent models, which are only available in Abaqus/Standard, are written in terms of the first two stress invariants only. The hyperbolic yield criterion is a continuous combination of the maximum tensile stress condition of Rankine (tensile cutoff) and the linear Drucker-Prager condition at high confining stress. It is written as

$$F = \sqrt{l_0{}^2 + q^2} - p\tan\beta - d' = 0, \tag{4.4.2–6}$$

where $l_0 = (d'|_0 - p_t|_0 \tan\beta)$, $p_t|_0$ is the initial hydrostatic tension strength of the material, $d'|_0$ is the initial value of d', and $\beta(\theta, f^\alpha)$ is the friction angle measured at high confining pressure, as shown in Figure 4.4.2–1(b). $d'(\bar{\sigma})$ is the hardening parameter, which is obtained from test data:

$$\begin{aligned} d' &= \sqrt{l_0^2 + {\sigma_c}^2} - \frac{\sigma_c}{3}\tan\beta \quad \text{if hardening is defined by the uniaxial compression yield stress, } \sigma_c; \\ &= \sqrt{l_0^2 + {\sigma_t}^2} + \frac{\sigma_t}{3}\tan\beta \quad \text{if hardening is defined by the uniaxial tension yield stress, } \sigma_t; \\ &= \sqrt{l_0^2 + d^2} \quad \text{if hardening is defined by the shear (cohesion) yield stress, } d. \end{aligned}$$

The isotropic hardening assumed in this model treats β as constant with respect to stress and is depicted in Figure 4.4.2–4. Calibration of this model is described in the Abaqus Analysis User's Guide.

The general exponent form provides the most general yield criterion available in this class of models. The yield function is written as

$$F = aq^b - p - p_t = 0, \tag{4.4.2–7}$$

where $a(\theta, f^\alpha)$ and $b(\theta, f^\alpha)$ are material parameters that are independent of plastic deformation and $p_t(\bar{\sigma})$ is the hardening parameter that represents the hydrostatic tension strength of the material, as shown in Figure 4.4.2–1(c). $p_t(\bar{\sigma})$ is related to test data as

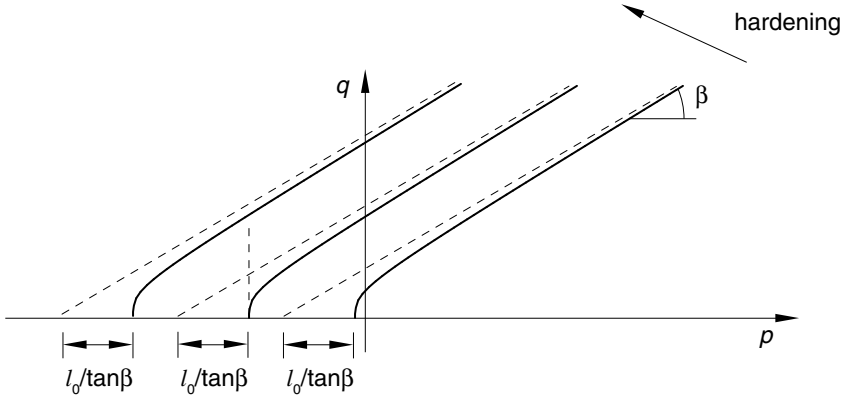


Figure 4.4.2–4 Schematic diagram of hardening for the hyperbolic model in the p–q plane.

$$p_t = a\sigma_c{}^b - \frac{\sigma_c}{3} \quad \text{if hardening is defined by the uniaxial compression yield stress, } \sigma_c;$$
$$= a\sigma_t{}^b + \frac{\sigma_t}{3} \quad \text{if hardening is defined by the uniaxial tension yield stress, } \sigma_t;$$
$$= ad^b \quad \text{if hardening is defined by the shear (cohesion) yield stress, } d.$$

The isotropic hardening assumed in this model treats a and b as constant with respect to stress and is depicted in Figure 4.4.2–5.

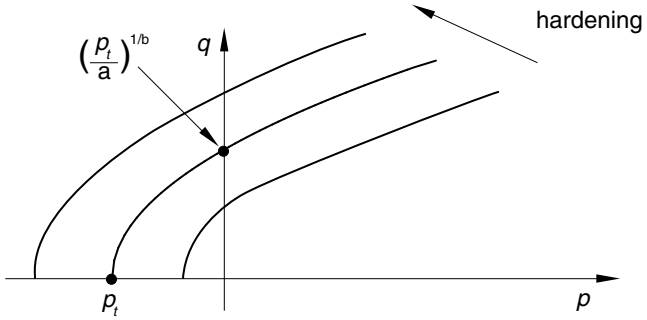


Figure 4.4.2–5 Schematic diagram of hardening for the general exponent model in the p–q plane.

The material parameters a, b, and p_t can be given directly; or, if triaxial test data at different levels of confining pressure are available, Abaqus will determine the material parameters from the triaxial test

data. A least squares fit, which minimizes the relative error in stress, is used to obtain the "best fit" values for a, b, and p_t.

Flow rule

Potential flow in the hyperbolic and general exponent models is assumed, so that

$$d\boldsymbol{\varepsilon}^{pl} = \frac{d\bar{\varepsilon}^{pl}}{f}\frac{\partial G}{\partial \boldsymbol{\sigma}}, \tag{4.4.2–8}$$

where f depends on how the hardening is defined (by uniaxial compression, uniaxial tension, or pure shear data) but can be written in general as

$$f = \frac{1}{\bar{\sigma}}\,\boldsymbol{\sigma} : \frac{\partial G}{\partial \boldsymbol{\sigma}},$$

and

$$\begin{aligned} d\bar{\varepsilon}^{pl} &= |d\epsilon_{11}^{pl}| \quad \text{in the uniaxial compression case,} \\ &= d\epsilon_{11}^{pl} \quad \text{in the uniaxial tension case,} \\ &= \frac{d\gamma^{pl}}{\sqrt{3}} \quad \text{in the pure shear case, where } \gamma^{pl} \text{ is the engineering shear plastic strain.} \end{aligned}$$

G is the flow potential, chosen in these models as a hyperbolic function:

$$G = \sqrt{(\epsilon\bar{\sigma}|_0 \tan\psi)^2 + q^2} - p\tan\psi, \tag{4.4.2–9}$$

where $\psi(\theta, f^\alpha)$ is the dilation angle measured in the p–q plane at high confining pressure; $\bar{\sigma}|_0 = \bar{\sigma}|_{\bar{\varepsilon}^{pl}=0,\dot{\bar{\varepsilon}}^{pl}=0}$ is the initial equivalent yield stress; and ϵ is a parameter, referred to as the eccentricity, that defines the rate at which the function approaches the asymptote (the flow potential tends to a straight line as the eccentricity tends to zero). This flow potential, which is continuous and smooth, ensures that the flow direction is defined uniquely. The function asymptotically approaches the linear Drucker-Prager flow potential at high confining pressure stress and intersects the hydrostatic pressure axis at 90°. A family of hyperbolic potentials in the meridional stress plane is shown in Figure 4.4.2–6. The flow potential is a von Mises circle in the deviatoric stress plane (the Π-plane).

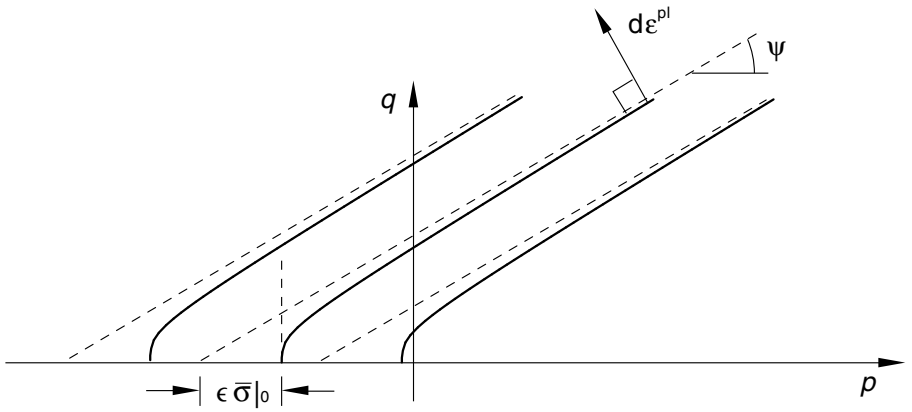


Figure 4.4.2–6 Family of hyperbolic flow potentials in the p–q plane.

In both models flow is associated in the deviatoric stress plane. In the general exponent model, flow is always nonassociated in the meridional p–q plane. In the hyperbolic model comparison of Equation 4.4.2–6 and Equation 4.4.2–9 shows that the flow is nonassociated in the p–q plane if the dilation angle, ψ, and the material friction angle, β, are different. The hyperbolic model provides associated flow in the p–q plane only when $\beta = \psi$ and $d'|_0/\tan\beta - p_t|_0 = \epsilon\bar{\sigma}|_0$.

Creep models

Classical "creep" behavior of materials that exhibit plastic behavior according to the extended Drucker-Prager models can be defined.

The creep behavior in such materials is intimately tied to the plasticity behavior (through the definition of the creep flow potential and test data), so it is necessary to define the Drucker-Prager plasticity and hardening behavior as well. The elastic part of the behavior must be linear.

The rate-independent part of the plastic behavior is limited to the linear Drucker-Prager model with a von Mises (circular) section in the deviatoric stress plane (K=1). The plastic potential is the hyperbolic flow potential described in conjunction with the hyperbolic and general exponent models (Equation 4.4.2–9).

Creep behavior

We adopt the notion of creep isosurfaces (or equivalent creep surfaces) of stress points that share the same creep "intensity," as measured by an equivalent creep stress. When the material plastifies, the equivalent creep surface should coincide with the yield surface; therefore, we define the equivalent creep surfaces by homogeneously scaling down the yield surface. In the p–q plane that translates into parallels to the yield surface, as depicted in Figure 4.4.2–7.

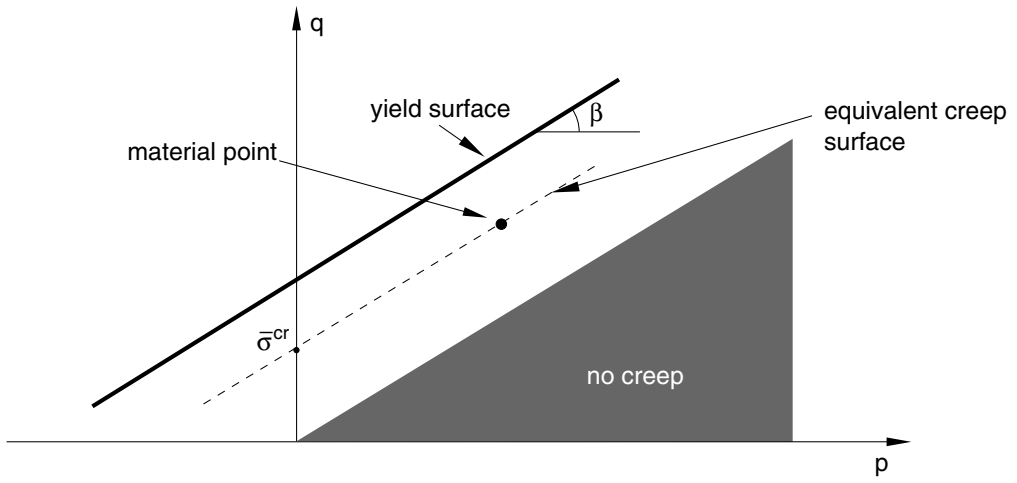


Figure 4.4.2–7 Equivalent creep stress defined as the shear stress.

Abaqus requires that creep properties be defined through the same type of test data used to define work hardening properties. The equivalent creep stress, $\bar{\sigma}^{cr}$, is determined as the intersection of the equivalent creep surface with the appropriate stress path. As a result,

$$\begin{aligned}
\bar{\sigma}^{cr} &= \frac{(q - p\tan\beta)}{(1 - \frac{1}{3}\tan\beta)} \quad \text{if creep is defined in terms of the uniaxial compression stress}, \sigma_c; \\
&= \frac{(q - p\tan\beta)}{(1 + \frac{1}{3}\tan\beta)} \quad \text{if creep is defined in terms of the uniaxial tension stress}, \sigma_t; \\
&= (q - p\tan\beta) \quad \text{if creep is defined in terms of the shear (cohesion) stress}, d;
\end{aligned}$$

where $\beta(\theta, f^{\alpha})$ is the material angle of friction.

Figure 4.4.2–7 shows how the equivalent creep stress is determined when the material properties are defined via a shear test: a parallel to the yield surface is drawn, such that it passes by the material point; the intersection of such a line with the test stress path ($p = 0$) produces $\bar{\sigma}^{cr}$.

This approach has the consequence that the creep strain rate is a function of both q and p and allows realistic material properties to be determined in cases in which, due to high hydrostatic pressures, q is very high. If one looks at the yield strength of this material to be a composite of cohesion strength and friction strength, this model corresponds to cohesion-determined creep. Thus, there is a cone in p–q space inside which there is no creep.

The built-in Abaqus creep laws or the uniaxial laws defined through user subroutine **CREEP** can be used. The integration of the creep strain rate is first attempted explicitly, as described in "Rate-dependent metal plasticity (creep)," Section 4.3.4. If the stability limit is exceeded, a geometrically nonlinear

analysis is being performed, or plasticity becomes active, the integration is done by the backward Euler method, as described in "Rate-dependent metal plasticity (creep)," Section 4.3.4.

Creep flow rule

The creep flow rule is derived from a creep potential, G^{cr}, in such a way that

$$d\boldsymbol{\varepsilon}^{cr} = \frac{d\bar{\varepsilon}^{cr}}{f^{cr}} \frac{\partial G^{cr}}{\partial \boldsymbol{\sigma}}, \tag{4.4.2–10}$$

where $d\bar{\varepsilon}^{cr}$ is the equivalent creep strain rate, which must be work conjugate to the equivalent creep stress:

$$\begin{aligned} d\bar{\varepsilon}^{cr} &= |d\varepsilon_{11}^{cr}| \quad \text{in the uniaxial compression case,} \\ &= d\varepsilon_{11}^{cr} \quad \text{in the uniaxial tension case,} \\ &= \frac{d\gamma^{cr}}{\sqrt{3}} \quad \text{in the pure shear case, where } \gamma^{cr} \text{ is the engineering shear creep strain.} \end{aligned}$$

Since $d\boldsymbol{\varepsilon}^{cr}$ is obviously work conjugate to $\boldsymbol{\sigma}$, f^{cr} is defined by

$$f^{cr} = \frac{1}{\bar{\sigma}^{cr}} \boldsymbol{\sigma} : \frac{\partial G^{cr}}{\partial \boldsymbol{\sigma}}.$$

The equivalent creep strain rate is then determined from the "uniaxial" creep law:

$$d\bar{\varepsilon}^{cr} = h(\bar{\sigma}^{cr}, \bar{\varepsilon}^{cr}, \theta, f^{\alpha}).$$

The creep strain rate is assumed to follow from the same hyperbolic potential as the plastic strain rate

$$G^{cr} = \sqrt{(\epsilon \bar{\sigma}|_0 \tan\psi)^2 + q^2} - p \tan\psi, \tag{4.4.2–11}$$

where $\psi(\theta, f^{\alpha})$ is the dilation angle measured in the $\boldsymbol{p}$–$\boldsymbol{q}$ plane at high confining pressure; $\bar{\sigma}|_0 = \bar{\sigma}|_{\bar{\varepsilon}^{pl}=0, \dot{\bar{\varepsilon}}^{pl}=0}$ is the initial yield stress; and ϵ is a parameter, referred to as the eccentricity, that defines the rate at which the function approaches the asymptote (the creep potential tends to a straight line as the eccentricity tends to zero). This creep potential, which is continuous and smooth, ensures that the creep flow direction is always uniquely defined. The function approaches the linear Drucker-Prager creep potential asymptotically at high confining pressure stress and intersects the hydrostatic pressure axis at 90°. A family of hyperbolic potentials in the meridional stress plane is shown in Figure 4.4.2–6. The creep potential is the von Mises circle in the deviatoric stress plane (the Π-plane).

Equation 4.4.2–10 and Equation 4.4.2–11 produce the complete flow rule

$$\Delta\boldsymbol{\varepsilon}^{cr} = \frac{\Delta\bar{\varepsilon}^{cr}}{f^{cr}} \left(\frac{q}{\sqrt{(\epsilon \bar{\sigma}|_0 \tan\psi)^2 + q^2}} \mathbf{n} + \frac{1}{3} \tan\psi \mathbf{I} \right), \tag{4.4.2–12}$$

where

$$\mathbf{n} = \frac{\partial q}{\partial \boldsymbol{\sigma}} = \frac{3}{2}\frac{\boldsymbol{S}}{q},$$

and

$$\begin{aligned} f^{cr} &= \frac{\bar{\sigma}^{cr}}{\sqrt{(\epsilon\bar{\sigma}_{|0}\tan\psi)^2 + (\bar{\sigma}^{cr})^2}} - \frac{1}{3}\tan\psi \quad \text{if creep is defined via the uniaxial compression data,} \\ &= \frac{\bar{\sigma}^{cr}}{\sqrt{(\epsilon\bar{\sigma}_{|0}\tan\psi)^2 + (\bar{\sigma}^{cr})^2}} + \frac{1}{3}\tan\psi \quad \text{if creep is defined via the uniaxial tension data,} \\ &= \frac{\bar{\sigma}^{cr}}{\sqrt{(\epsilon\bar{\sigma}_{|0}\tan\psi)^2 + (\bar{\sigma}^{cr})^2}} \quad \text{if creep is defined via the shear data.} \end{aligned}$$

The expressions for f^{cr} indicate that when creep properties are defined in terms of uniaxial compression data, f^{cr} will become negative if

$$\bar{\sigma}^{cr} < \epsilon\bar{\sigma}_{|0}\frac{\frac{1}{3}(\tan\psi)^2}{\sqrt{1-(\frac{1}{3}\tan\psi)^2}}.$$

Thus, below this stress level, which for typical materials will be very low, the stress vector and the normal to the creep potential are pointing in opposite directions:

$$\boldsymbol{\sigma} : \frac{\partial G^{cr}}{\partial \boldsymbol{\sigma}} < 0,$$

which is equivalent to

$$q - p\tan\psi < q - \frac{q^2}{\sqrt{(\epsilon\bar{\sigma}_{|0}\tan\psi)^2 + q^2}}.$$

Therefore, if $\psi = \beta$, there is a small zone just outside the "no creep" cone for which this is the case. Consequently, creep data obtained within this zone (such as data obtained in uniaxial compression) should show a creep strain rate in the opposite direction from the applied stress at very low stress levels, which will usually not be the case. To overcome this difficulty, Abaqus will modify the creep data entered such that $f^{cr} \geq 0.1$. Thus, one would not expect correspondence between calculated creep strains and measured creep properties in a region defined by

$$\bar{\sigma}^{cr} < \epsilon\bar{\sigma}_{|0}\frac{\alpha\tan\psi}{\sqrt{1-\alpha^2}},$$

where

$$\alpha = 0.1 + \frac{1}{3}\tan\psi \quad \text{if the model is defined in terms of uniaxial compression data,}$$
$$= 0.1 - \frac{1}{3}\tan\psi \quad \text{if the model is defined in terms of uniaxial tension data,}$$
$$= 0.1 \quad \text{if the model is defined in terms of shear data.}$$

The exact size of this region depends on the value of $\tan\psi$ and the type of test data entered. This modification is usually not significant since typical creep analyses have loads that are applied quickly, followed by long-term creep. Hence, the stress level for most of the analysis will usually be well beyond the modified zone.

An example of "slow" loading in which the approximation is visible is included in "Verification of creep integration," Section 3.2.6 of the Abaqus Benchmarks Guide. As is clear in the example, the effect of the approximation is small in spite of the fact that the load is ramped up over the step.

Although creep flow is associated in the deviatoric stress plane, the use of a creep potential different from the equivalent creep surface implies that creep flow is nonassociated.

Reference

- "Extended Drucker-Prager models," Section 23.3.1 of the Abaqus Analysis User's Guide

4.4.3 CRITICAL STATE MODELS

Products: Abaqus/Standard Abaqus/Explicit

The inelastic constitutive theory provided in Abaqus/Standard for modeling cohesionless materials is based on the critical state plasticity theory developed by Roscoe and his colleagues at Cambridge (Schofield et al., 1968, and Parry, 1972). The specific model implemented is an extension of the "modified Cam-clay" theory. The discussion is entirely in terms of effective stress: the soil may be saturated with a permeating fluid that carries a pressure stress and is assumed to flow according to Darcy's law. The continuum theory of this two phase material is described in "Continuity statement for the wetting liquid phase in a porous medium," Section 2.8.4.

The modified Cam-clay theory is a classical plasticity model. It uses a strain rate decomposition in which the rate of mechanical deformation of the soil is decomposed into an elastic and a plastic part; an elasticity theory; a yield surface; a flow rule; and a hardening rule. These various parts of the theory are defined in this section. The model is implemented numerically using backward Euler integration of the flow rule and hardening rule: this approach is used throughout Abaqus for plasticity models.

The basic ideas of the Cam-clay model are shown geometrically in Figure 4.4.3–1 to Figure 4.4.3–7. The main features of the model are the use of an elastic model (either linear elasticity or the porous elasticity model, which exhibits an increasing bulk elastic stiffness as the material undergoes compression) and for the inelastic part of the deformation a particular form of yield surface with associated flow and a hardening rule that allows the yield surface to grow or shrink.

A key feature of the model is the hardening/softening concept, which is developed around the introduction of a "critical state" surface: the locus of effective stress states where unrestricted, purely deviatoric, plastic flow of the soil skeleton occurs under constant effective stress. This critical state surface is assumed to be a cone in the space of principal effective stress (Figure 4.4.3–1), whose vertex is the origin (zero effective stress) and whose axis is the equivalent pressure stress, p. The section of the surface in the Π-plane (the plane in principal stress space orthogonal to the equivalent pressure stress axis) is circular in the original form of the critical state model: in Abaqus this has been extended to the more general shape shown in Figure 4.4.3–5. In the section of effective stress space defined by the equivalent pressure stress—p—and a measure of equivalent deviatoric stress—t (the definition of t is given later in this section)—the critical state surface appears as a straight line, passing through the origin, with slope M (see Figure 4.4.3–2 and Figure 4.4.3–3). The modified Cam-clay yield surface has the same shape in the Π-plane as the critical state surface, but in the p–t plane it is assumed to be made up of two elliptic arcs: one arc passes through the origin with its tangent at right angles to the pressure stress axis and intersects the critical state line where its tangent is parallel to the pressure stress axis, while the other arc is a smooth continuation of the first arc through the critical state line and intersects the pressure stress axis at some nonzero value of pressure stress, again with its tangent at right angles to that axis (see Figure 4.4.3–4). Plastic flow is assumed to occur normal to this surface.

The hardening/softening assumption controls the size of the yield surface in effective stress space. The hardening/softening is assumed to depend only on the volumetric plastic strain component and is such that, when the volumetric plastic strain is compressive (that is, when the soil skeleton is compacted), the yield surface grows in size, while inelastic increase in the volume of the soil skeleton causes the yield surface

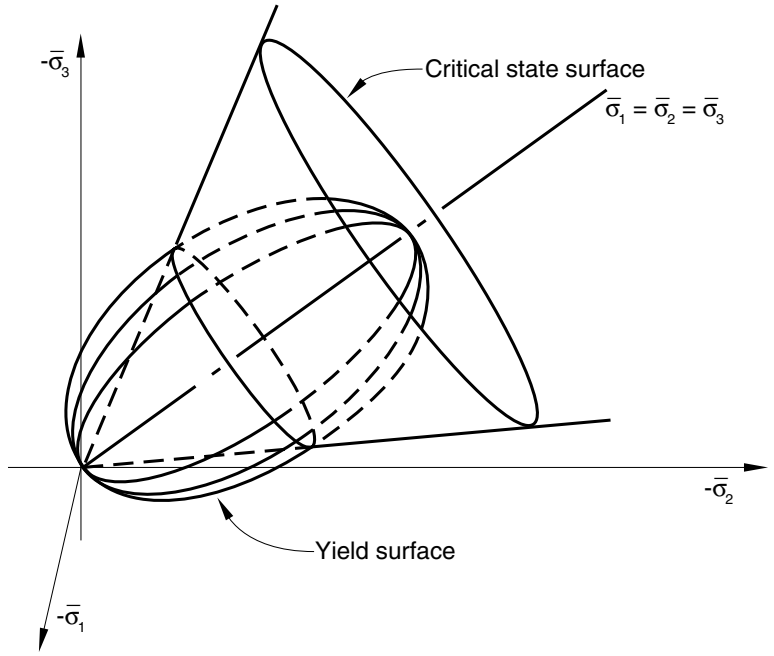


Figure 4.4.3–1 Cam-clay yield and critical state surfaces in principal stress space.

to shrink. The choice of elliptical arcs for the yield surface in the (p, t) plane, together with the associated flow assumption, thus causes softening of the material for yielding states where $t > Mp$ (to the left of the critical state line in Figure 4.4.3–2, the "dry" side of critical state) and hardening of the material for yielding states where $t < Mp$ (to the right of the critical state line in Figure 4.4.3–3, the "wet" side of critical state). The resulting stress-strain behavior under states of constant effective pressure stress but increasing shear (deviatoric) strain is then as shown in Figure 4.4.3–2 and Figure 4.4.3–3: following initial yield (which is governed by the initially assumed yield surface size; that is, by the extent of initial overconsolidation) strain softening or strain hardening occurs until the stress state lies on the critical state surface when unrestricted deviatoric plastic flow (perfect plasticity) occurs. The terms "wet" and "dry" come from the idea of working a specimen of soil by hand. On the "wet" side of critical state the soil skeleton is too loosely compacted to support pressure stress—such stress, if applied (such as by squeezing the soil by hand) passes immediately into the pore water and thus causes this water to bleed out of the specimen and wet the hands. The opposite effect occurs when the soil is on the "dry" side of critical state.

The preceding discussion describes the concepts of the theory. These are now formalized, as they are implemented in Abaqus/Standard.

The strain rate decomposition

The volume change is decomposed as

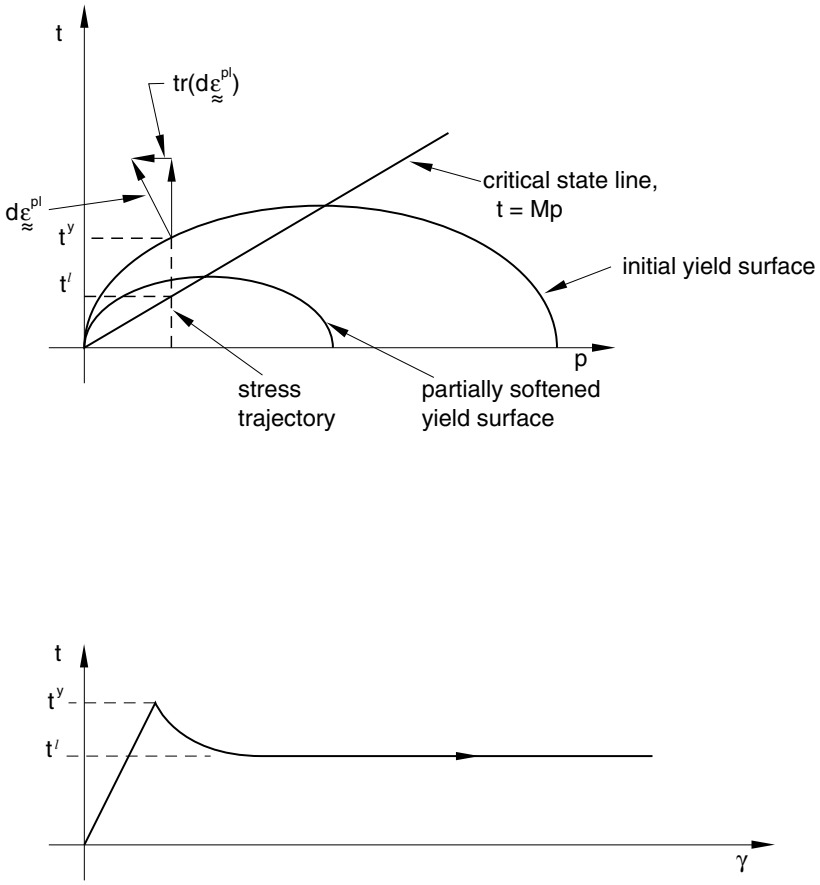


Figure 4.4.3–2 Shear test response on the "dry" side of critical state ($t > Mp$).

$$J = J^g \cdot J^{el} \cdot J^{pl}, \tag{4.4.3–1}$$

where $\boldsymbol{J}$ is the ratio of current volume to original volume, J^g is the ratio of current to original volume of the soil grain particles, J^{el} is the elastic (recoverable) part of the ratio of current to original volume of the soil volume, and J^{pl} is the plastic (nonrecoverable) part of the ratio of current to original volume of the soil volume.

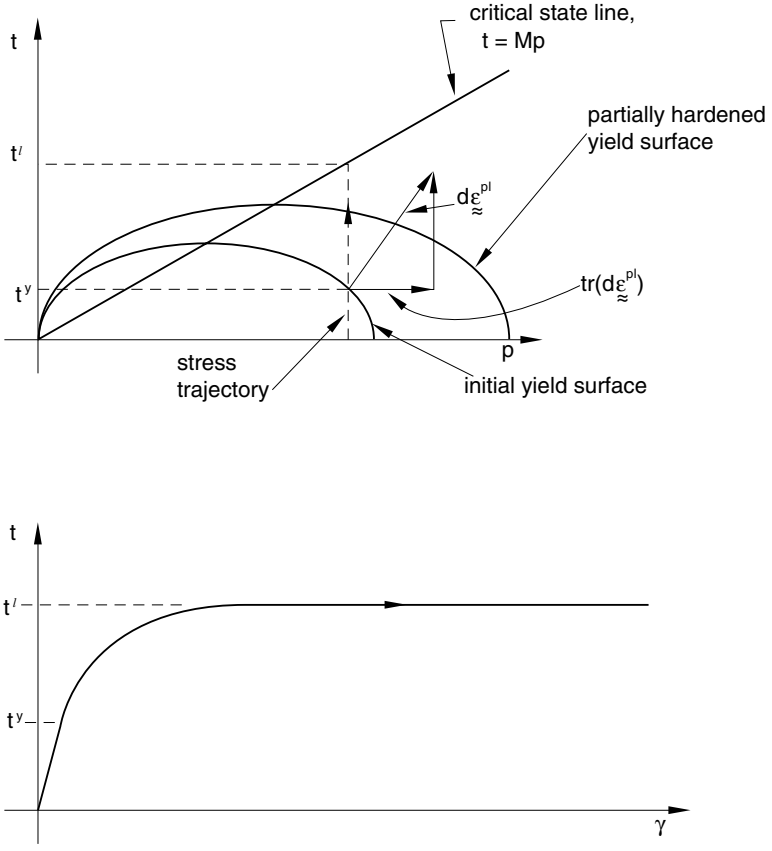


Figure 4.4.3–3 Shear test response on the "wet" side of critical state ($t < Mp$).

Volumetric strains are defined as

$$\begin{aligned}\varepsilon_{\text{vol}} &= \ln J,\\ \varepsilon^{el}_{\text{vol}} &= \ln J^{el},\\ \varepsilon^{pl}_{\text{vol}} &= \ln J^{pl}.\end{aligned}$$

These definitions and Equation 4.4.3–1 result in the usual additive strain rate decomposition for volumetric strain rates:

$$d\varepsilon_{\text{vol}} = d\varepsilon^{g}_{\text{vol}} + d\varepsilon^{el}_{\text{vol}} + d\varepsilon^{pl}_{\text{vol}}. \tag{4.4.3–2}$$

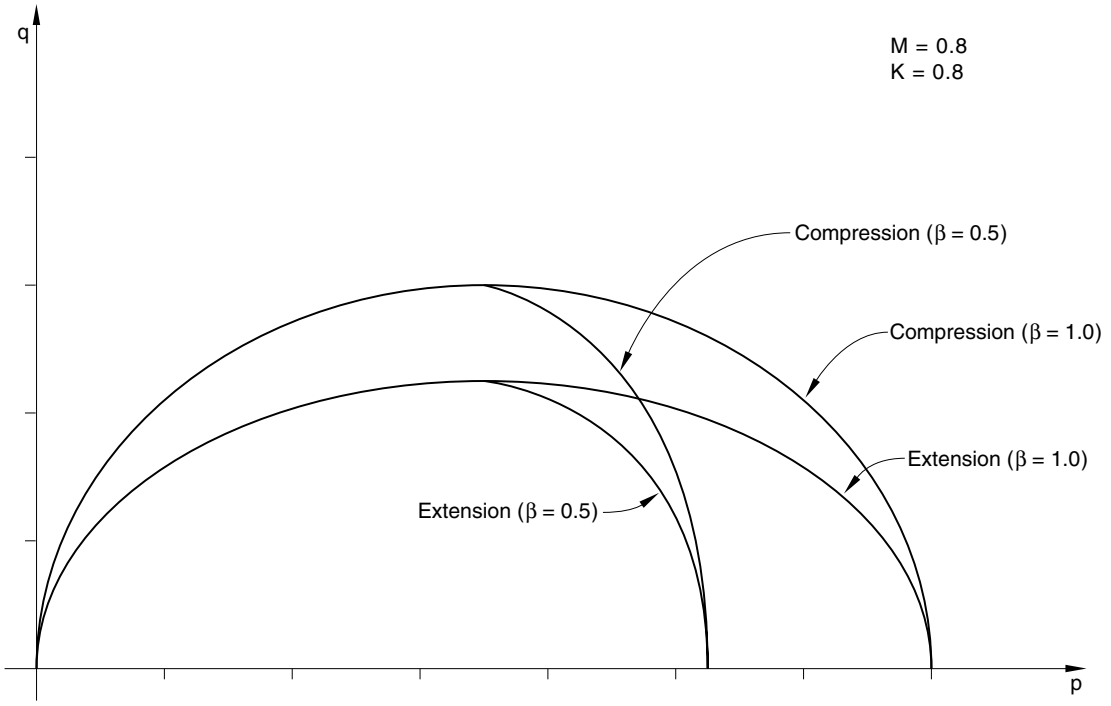


Figure 4.4.3–4 Cam-clay yield surface in the p–q plane.

The model also assumes the deviatoric strain rates decompose in an additive manner, so that the total strain rates decompose as

$$d\boldsymbol{\varepsilon} = d\varepsilon^{g}_{\mathrm{vol}}\,\mathbf{I} + d\boldsymbol{\varepsilon}^{el} + d\boldsymbol{\varepsilon}^{pl},$$

where $\mathbf{I}$ is a unit matrix.

Elastic behavior

The elastic behavior can be modeled as linear or by using the porous elasticity model, typically with a zero tensile strength, as described in "Porous elasticity," Section 4.4.1.

Plastic behavior

The modified Cam-clay yield function is defined in terms of the equivalent effective pressure stress, p, and the Mises equivalent stress and third stress invariant, defined as

$$p = -\frac{1}{3}\text{trace}\,\boldsymbol{\sigma} = -\frac{1}{3}\boldsymbol{\sigma} : \mathbf{I}$$
$$q = \sqrt{\frac{3}{2}\mathbf{S} : \mathbf{S}}$$
$$r^3 = \frac{9}{2}\mathbf{S} : \mathbf{S} \cdot \mathbf{S}.$$

The surface is

$$f(p, q, r) = \frac{1}{\beta^2}\left(\frac{p}{a} - 1\right)^2 + \left(\frac{t}{Ma}\right)^2 - 1 = 0. \tag{4.4.3–3}$$

In this equation $\beta = \beta(\theta, f^\alpha)$ is a user-specified constant that can be a function of temperature θ and other predefined field variables $f^\alpha, \alpha = 1, 2....$ This constant is used to modify the shape of the yield surface on the "wet" side of critical state, so the elliptic arc on the "wet" side of critical state has a different curvature from the elliptic arc used on the "dry" side: $\beta = 1$ on the "dry" side of critical state, while $\beta < 1$ in most cases on the "wet" side, as shown in Figure 4.4.3–4. $a(\theta, f^\alpha)$ defines the hardening of the plasticity model, and is the point on the ***p***-axis at which the elliptic arcs of the yield surface intersect the critical state line, as indicated in Figure 4.4.3–4. $M(\theta, f^\alpha)$ is the slope of the critical state line in the ***p***–***t*** plane (the ratio of ***t*** to ***p*** at critical state); and $t = q/g$, where ***g*** is used to shape the yield surface in the Π plane, and is defined as

$$g = \frac{2K}{1 + K + (1 - K)\,(r/q)^3},$$

where $K(\theta, f^\alpha)$ is a user-defined constant. If $K = 1$, the yield surface does not depend on the third stress invariant, and the Π-plane section of the yield surface is a circle: this choice gives the original form of the Cam-clay model. The effect of different values of ***K*** on the shape of the yield surface in the Π-plane is shown in Figure 4.4.3–5. To ensure convexity of the yield surface, $0.778 \leq K \leq 1.0$.

Associated flow is used with the modified Cam-clay plasticity model. The size of the yield surface is defined by ***a***: the evolution of this variable, therefore, characterizes the hardening or softening of the material. It is observed experimentally that, during plastic deformation,

$$de = -\lambda\, d(\ln p),$$

where λ is a constant. Integrating this equation, and using Equation 4.4.3–1, Equation 4.4.1–2, and Equation 4.4.1–4, we obtain

$$a = a_0 \exp\left[(1 + e_0)\frac{1 - J^{pl}}{\lambda - \kappa J^{pl}}\right], \tag{4.4.3–4}$$

where a_0 defines the position of ***a*** at the beginning of the analysis—the initial overconsolidation of the material. The value of a_0 can be specified directly by the user or can be computed as

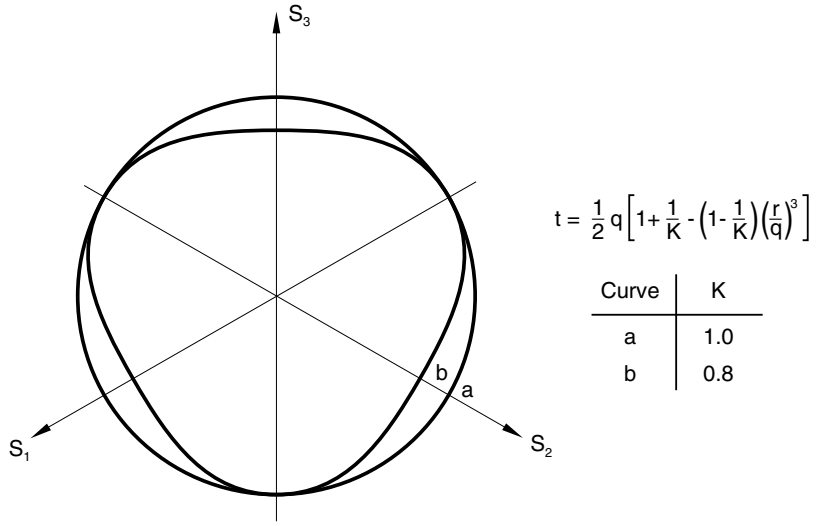


Figure 4.4.3–5 Cam-clay surfaces in the deviatoric plane.

$$a_0 = \frac{1}{2}\exp\left(\frac{e_1 - e_0 - \kappa \ln p_0}{\lambda - \kappa}\right),$$

where p_0 is the initial value of the equivalent pressure stress, and e_1 is the intercept of the virgin consolidation line with the void ratio axis in a plot of void ratio versus equivalent pressure stress, shown in Figure 4.4.3–6.

The evolution of the yield surface can alternatively be defined as a piecewise linear function relating the yield stress in hydrostatic compression, p_c, and the corresponding volumetric plastic strain $\varepsilon_{\text{vol}}^{pl}$ (Figure 4.4.3–7):

$$p_c = p_c(\varepsilon_{\text{vol}}^{pl}).$$

The evolution parameter, a, is then given by

$$a = \frac{p_c}{(1+\beta)}.$$

Note that the volumetric plastic strain axis has an arbitrary origin: $\varepsilon_{\text{vol}}^{pl}|_0$ is the position on this axis corresponding to the initial state of the material, thus defining the initial hydrostatic pressure, $p_c|_0$, and, hence, the initial yield surface size, a_0.

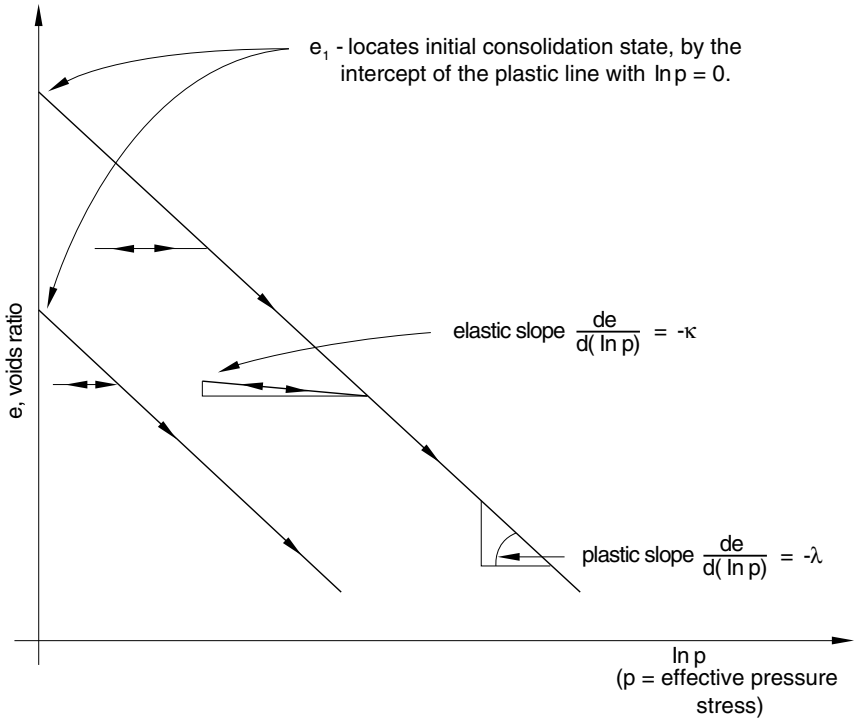


Figure 4.4.3–6 Assumed soil response in pure compression (exponential hardening/softening case).

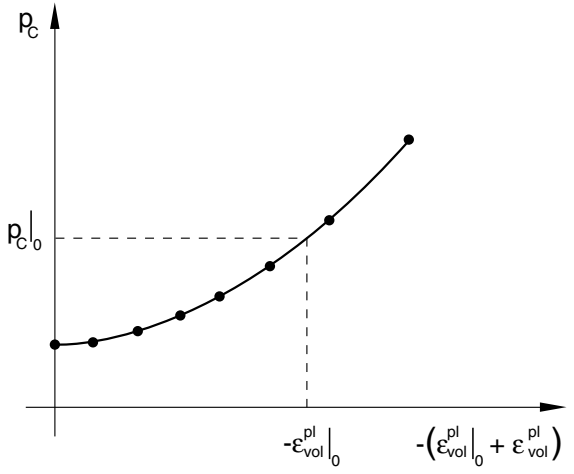


Figure 4.4.3–7 Piecewise linear hardening/softening curve.

Abaqus checks that the initial effective stress state lies inside or on the initial yield surface. At any material point where the yield function is violated, a_0 is adjusted so that Equation 4.4.3–3 is satisfied exactly (and, hence, the initial stress state lies on the yield surface).

Reference

- "Critical state (clay) plasticity model," Section 23.3.4 of the Abaqus Analysis User's Guide

4.4.4 DRUCKER-PRAGER/CAP MODEL FOR GEOLOGICAL MATERIALS

Products: Abaqus/Standard Abaqus/Explicit

The modified Drucker-Prager/Cap plasticity model in Abaqus is intended for geological materials that exhibit pressure-dependent yield. The yield surface includes two main segments: a shear failure surface, providing dominantly shearing flow, and a "cap," which intersects the equivalent pressure stress axis (Figure 4.4.4–1).

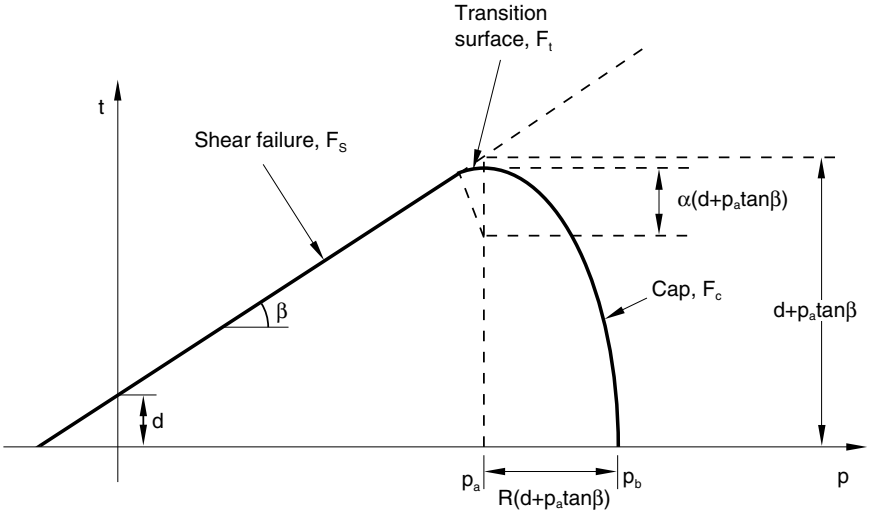


Figure 4.4.4–1 Modified Drucker-Prager/Cap model: yield surfaces in the p–t plane.

There is a transition region between these segments, introduced to provide a smooth surface. The cap serves two main purposes: it bounds the yield surface in hydrostatic compression, thus providing an inelastic hardening mechanism to represent plastic compaction, and it helps to control volume dilatancy when the material yields in shear by providing softening as a function of the inelastic volume increase created as the material yields on the Drucker-Prager shear failure and transition yield surfaces.

The model uses associated flow in the cap region and nonassociated flow in the shear failure and transition regions. The model has been extended to include creep, with certain limitations that are outlined in this section. The creep behavior is envisaged as arising out of two possible mechanisms: one dominated by shear behavior and the other dominated by hydrostatic compression.

Strain rate decomposition

A linear strain rate decomposition is assumed, so

$$d\boldsymbol{\varepsilon} = d\boldsymbol{\varepsilon}^{el} + d\boldsymbol{\varepsilon}^{pl} + d\boldsymbol{\varepsilon}^{cr},$$

where $d\boldsymbol{\varepsilon}$ is the total strain rate, $d\boldsymbol{\varepsilon}^{el}$ is the elastic strain rate, $d\boldsymbol{\varepsilon}^{pl}$ is the inelastic (plastic) time-independent strain rate, and $d\boldsymbol{\varepsilon}^{cr}$ is the inelastic (creep) time-dependent strain rate.

Elastic behavior

The elastic behavior can be modeled as linear elastic or by using the porous elasticity model including tensile strength, described in "Porous elasticity," Section 4.4.1. If creep has been defined, the elastic behavior must be modeled as linear.

Plastic behavior

The yield/failure surfaces used with this model are written in terms of the three stress invariants: the equivalent pressure stress,

$$p = -\frac{1}{3}\text{trace}(\boldsymbol{\sigma});$$

the Mises equivalent stress,

$$q = \sqrt{\frac{3}{2}(\mathbf{S} : \mathbf{S})};$$

and the third invariant of deviatoric stress,

$$r = (\frac{9}{2}\mathbf{S} \cdot \mathbf{S} : \mathbf{S})^{\frac{1}{3}},$$

where $\mathbf{S}$ is the stress deviator, defined as

$$\mathbf{S} = \boldsymbol{\sigma} + p\mathbf{I}.$$

We also define the deviatoric stress measure

$$t = \frac{q}{2}\left[1 + \frac{1}{K} - \left(1 - \frac{1}{K}\right)\left(\frac{r}{q}\right)^3\right],$$

where $K(\theta, f^\alpha)$ is a material parameter that may depend on temperature, θ, and other predefined fields $f^\alpha, \alpha = 1, 2, \ldots$. This measure of deviatoric stress is used because it allows matching of different stress values in tension and compression in the deviatoric plane, thereby providing flexibility in fitting experimental results and a smooth approximation to the Mohr-Coulomb surface. Since $r/q = 1$ in uniaxial tension, $t = q/K$ in this case; since $r/q = -1$ in uniaxial compression, $t = q$ in that case. When $K = 1$, the dependence on the third deviatoric stress invariant is removed; and the Mises circle is recovered in the deviatoric plane: $t = q$. Figure 4.4.4–2 shows the dependence of $\boldsymbol{t}$ on $\boldsymbol{K}$. To ensure convexity of the yield surface, $0.778 \leq K \leq 1.0$.

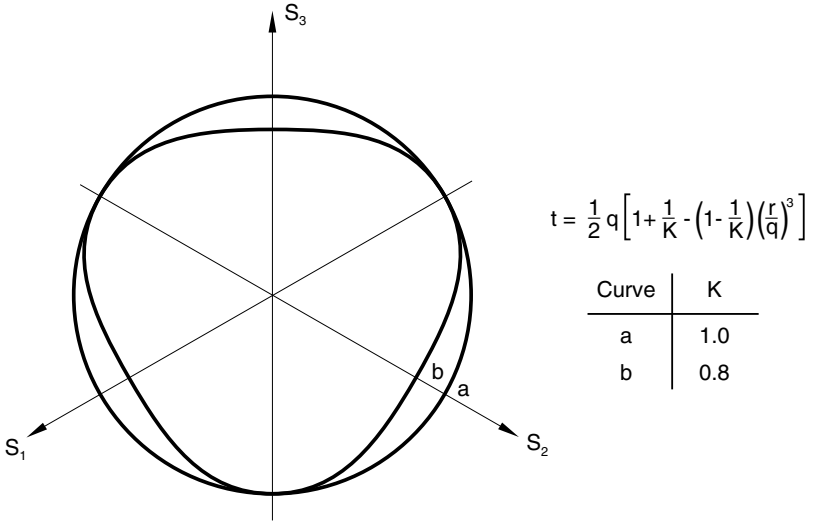


Figure 4.4.4–2 Typical yield/flow surfaces in the deviatoric plane.

With this expression for the deviatoric stress measure, the Drucker-Prager failure surface is written as

$$F_s = t - p\tan\beta - d = 0,$$

where $\beta(\theta, f^\alpha)$ is the material's angle of friction and $d(\theta, f^\alpha)$ is its cohesion (see Figure 4.4.4–1).

The cap yield surface has an elliptical shape with constant eccentricity in the meridional (***p–t***) plane (Figure 4.4.4–1) and also includes dependence on the third stress invariant in the deviatoric plane (Figure 4.4.4–2). The cap surface hardens or softens as a function of the volumetric plastic strain: volumetric plastic compaction (when yielding on the cap) causes hardening, while volumetric plastic dilation (when yielding on the shear failure surface) causes softening. The cap yield surface is written as

$$F_c = \sqrt{(p - p_a)^2 + \left[\frac{Rt}{(1 + \alpha - \alpha/\cos\beta)}\right]^2} - R(d + p_a\tan\beta) = 0,$$

where $R(\theta, f^\alpha)$ is a material parameter that controls the shape of the cap, $\alpha(\theta, f^\alpha)$ is a small number that is defined below, and p_a is an evolution parameter that represents the volumetric plastic strain driven hardening/softening. The hardening/softening law is a user-defined piecewise linear function relating the hydrostatic compression yield stress, p_b, and the corresponding volumetric inelastic (plastic and/or creep) strain, $p_b = p_b(\varepsilon_{\text{vol}}^{in}, \theta, f^\alpha)$ (Figure 4.4.4–3), where $\varepsilon_{\text{vol}}^{in} = \varepsilon_{\text{vol}}^{pl} + \varepsilon_{\text{vol}}^{cr}$.

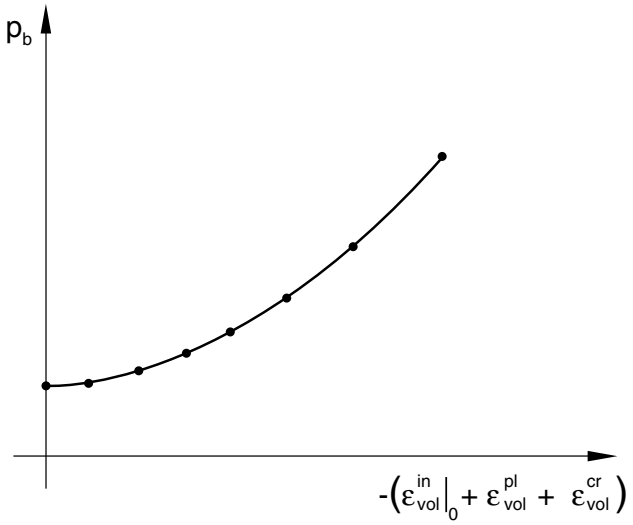


Figure 4.4.4–3 Typical Cap hardening.

The evolution parameter, p_a, is defined as

$$p_a = \frac{p_b - Rd}{(1 + R\tan\beta)}.$$

The parameter α is a small number (typically 0.01 to 0.05) used to define a smooth transition surface between the shear failure surface and the cap:

$$F_t = \sqrt{(p - p_a)^2 + \left[t - (1 - \frac{\alpha}{\cos\beta})(d + p_a\tan\beta)\right]^2} - \alpha(d + p_a\tan\beta) = 0.$$

Flow rule

Plastic flow is defined by a flow potential that is associated on the cap and nonassociated on the failure yield surface and transition yield surfaces. The nonassociated nature of these surfaces stems from the shape of the flow potential in the meridional plane. The flow potential surface in the meridional plane is shown in Figure 4.4.4–4.

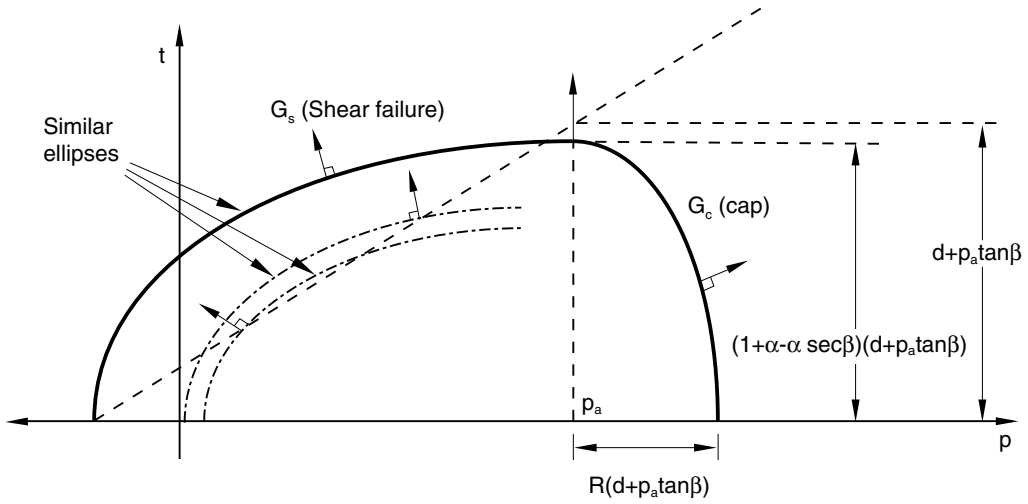


Figure 4.4.4–4 Modified Drucker-Prager/Cap model: flow potential in the p–t plane.

It is made up of an elliptical portion in the cap region that is identical to the cap yield surface:

$$G_c = \sqrt{(p - p_a)^2 + \left[\frac{Rt}{1 + \alpha - \alpha/\cos\beta}\right]^2}$$

and another elliptical portion in the failure and transition regions that provides the nonassociated flow component in the model:

$$G_s = \sqrt{[(p_a - p)\tan\beta]^2 + \left[\frac{t}{1 + \alpha - \alpha/\cos\beta}\right]^2}.$$

The two elliptical portions, G_c and G_s, form a continuous and smooth potential surface.

Nonassociated flow implies that the material stiffness matrix is not symmetric, so the unsymmetric solver should be invoked by the user. However, if the region of the model in which nonassociated inelastic deformation is occurring is confined, it is possible that a symmetric approximation to the material stiffness matrix will give an acceptable convergence rate: in such cases the unsymmetric solver may not be needed.

Creep model

Classical "creep" behavior of materials that also exhibit plastic behavior according to the modified Drucker-Prager/Cap model can be defined.

The creep behavior in such materials is intimately tied to the plasticity behavior (through the definition of creep flow potentials and test data), so it is necessary to define the modified

Drucker-Prager/Cap plasticity and hardening behavior as well. The elastic part of the behavior must be linear.

The rate-independent part of the plastic behavior is limited by the following restrictions:

$\alpha = 0$—that is, no transition zone is allowed;

K=1—that is, no third stress invariant effects are taken into account.

In such a case, the deviatoric stress measure t is equal to the Mises equivalent stress, q, and the yield surface has a von Mises (circular) section in the deviatoric stress plane.

Creep behavior

The built-in Abaqus creep laws or uniaxial laws defined through user subroutine **CREEP** can be used. The integration of the creep strain rate is first attempted explicitly, as described in "Rate-dependent metal plasticity (creep)," Section 4.3.4. The integration is done by the backward Euler method (as described in "Rate-dependent metal plasticity (creep)," Section 4.3.4) if the stability limit is exceeded, a geometrically nonlinear analysis is being performed, or plasticity becomes active.

In this model we assume the existence of two separate and independent creep mechanisms. One is a cohesion mechanism, which operates similarly to the Drucker-Prager creep model described in "Models for granular or polymer behavior," Section 4.4.2. The other is a consolidation mechanism, which operates similarly to the cap zone plasticity. We then have

$$d\boldsymbol{\varepsilon}^{cr} = d\boldsymbol{\varepsilon}_s^{cr} + d\boldsymbol{\varepsilon}_c^{cr},$$

where $d\boldsymbol{\varepsilon}_s^{cr}$ is the creep strain rate due to the cohesion mechanism and $d\boldsymbol{\varepsilon}_c^{cr}$ is the creep strain rate due to the consolidation mechanism.

As described above, the cap surface hardens or softens as a function of the volumetric plastic strain and volumetric creep strain: volumetric inelastic compaction (when yielding on the cap or creeping through the consolidation mechanism) causes hardening, while volumetric plastic dilation (when yielding on the shear failure surface or creeping through the cohesion mechanism) causes softening. The separation between the two yield surfaces and the dominant regions for the two creep mechanisms are defined by the evolution parameter, p_a, which relates to the user-defined hydrostatic compression yield stress, $p_b = p_b(\varepsilon_{\text{vol}}^{pl} + \varepsilon_{\text{vol}}^{cr}, \theta, f^{\alpha})$ (Figure 4.4.4–3).

The cohesion mechanism is active for all stress states that have a positive equivalent creep stress as explained below. The consolidation mechanism is active for all stress states in which the pressure is larger than p_a. Figure 4.4.4–5 illustrates the active regions in this formulation.

We adopt the notion of the existence of creep isosurfaces (or equivalent creep surfaces) of stress points that share the same creep "intensity," as measured by an equivalent creep stress. Consider the cohesion creep mechanism first. When the material plastifies, the equivalent creep surface should coincide with the yield surface; therefore, we define the equivalent creep surfaces by homogeneously scaling down the yield surface. In the p–q plane that translates into parallels to the yield surface, as depicted in Figure 4.4.4–6. Abaqus requires that cohesion creep properties be measured in a uniaxial compression test.

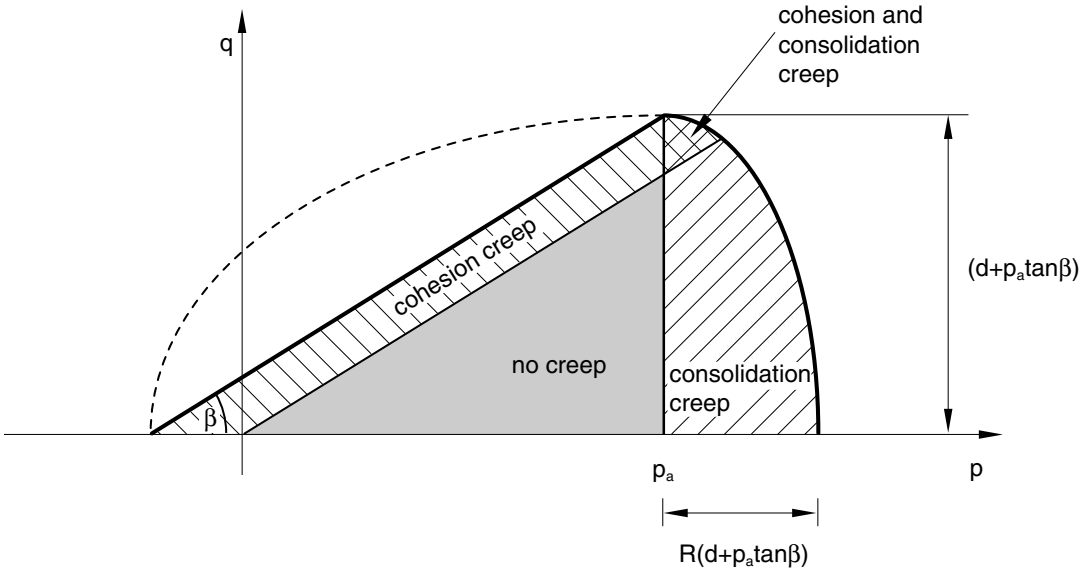


Figure 4.4.4–5 Regions of activity of cohesion and consolidation creep mechanisms.

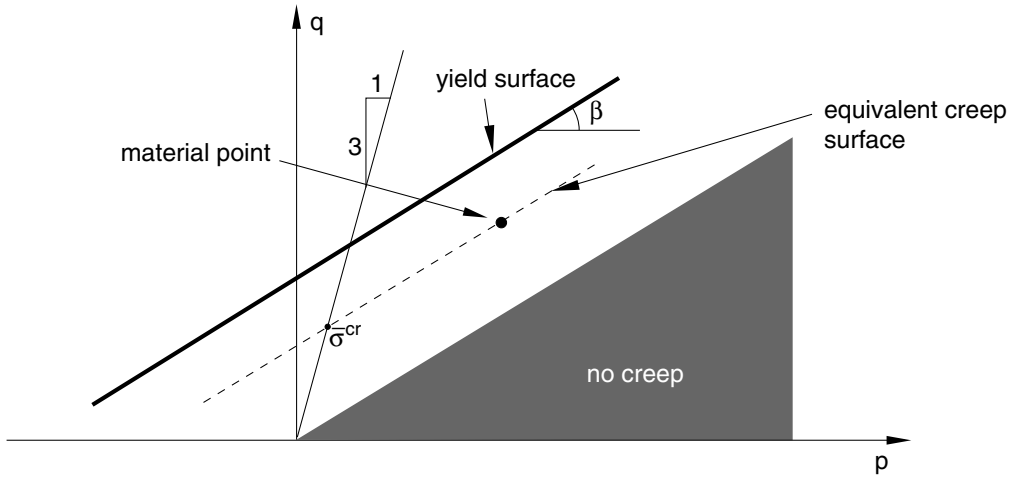


Figure 4.4.4–6 Equivalent creep stress for cohesion creep.

The equivalent creep stress, $\bar{\sigma}^{cr}$, is determined as the intersection of the equivalent creep surface with the uniaxial compression curve. As a result,

$$\bar{\sigma}^{cr} = \frac{(q - p\tan\beta)}{(1 - \frac{1}{3}\tan\beta)}, \tag{4.4.4–1}$$

where $\beta(\theta, f^{\alpha})$ is the material angle of friction. Figure 4.4.4–6 shows how the equivalent creep stress was determined. In uniaxial compression $p = \frac{1}{3}\bar{\sigma}^{cr}$; therefore, the uniaxial compression test line has a slope of 1/3. This approach has several consequences. One is that the cohesion creep strain rate is a function of both q and p. This allows the determination of realistic material properties in cases in which, due to high hydrostatic pressures, q is very high. If one looks at the yield strength of the material in this region to be a composite of cohesion strength and friction strength, this model corresponds to cohesion-determined creep. As a result, there is a cone in p–q space inside which there is no cohesion creep.

Next consider the consolidation creep mechanism. In this case we wish to make creep dependent on the hydrostatic pressure above a threshold value of p_a, with a smooth transition to the areas in which the mechanism is not active ($p < p_a$). Therefore, we define equivalent creep surfaces as constant pressure surfaces. In the p–q plane that translates into vertical lines. Abaqus requires that consolidation creep properties be measured in a hydrostatic compression test. The effective creep pressure, $\bar{p}^{cr}$, is then the point on the p-axis with a relative pressure

$$\bar{p}^{cr} = p - p_a. \tag{4.4.4–2}$$

This value is used in the uniaxial creep law. The equivalent volumetric creep strain rate produced by this type of law is defined as positive for a positive equivalent pressure. The internal tensor calculations in Abaqus will account for the fact that a positive pressure will produce negative (that is, compressive) volumetric creep components.

Creep flow rule

The creep flow rules are derived from creep potentials, G^{cr}, in such a way that

$$d\boldsymbol{\varepsilon}^{cr} = \frac{d\bar{\varepsilon}^{cr}}{f^{cr}}\frac{\partial G^{cr}}{\partial \boldsymbol{\sigma}}, \tag{4.4.4–3}$$

where $d\bar{\varepsilon}^{cr}$ is the equivalent creep strain rate, which must be work conjugate to the equivalent creep stress:

$$\begin{aligned} d\bar{\varepsilon}^{cr} &= |d\varepsilon_{11}^{cr}| \quad \text{in the uniaxial compression case,} \\ &= |d\varepsilon_{vol}^{cr}| \quad \text{in the volumetric compression case.} \end{aligned}$$

Since $d\boldsymbol{\varepsilon}^{cr}$ is obviously work conjugate to $\boldsymbol{\sigma}$, f^{cr} is a proportionality factor defined by

$$f^{cr} = \frac{1}{\bar{\sigma}^{cr}}\boldsymbol{\sigma} : \frac{\partial G^{cr}}{\partial \boldsymbol{\sigma}}, \tag{4.4.4–4}$$

with

$$\bar{\sigma}^{cr} = |\sigma_{11}| \quad \text{in the uniaxial compression case,}$$
$$= p \quad \text{in the volumetric compression case.}$$

Cohesion creep

For the cohesion mechanism the creep potential is assumed to follow the same potential as the creep strain rate in the Drucker-Prager creep model ("Models for granular or polymer behavior," Section 4.4.2); that is, a hyperbolic function. This creep flow potential, which is continuous and smooth, ensures that the flow direction is always uniquely defined. The function approaches a parallel to the shear-failure yield surface asymptotically at high confining pressure stress and intersects the hydrostatic pressure axis at a right angle. A family of hyperbolic potentials in the meridional stress plane is shown in Figure 4.4.4–7:

$$G_s^{cr} = \sqrt{(0.1\frac{d}{(1-\frac{1}{3}\tan\beta)}\tan\beta)^2 + q^2} - p\tan\beta, \tag{4.4.4–5}$$

where $\boldsymbol{d}$ is the material cohesion.

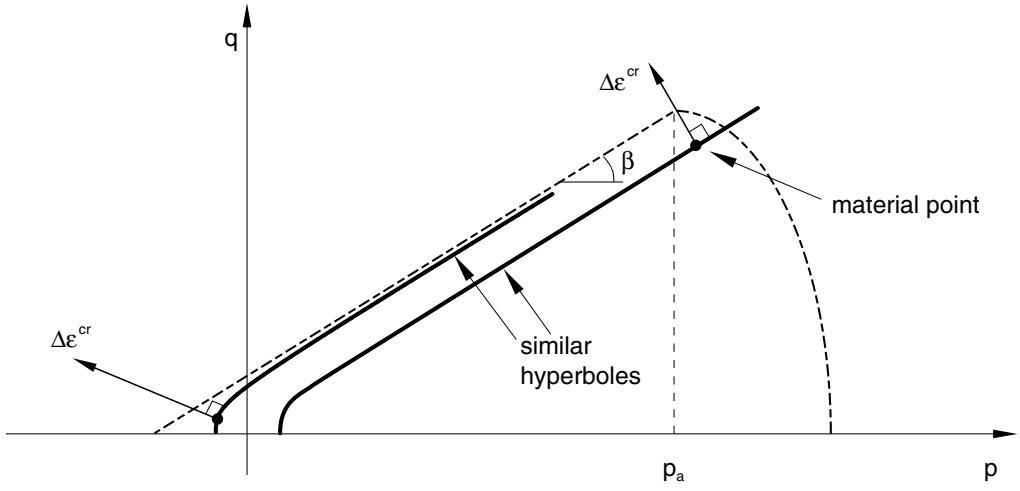


Figure 4.4.4–7 Creep potentials: cohesion mechanism.

The equivalent cohesion creep strain rate is then determined from the uniaxial law:

$$\Delta\bar{\varepsilon}_s^{cr} = h_s(\bar{\sigma}^{cr}, \bar{\varepsilon}_s^{cr}, \theta, f^{\alpha}).$$

Equation 4.4.4–1, Equation 4.4.4–3, and Equation 4.4.4–5 produce the flow rule for this mechanism

$$\Delta\boldsymbol{\varepsilon}_s^{cr} = \frac{\Delta\bar{\varepsilon}_s^{cr}}{f^{cr}}\left(\frac{q}{\sqrt{(0.1\frac{d}{(1-\frac{1}{3}\tan\beta)}\tan\beta)^2+q^2}}\mathbf{n}+\frac{1}{3}\tan\beta\mathbf{I}\right),$$

where

$$\mathbf{n} = \frac{\partial q}{\partial \boldsymbol{\sigma}} = \frac{3}{2}\frac{\boldsymbol{S}}{\boldsymbol{q}},$$

and

$$f^{cr} = \frac{\bar{\sigma}^{cr}}{\sqrt{(0.1\frac{d}{(1-\frac{1}{3}\tan\beta)}\tan\beta)^2+\left(\bar{\sigma}^{cr}\right)^2}} - \frac{1}{3}\tan\beta.$$

The proportionality factor, f^{cr}, is not a constant in this model. Its expression indicates that it will become negative if

$$\bar{\sigma}^{cr} < 0.1\frac{d}{(1-\frac{1}{3}\tan\beta)}\frac{\frac{1}{3}(\tan\beta)^2}{\sqrt{1-(\frac{1}{3}\tan\beta)^2}}.$$

It turns out that below this stress level, which for typical materials will be very low, the stress vector and the normal to the creep potential are pointing in opposite directions:

$$\boldsymbol{\sigma} : \frac{\partial G_s^{cr}}{\partial \boldsymbol{\sigma}} < 0,$$

which is equivalent to

$$q - p\tan\beta < q - \frac{q^2}{\sqrt{(0.1\frac{d}{(1-\frac{1}{3}\tan\beta)}\tan\beta)^2+q^2}}.$$

Thus, there is a small zone just outside the "no creep" cone for which this is the case. Consequently, creep data obtained within this zone should show a creep strain rate in the opposite direction from the applied stress at very low stress levels, which will usually not be the case. To overcome this difficulty, Abaqus will modify the creep data entered such that $f^{cr} \geq 0.1$. Therefore, you would not expect correspondence between calculated creep strains and measured creep properties in a region defined by

$$\bar{\sigma}^{cr} < 0.1\frac{d}{(1-\frac{1}{3}\tan\beta)}\frac{(0.1+\frac{1}{3}\tan\beta)\tan\beta}{\sqrt{1-(0.1+\frac{1}{3}\tan\beta)^2}}.$$

This modification is usually not significant, since typical creep analyses have loads that are applied quickly, followed by long-term creep. Hence, the stress level for most of the analysis will usually be well beyond the modified zone.

An example of "slow" loading in which the approximation is visible is included in "Verification of creep integration," Section 3.2.6 of the Abaqus Benchmarks Guide. As is clear in the example, the effect of the approximation is small in spite of the fact that the load is ramped up over the step.

The equivalent cohesion creep strain rate is a function of both q and p through $\bar{\sigma}^{cr}$. The creep potential is the von Mises circle in the deviatoric stress plane (the Π-plane). Although creep flow is associated in the deviatoric stress plane, the use of a creep potential different from the equivalent creep surface implies that creep flow is nonassociated.

Consolidation creep

For the consolidation mechanism the creep potential is derived from the plastic potential of the cap zone (Figure 4.4.4–8):

$$G_c^{cr} = \sqrt{(p - p_a)^2 + (Rq)^2}. \tag{4.4.4–6}$$

Recall that this mechanism is active only if $p \geq p_a$.

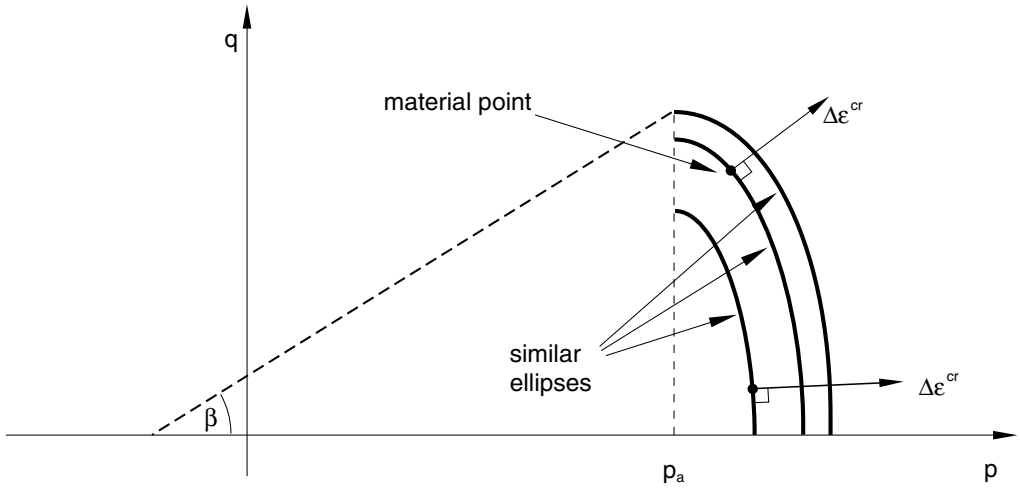


Figure 4.4.4–8 Creep potentials: consolidation mechanism.

The equivalent consolidation creep strain rate is then determined from the uniaxial law

$$\Delta\bar{\varepsilon}_c^{cr} = h_c(\bar{p}^{cr}, \bar{\varepsilon}_c^{cr}, \theta, f^{\alpha}).$$

Equation 4.4.4–3 and Equation 4.4.4–4 produce $f^{cr} = 1$; and Equation 4.4.4–2, Equation 4.4.4–3, and Equation 4.4.4–6 produce the flow rule for this mechanism:

$$\Delta\boldsymbol{\varepsilon}_c^{cr} = \frac{\Delta\bar{\varepsilon}_c^{cr}}{G_c^{cr}}(R^2\, q\, \mathbf{n} - \frac{1}{3}(p - p_a)\, \mathbf{I}).$$

Note that there is an equivalent pressure stress, $\bar{p}$, work conjugate of the equivalent consolidation creep strain, which is different from the effective creep pressure, $\bar{p}^{cr}$. Such equivalent pressure stress is given by

$$\bar{p} = \frac{R^2\, q^2 + p\,(p - p_a)}{G_c^{cr}}$$

and has the characteristic that it reduces to the pressure in a hydrostatic compression test.

The creep potential is the von Mises circle in the deviatoric stress plane (the Π-plane). Creep flow is nonassociated in this mechanism.

This formulation is quite simplistic and ignores the effects of q on the creep function, h_c. The two creep mechanisms operate independently from each other. This implies that h_s does not depend on $\bar{\varepsilon}_c^{cr}$ and that h_c does not depend on $\bar{\varepsilon}_s^{cr}$. The only cross effects between both mechanisms are obtained through the dependency of p_a on the volumetric creep from any of them.

Reference

- "Modified Drucker-Prager/Cap model," Section 23.3.2 of the Abaqus Analysis User's Guide

4.4.5 MOHR-COULOMB MODEL

Products: Abaqus/Standard Abaqus/Explicit

The Mohr-Coulomb failure or strength criterion has been widely used for geotechnical applications. Indeed, a large number of the routine design calculations in the geotechnical area are still performed using the Mohr-Coulomb criterion.

The Mohr-Coulomb criterion assumes that failure is controlled by the maximum shear stress and that this failure shear stress depends on the normal stress. This can be represented by plotting Mohr's circle for states of stress at failure in terms of the maximum and minimum principal stresses. The Mohr-Coulomb failure line is the best straight line that touch es these Mohr's circles (Figure 4.4.5–1). Thus, the Mohr-Coulomb criterion can be written as

$$\tau = c - \sigma \tan \phi,$$

where τ is the shear stress, σ is the normal stress (negative in compression), c is the cohesion of the material, and ϕ is the material angle of friction.

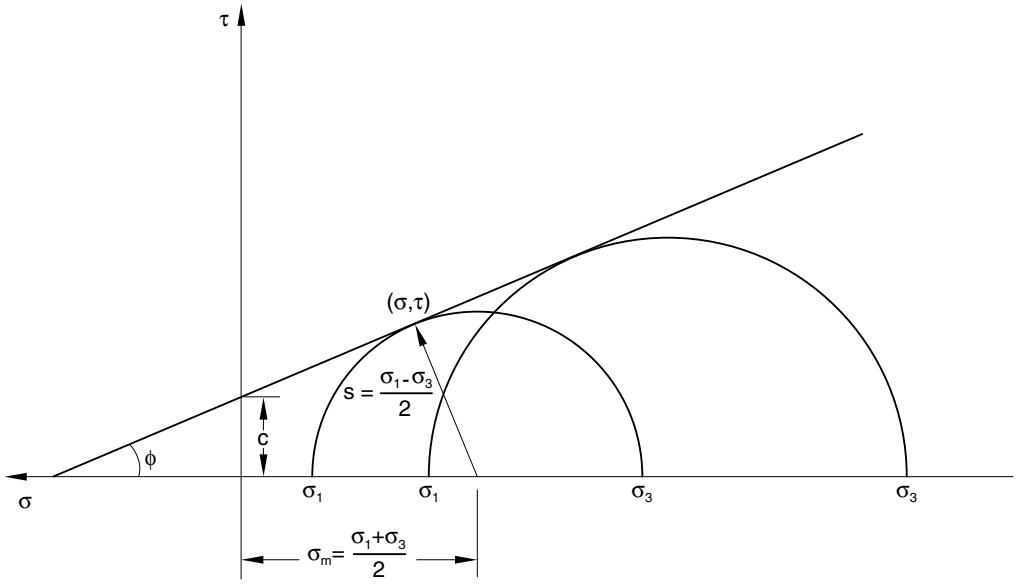


Figure 4.4.5–1 Mohr-Coulomb failure criterion.

From Mohr's circle,

$$\tau = s\cos\phi,$$
$$\sigma = \sigma_m + s\sin\phi.$$

Substituting for τ and σ, the Mohr-Coulomb criterion can be rewritten as

$$s + \sigma_m \sin\phi - c\cos\phi = 0,$$

where

$$s = \frac{1}{2}(\sigma_1 - \sigma_3)$$

is half of the difference between the maximum and minimum principal stresses (and is, therefore, the maximum shear stress) and

$$\sigma_m = \frac{1}{2}(\sigma_1 + \sigma_3)$$

is the average of the maximum and minimum principal stresses (the normal stress). Thus, unlike the Drucker-Prager criterion, the Mohr-Coulomb criterion assumes that failure is independent of the value of the intermediate principal stress. The failure of typical geotechnical materials generally includes some small dependence on the intermediate principal stress, but the Mohr-Coulomb model is generally considered to be sufficiently accurate for most applications. This failure model has vertices in the deviatoric stress plane (see Figure 4.4.5–2).

The constitutive model described here is an extension of the classical Mohr-Coulomb failure criterion. It is an elastoplastic model that uses a yield function of the Mohr-Coulomb form; this yield function includes isotropic cohesion hardening/softening. However, the model uses a flow potential that has a hyperbolic shape in the meridional stress plane and has no corners in the deviatoric stress space. This flow potential is then completely smooth and, therefore, provides a unique definition of the direction of plastic flow.

Strain rate decomposition

An additive strain rate decomposition is assumed:

$$d\boldsymbol{\varepsilon} = d\boldsymbol{\varepsilon}^{el} + d\boldsymbol{\varepsilon}^{pl}, \tag{4.4.5–1}$$

where $d\boldsymbol{\varepsilon}$ is the total strain rate, $d\boldsymbol{\varepsilon}^{el}$ is the elastic strain rate, and $d\boldsymbol{\varepsilon}^{pl}$ is the inelastic (plastic) strain rate.

Elastic behavior

The elastic behavior is modeled as linear and isotropic.

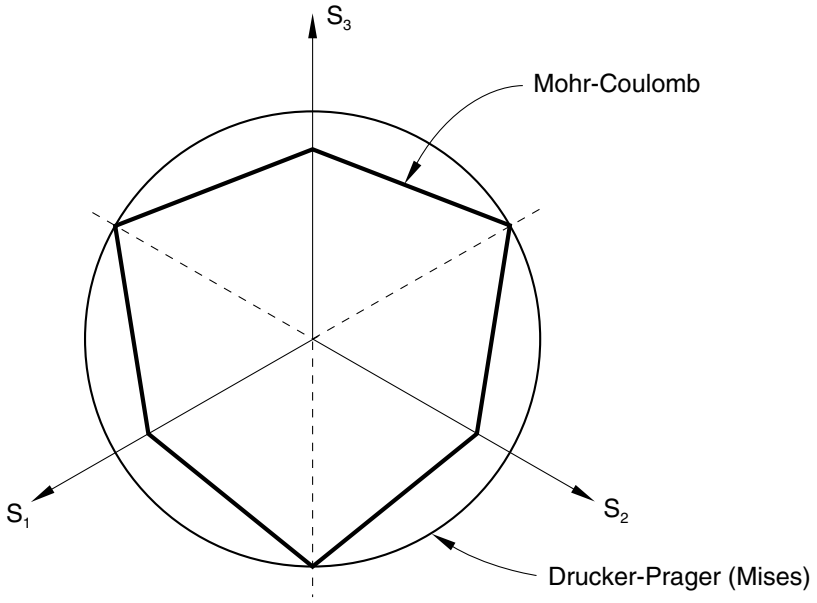


Figure 4.4.5–2 Mohr-Coulomb model in the deviatoric plane.

Yield behavior

The Mohr-Coulomb criterion written above in terms of the maximum and minimum principal stresses can be written for general states of stress in terms of three stress invariants. These invariants are the equivalent pressure stress,

$$p = -\frac{1}{3}\mathrm{trace}\,(\boldsymbol{\sigma});$$

the Mises equivalent stress,

$$q = \sqrt{\frac{3}{2}(\mathbf{S} : \mathbf{S})},$$

where $\mathbf{S}$ is the stress deviator, defined as

$$\mathbf{S} = \boldsymbol{\sigma} + \mathbf{pI};$$

and the third invariant of deviatoric stress,

$$r = (\frac{9}{2}\mathbf{S} \cdot \mathbf{S} : \mathbf{S})^{\frac{1}{3}}.$$

The Mohr-Coulomb yield surface is then written as

$$F = R_{mc}q - p\tan\phi - c = 0, \tag{4.4.5–2}$$

where $\phi(\theta, f^\alpha)$ is the friction angle of the material in the meridional stress plane, where θ is the temperature and $f^\alpha, \alpha = 1, 2...$ are other predefined field variables; $c(\bar{\varepsilon}^{pl}, \theta, f^\alpha)$ represents the evolution of the cohesion of the material in the form of isotropic hardening (or softening); $\bar{\varepsilon}^{pl}$ is the equivalent plastic strain, its rate defined by the plastic work expression

$$c\,\dot{\bar{\varepsilon}}^{pl} = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{pl};$$

and R_{mc} is the Mohr-Coulomb deviatoric stress measure defined as

$$R_{mc}(\Theta, \phi) = \frac{1}{\sqrt{3}\cos\phi}\sin\left(\Theta + \frac{\pi}{3}\right) + \frac{1}{3}\cos\left(\Theta + \frac{\pi}{3}\right)\tan\phi,$$

where Θ is the deviatoric polar angle (Chen and Han, 1988) defined as

$$\cos(3\Theta) = \left(\frac{r}{q}\right)^3.$$

The friction angle of the material, ϕ, also controls the shape of the yield surface in the deviatoric plane as shown in Figure 4.4.5–3. The range of values the friction angle can have is $0° \le \phi < 90°$. In the case of $\phi = 0°$ the Mohr-Coulomb model reduces to the pressure-independent Tresca model with a perfectly hexagonal deviatoric section. In the case of $\phi = 90°$ the Mohr-Coulomb model would reduce to the "tension cutoff" Rankine model with a triangular deviatoric section and $R_{mc} = \infty$ (this limiting case is not permitted within the Mohr-Coulomb model described here).

Flow rule

Potential flow is assumed, so

$$d\boldsymbol{\varepsilon}^{pl} = \frac{d\bar{\varepsilon}^{pl}}{g}\frac{\partial G}{\partial \boldsymbol{\sigma}}, \tag{4.4.5–3}$$

where g can be written as

$$g = \frac{1}{c}\,\boldsymbol{\sigma} : \frac{\partial G}{\partial \boldsymbol{\sigma}}$$

and G is the flow potential, chosen as a hyperbolic function in the meridional stress plane and a smooth elliptic function in the deviatoric stress plane:

$$G = \sqrt{(\epsilon c|_0 \tan\psi)^2 + (R_{mw}q)^2} - p\tan\psi, \tag{4.4.5–4}$$

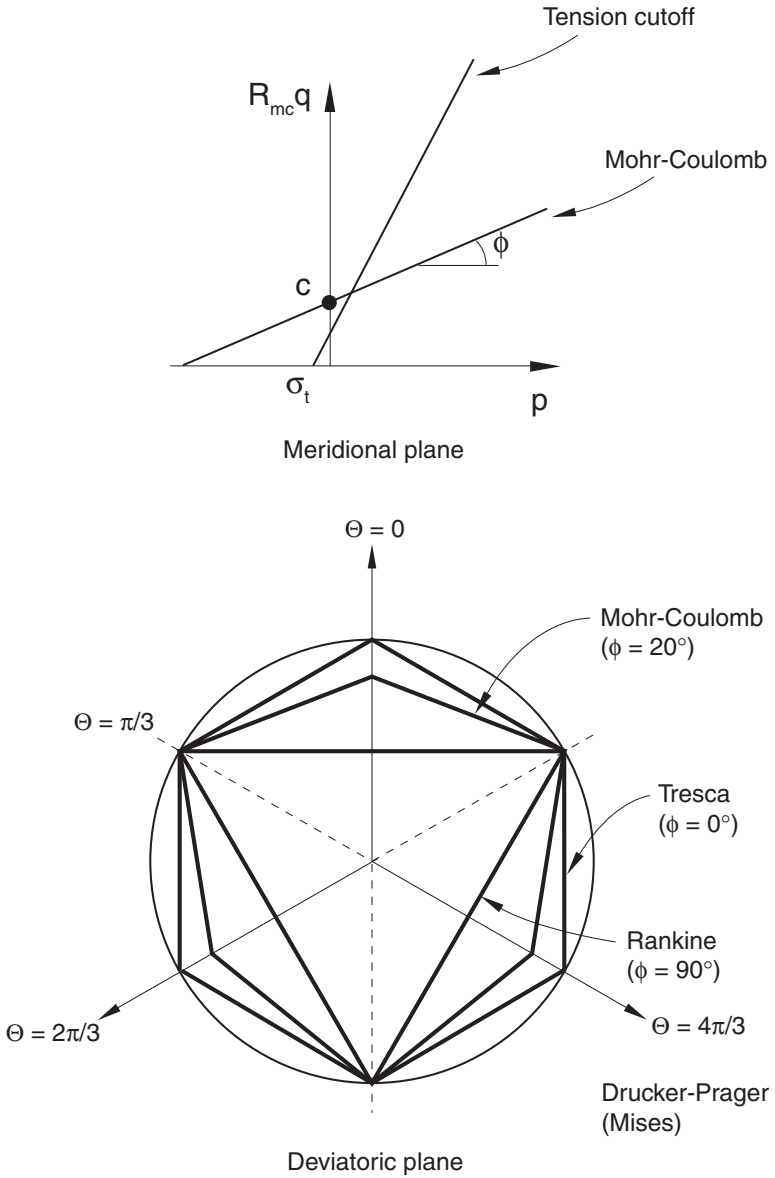


Figure 4.4.5–3 Mohr-Coulomb yield surface in meridional and deviatoric planes.

where $\psi(\theta, f^{\alpha})$ is the dilation angle measured in the $\boldsymbol{p}$–$R_{mw}q$ plane at high confining pressure; $c|_0 = c|_{\bar{\varepsilon}^{pl}=0}$ is the initial cohesion yield stress; and ϵ is a parameter, referred to as the eccentricity, that defines the rate at which the function approaches the asymptote (the flow potential tends to a straight line as

the eccentricity tends to zero). This flow potential, which is continuous and smooth in the meridional stress plane, ensures that the flow direction is defined uniquely in this plane. The function asymptotically approaches a linear flow potential at high confining pressure stress and intersects the hydrostatic pressure axis at 90°. A family of hyperbolic potentials in the meridional stress plane is shown in Figure 4.4.5–4.

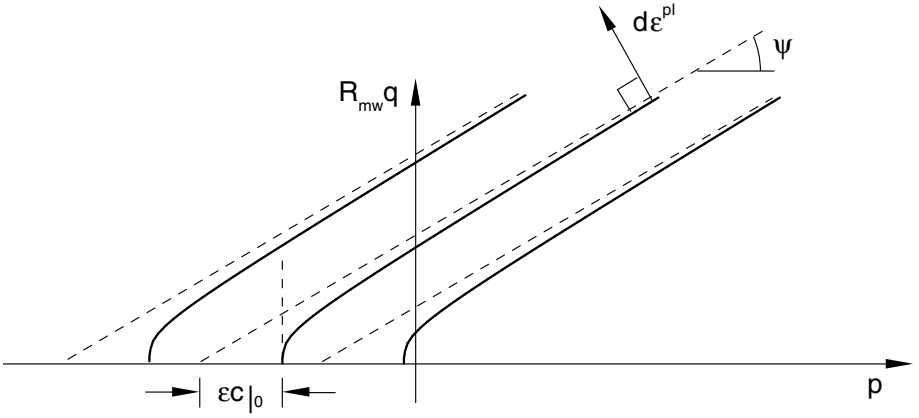


Figure 4.4.5–4 Family of hyperbolic flow potentials in the meridional plane.

The flow potential is also continuous and smooth in the deviatoric stress plane (the Π-plane); we adopt the deviatoric elliptic function used by Menétrey and Willam (1995):

$$R_{mw}(\Theta, e) = \frac{4(1-e^2)\cos^2\Theta + (2e-1)^2}{2(1-e^2)\cos\Theta + (2e-1)\sqrt{4(1-e^2)\cos^2\Theta + 5e^2 - 4e}}\, R_{mc}(\frac{\pi}{3}, \phi),$$

where Θ is the deviatoric polar angle defined previously, $R_{mc}(\frac{\pi}{3}, \phi) = (3 - \sin\phi)/6\cos\phi$, and e is a parameter that describes the "out-of-roundedness" of the deviatoric section in terms of the ratio between the shear stress along the extension meridian ($\Theta = 0$) and the shear stress along the compression meridian ($\Theta = \frac{\pi}{3}$). The elliptic function has the value $R_{mw}(\Theta = 0, e) = R_{mc}(\frac{\pi}{3}, \phi)/e$ along the extension meridian and has the value $R_{mw}(\Theta = \frac{\pi}{3}, e) = R_{mc}(\frac{\pi}{3}, \phi)$ along the compression meridian; this ensures that the flow potential matches the yield surface at the triaxial compression and extension in the deviatoric plane provided that e is defined appropriately (see further discussion later). Although the elliptic function is defined only in the sector $0 \le \Theta \le \pi/3$, the polar radius $R_{mw}(\Theta, e)$ extends to all polar directions $0 \le \Theta \le 2\pi$ using the three-fold symmetry shown in Figure 4.4.5–5.

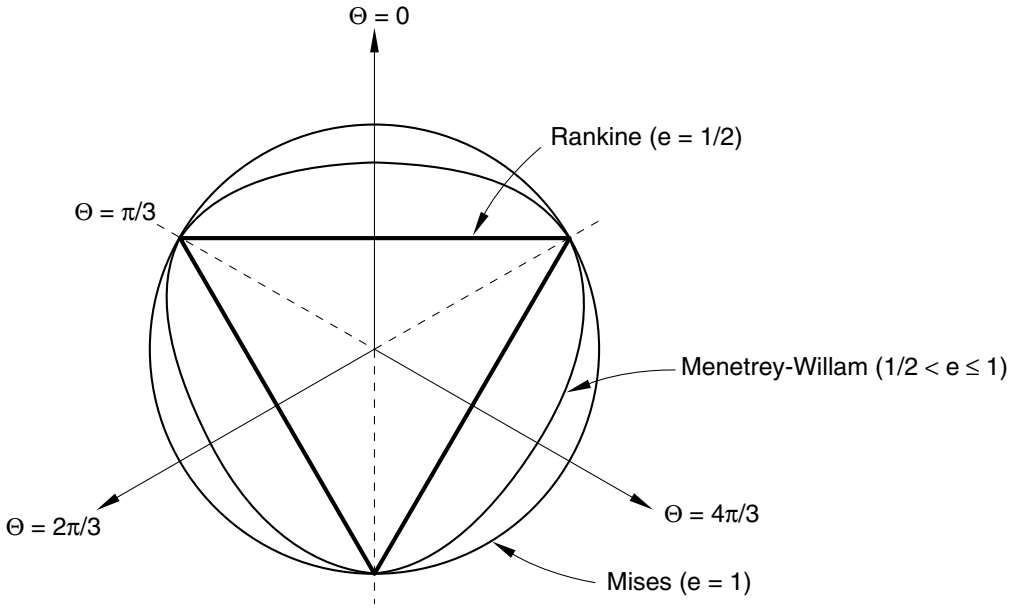


Figure 4.4.5–5 Menétrey-Willam flow potential in the deviatoric plane.

By default, the out-of-roundedness parameter, ***e***, is dependent on the friction angle ϕ; it is calculated by matching the flow potential to the yield surface in both triaxial tension and compression in the deviatoric plane:

$$e = \frac{3 - \sin\phi}{3 + \sin\phi}.$$

Alternatively, ***e*** can also be considered to be an independent material parameter; in this case the user can provide its value directly. Convexity and smoothness of the elliptic function requires that $1/2 < e \leq 1$. The upper limit, $e = 1$ (or $\phi = 0°$), leads to $R_{mw}(\Theta, e = 1) = R_{mc}(\frac{\pi}{3}, \phi)$, which describes the Mises circle in the deviatoric plane. The lower limit, $e = 1/2$ (or $\phi = 90°$), leads to $R_{mw}(\Theta, e = 1/2) = 2R_{mc}(\frac{\pi}{3}, \phi)\cos\Theta$ and would describe the Rankine triangle in the deviatoric plane (this limiting case is not permitted within the Mohr-Coulomb model described here).

Flow in the meridional stress plane can be close to associated when the angle of friction, ϕ, and the angle of dilation, ψ, are equal and the eccentricity parameter, ϵ, is very small; however, flow in this plane is, in general, nonassociated. Flow in the deviatoric stress plane is always nonassociated. Therefore, the use of this Mohr-Coulomb model generally requires the solution of nonsymmetric equations.

Reference

- "Mohr-Coulomb plasticity," Section 23.3.3 of the Abaqus Analysis User's Guide

4.4.6 MODELS FOR CRUSHABLE FOAMS

Products: Abaqus/Standard Abaqus/Explicit

The constitutive models described here are available in Abaqus for the analysis of crushable foams typically used in energy absorption structures. Two phenomenological constitutive models are presented: the volumetric hardening model and the isotropic hardening model. Both models use a yield surface with an elliptical dependence of deviatoric stress on pressure stress in the meridional plane.

The volumetric hardening model is motivated by the experimental observation that foam structures usually experience a different response in compression and tension. In compression the ability of the material to deform volumetrically is enhanced by cell wall buckling processes as described by Gibson et al. (1982), Gibson and Ashby (1982), and Maiti et al. (1984). It is assumed that the foam cell deformation is not recoverable instantaneously and can, thus, be idealized as being plastic for short duration events. In tension, on the other hand, cell walls break readily; and as a result the tensile load bearing capacity of crushable foams may be considerably smaller than its compressive load bearing capacity. Under monotonic loading, the volumetric hardening model assumes perfectly plastic behavior for pure shear and negative hydrostatic pressure stress states, while hardening takes place for positive hydrostatic pressure stress states.

The isotropic hardening model was originally developed for metallic foams by Deshpande and Fleck (2000). It assumes symmetric behavior in tension and compression, and the evolution of the yield surface is governed by an equivalent plastic strain, which has contributions from both the volumetric plastic strain and the deviatoric plastic strain.

The mechanical behavior of crushable foams is known to be sensitive to the rate of straining. This effect can be introduced by a piecewise linear law or by the overstress power law model.

The strain rate decomposition

The volume change is decomposed as

$$J = J^{el} \cdot J^{pl}, \tag{4.4.6–1}$$

where J is the ratio of current volume to original volume, J^{el} is the elastic (recoverable) part of the ratio of current to original foam volume, and J^{pl} is the plastic (nonrecoverable) part of the ratio of current to original foam volume.

Volumetric strains are defined as

$$\begin{aligned} \varepsilon_{\text{vol}} &= \ln J, \\ \varepsilon^{el}_{\text{vol}} &= \ln J^{el}, \\ \varepsilon^{pl}_{\text{vol}} &= \ln J^{pl}. \end{aligned}$$

These definitions and Equation 4.4.6–1 result in the usual additive strain rate decomposition for volumetric strains:

$$\dot{\varepsilon}_{\text{vol}} = \dot{\varepsilon}^{el}_{\text{vol}} + \dot{\varepsilon}^{pl}_{\text{vol}}.$$

The model also assumes the deviatoric strain rates decompose additively, so that the total strain rates decompose as

$$\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}}^{el} + \dot{\boldsymbol{\varepsilon}}^{pl}.$$

Elastic behavior

The elastic behavior can be modeled only as linear elastic

$$\boldsymbol{\sigma} = \mathbf{D}^{el} : \boldsymbol{\varepsilon}^{el},$$

where $\mathbf{D}^{el}$ represents the fourth-order elasticity tensor and $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}^{el}$ are the second-order stress and elastic strain tensors, respectively.

Plastic behavior

The yield surface and the flow potential for the crushable foam models are defined in terms of the pressure stress

$$p = -\frac{1}{3}\,\text{trace}\,\boldsymbol{\sigma} = -\frac{1}{3}\,\boldsymbol{\sigma} : \mathbf{I}$$

and the Mises stress

$$q = \sqrt{\frac{3}{2}\,\mathbf{S} : \mathbf{S}}.$$

The yield surface is defined as

$$F = \sqrt{q^2 + \alpha^2\,(p - p_0)^2} - B = 0 \tag{4.4.6–2}$$

and the flow potential is defined as

$$G = \sqrt{q^2 + \beta^2\,p^2}. \tag{4.4.6–3}$$

F and G can each be represented as an ellipse in the p–q stress plane with α and β representing the shape of the yield ellipse and the ellipse for the flow potential, respectively; p_0 is the center of the yield ellipse, and B is the length of the (vertical) q–axis of the yield ellipse. The flow potential is an ellipse centered in the origin. The yield surface and the flow potential are depicted in Figure 4.4.6–1.

The parameters p_0 and B of the yield ellipse (Equation 4.4.6–2) are related to the yield strength in hydrostatic compression, p_c, and to the yield strength in hydrostatic tension, p_t, by

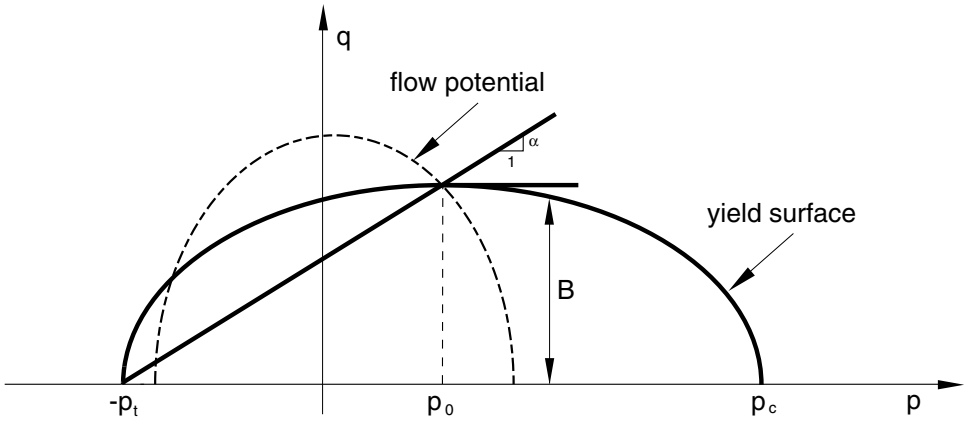


Figure 4.4.6–1 Typical yield surface and flow potential for the crushable foam model.

$$p_0 = \frac{p_c - p_t}{2} \qquad \text{and} \qquad B = \alpha\, A = \alpha\, \frac{p_c + p_t}{2},$$

where p_c and p_t are positive quantities and A is the length of the (horizontal) p-axis of the yield ellipse.

The shape factor, α, remains as a constant during any plastic deformation process. The evolution of the yield ellipse is controlled by a plastic strain measure, $\bar{\varepsilon}$, which is the volumetric compacting plastic strain, $-\varepsilon^{pl}_{\text{vol}}$, for the volumetric hardening model, and the equivalent plastic strain, $\bar{\varepsilon}^{pl}$ (to be defined later), for the isotropic hardening model.

To define the hardening behavior, uniaxial compression test data are required. A piecewise linear hardening curve of uniaxial Cauchy stress versus axial (logarithmic) plastic strain must be entered in a tabular form.

Crushable foam model with volumetric hardening

The volumetric hardening model assumes that the hydrostatic tension strength, p_t, remains constant throughout any plastic deformation process. By contrast, the hydrostatic compression strength evolves as a result of compaction (increase in density) or dilation (reduction in density) of the material:

$$p_c = p_c(\bar{\varepsilon}) \qquad \text{where} \qquad \bar{\varepsilon} \equiv -\varepsilon^{pl}_{\text{vol}} = -\text{trace}\, \boldsymbol{\varepsilon}^{pl}.$$

Yield surface

The yield surface for the crushable foam model, depicted in Figure 4.4.6–2, is defined by

$$F = \sqrt{q^2 + \alpha^2\, (p - p_0)^2} - B = 0,$$

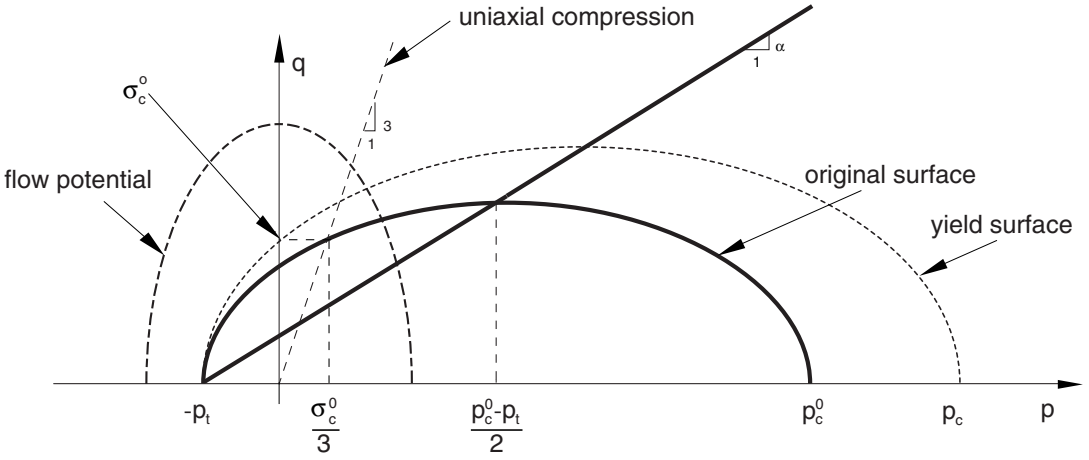


Figure 4.4.6–2 Yield surfaces and flow potential for the volumetric hardening model.

where the parameter α represents the shape of the yield ellipse in the p–q stress plane and can be calculated from the initial yield strength in uniaxial compression, σ_c^0, taken as a positive value; the initial yield strength in hydrostatic compression, p_c^0; and the yield strength in hydrostatic tension, p_t; as

$$\alpha = \frac{3k}{\sqrt{(3k_t + k)(3 - k)}} \qquad \text{with} \qquad k = \frac{\sigma_c^0}{p_c^0} \quad \text{and} \quad k_t = \frac{p_t}{p_c^0},$$

where the yield stress ratios, $k(\theta, f_i)$ and $k_t(\theta, f_i)$, are provided by the user and can be functions of temperature and other field variables. For a valid yield surface the choice of yield stress ratios must be such that $0 < k < 3$ and $k_t \geq 0$. The yield surface is the Mises circle in the deviatoric stress plane.

Flow potential

The plastic strain rate for the volumetric hardening model is assumed to be

$$\dot{\boldsymbol{\varepsilon}}^{pl} = \dot{\bar{\varepsilon}}^{pl} \frac{\partial G}{\partial \boldsymbol{\sigma}},$$

where $\dot{\bar{\varepsilon}}^{pl}$ is the equivalent plastic strain rate defined as

$$\dot{\bar{\varepsilon}}^{pl} = \frac{\boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{pl}}{G},$$

and G is the flow potential, chosen in this model as

$$G = \sqrt{q^2 + \frac{9}{2} p^2}. \tag{4.4.6–4}$$

This potential is a particular case of Equation 4.4.6–3 with $\beta = 3/\sqrt{2} \approx 2.12$. A geometrical representation of the flow potential in the ***p*–*q*** stress plane is shown in Figure 4.4.6–2.

Equation 4.4.6–4 gives a direction of flow that is identical to the stress direction for radial paths. This is motivated by simple laboratory experiments performed by Bilkhu (1987), which suggest that loading in any principal direction causes insignificant deformation in the other directions. As a result, the plastic flow is nonassociative. Therefore, the use of this foam model generally requires the solution of nonsymmetric equations.

Hardening

The yield surface intersects the ***p***-axis at $-p_t$ and p_c. We assume that p_t remains fixed throughout any plastic deformation process. By contrast, the compressive strength, p_c, evolves as a result of compaction (increase in density) or dilation (reduction in density) of the material. The evolution of the yield surface can be expressed through the evolution of the yield surface size on the hydrostatic stress axis, $p_c + p_t$, as a function of the value of volumetric compacting plastic strain, $-\varepsilon^{pl}_{\text{vol}}$. With p_t constant, this relation can be obtained from a user-provided uniaxial compression test data using

$$p_c(\varepsilon^{pl}_{\text{vol}}) = \frac{\sigma_c(\varepsilon^{pl}_{\text{axial}}) \left[\sigma_c(\varepsilon^{pt}_{\text{axial}}) \left(\frac{1}{\alpha^2} + \frac{1}{9} \right) + \frac{p_t}{3} \right]}{p_t + \frac{\sigma_c(\varepsilon^{pl}_{\text{axial}})}{3}}$$

along with the fact that $\varepsilon^{pl}_{\text{axial}} = \varepsilon^{pl}_{\text{vol}}$ in uniaxial compression (due to zero plastic Poisson's ratio). Thus, the user provides input to the hardening law by only specifying, in the usual tabular form, the value of the yield stress in uniaxial compression as a function of the absolute value of the axial plastic strain. The table must start with a zero plastic strain (corresponding to the virgin state of the material), and the tabular entries must be given in ascending magnitude of $\varepsilon^{pl}_{\text{axial}}$. If desired, the yield stress can also be a function of temperature and other predefined field variables.

Crushable foam model with isotropic hardening

The isotropic hardening model was originally developed for metallic foams by Deshpande and Fleck (2000). The model assumes similar behaviors in tension and compression. The yield surface is an ellipse centered at the origin in the ***p*–*q*** stress plane and evolves in a self-similar manner governed by the equivalent plastic strain.

Yield surface

The yield surface for the isotropic hardening model is defined as

$$F = \sqrt{q^2 + \alpha^2 p^2} - B = 0 \tag{4.4.6–5}$$

with

$$B = \alpha\, p_c = \sigma_c \sqrt{1 + \left(\frac{\alpha}{3}\right)^2},$$

where α represents the shape of the yield ellipse in the p–q stress plane, B defines the size of the yield ellipse, p_c is the yield strength in hydrostatic compression, and σ_c is the absolute value of the yield strength in uniaxial compression. The yield surface is the Mises circle in the deviatoric stress plane, and an ellipse in the meridional plane as depicted in Figure 4.4.6–3.

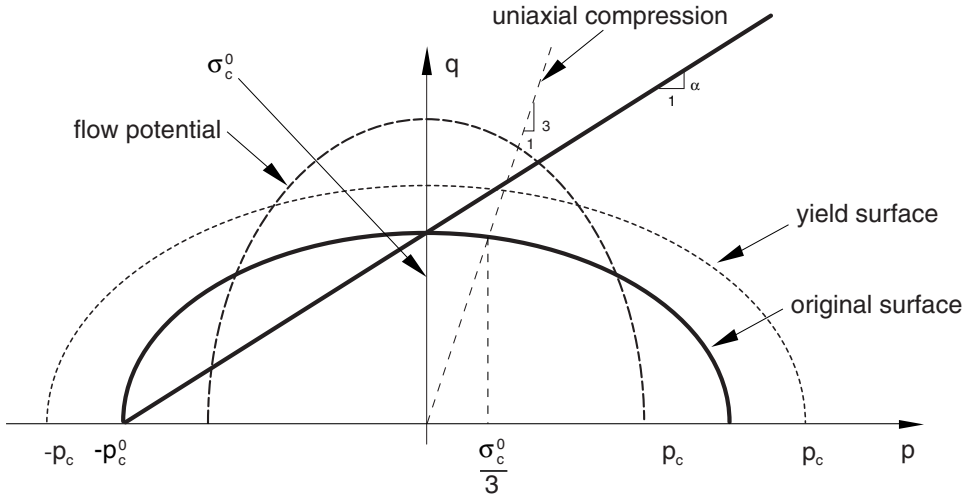


Figure 4.4.6–3 Yield surfaces and flow potential for the isotropic hardening model.

The parameter α can be calculated using the initial yield stress in uniaxial compression, σ_c^0, and the initial yield stress in hydrostatic compression, p_c^0, as

$$\alpha = \frac{3k}{\sqrt{9 - k^2}} \qquad \text{with} \qquad k = \frac{\sigma_c^0}{p_c^0}.$$

The strength ratio k is provided by the user and must be in the range of 0 and 3. For many low-density foams the initial yield surface is close to a circle in the p–q stress plane, which indicates that the value of α is approximately one. The special case of $k = 0$ corresponds to the Mises yield surface.

Flow potential

The flow potential for the isotropic hardening model is chosen as

$$G = \sqrt{q^2 + \beta^2\, p^2},$$

where β represents the shape of the flow potential in the p–q stress plane and is related to the plastic Poisson's ratio, ν_p, by

$$\beta = \frac{3}{\sqrt{2}} \sqrt{\frac{1 - 2\nu_p}{1 + \nu_p}}.$$

The plastic Poisson's ratio, which is the ratio of the transverse to the longitudinal plastic strain under uniaxial compression, should be defined by the user; and it must be in the range of -1 and 0.5. The upper limit, $\nu_p = 0.5$, corresponds to an incompressible plastic flow.

The plastic strains are defined to be normal to a family of self-similar flow potentials parametrized by the value of the potential G

$$\dot{\boldsymbol{\varepsilon}}^{pl} = \dot{\lambda} \frac{\partial G}{\partial \boldsymbol{\sigma}},$$

where $\dot{\lambda}$ is the nonnegative plastic flow multiplier. The hardening of the foam is described through $\sigma_c = \sigma_c(\bar{\varepsilon}^{pl})$, where $\bar{\varepsilon}^{pl}$ is the equivalent plastic strain. The evolution of $\bar{\varepsilon}^{pl}$ is assumed to be governed by the equivalent plastic work expression; i.e.,

$$\sigma_c \dot{\bar{\varepsilon}}^{pl} = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{pl}.$$

The equivalent plastic strain is equal to the absolute value of the axial plastic strain in uniaxial tension or compression.

The plastic flow is associative when the value of β is the same as that of α. In general, the plastic flow is nonassociated to allow for the independent calibrations of the shape of the yield surface and the plastic Poisson's ratio. For many low-density foams the plastic Poisson's ratio is nearly zero, which corresponds to a value of $\beta \approx 2.12$.

Hardening

A simple uniaxial compression test is sufficient to define the evolution of the yield surface. The hardening law defines the value of the yield stress in uniaxial compression as a function of the absolute value of the axial plastic strain. The piecewise linear relationship is entered in tabular form. The table must start with a zero plastic strain (corresponding to the virgin state of the materials) and must be given in ascending magnitude of $\varepsilon^{pl}_{\text{axial}}$. If desired, the yield stress can also be a function of temperature and other predefined field variables.

Reference

- "Crushable foam plasticity models," Section 23.3.5 of the Abaqus Analysis User's Guide

4.5 Other inelastic models

- "An inelastic constitutive model for concrete," Section 4.5.1
- "Damaged plasticity model for concrete and other quasi-brittle materials," Section 4.5.2
- "A cracking model for concrete and other brittle materials," Section 4.5.3
- "Constitutive model for jointed materials," Section 4.5.4

4.5.1 AN INELASTIC CONSTITUTIVE MODEL FOR CONCRETE

Product: Abaqus/Standard

This section describes the smeared crack model provided in Abaqus/Standard for plain concrete. The material library in Abaqus also includes a constitutive model for concrete based on theories of scalar plastic damage, described in "Damaged plasticity model for concrete and other quasi-brittle materials," Section 4.5.2, which is available both in Abaqus/Standard and Abaqus/Explicit. In Abaqus/Explicit plain concrete can also be analyzed with the cracking model described in "A cracking model for concrete and other brittle materials," Section 4.5.3. It is intended that reinforced concrete modeling be accomplished by combining standard elements, using this plain concrete model, with "rebar elements"—rods, defined singly or embedded in oriented surfaces, that use a one-dimensional strain theory and that may be used to model the reinforcing itself. These elements are superposed on the mesh of plain concrete elements and are used with standard metal plasticity models that describe the behavior of the rebar material. This modeling approach allows the concrete behavior to be considered independently of the rebar, so this section discusses the plain concrete model only. Effects associated with the rebar/concrete interface, such as bond slip and dowel action, cannot be considered in this approach, except by modifying some aspects of the plain concrete behavior to mimic them, such as the use of "tension stiffening" to simulate load transfer across cracks through the rebar.

The theory described in this section is intended as a model of concrete behavior for relatively monotonic loadings under fairly low confining pressures (less than four to five times the largest compressive stress that can be carried by the concrete in uniaxial compression). Cracking is assumed to be the most important aspect of the behavior, and it dominates the modeling. Cracking is assumed to occur when the stresses reach a failure surface, which we call the "crack detection surface." This failure surface is taken to be a simple Coulomb line written in terms of the first and second stress invariants, p and q, that are defined below. The anisotropy introduced by cracking is assumed to be important in the simulations for which the model is intended, so the model includes consideration of this anisotropy. The model is a smeared crack model, in the sense that it does not track individual "macro" cracks: rather, constitutive calculations are performed independently at each integration point of the finite element model, and the presence of cracks enters into these calculations by the way the cracks affect the stress and material stiffness associated with the integration point. Various objections have been raised against such smeared crack models. The principal concern is that this modeling approach inherently introduces mesh sensitivity in the solutions, in the sense that the finite element results do not converge to a unique result. For example, since cracking is associated with strain softening, mesh refinement will lead to narrower crack bands. Crisfield (1986) discusses this concern in detail and concludes that Hillerborg's (1976) approach, based on brittle fracture concepts, is adequate to deal with this issue for practical purposes. This aspect of the model is discussed below in the section on cracking. For simplicity of discussion in what follows, the term "crack" is used to mean a direction in which cracking has been detected at the single constitutive calculation point in question: the closest physical concept is that there exists a continuum of micro-cracks at the point, oriented as determined by the model.

When the principal stress components are dominantly compressive, the response of the concrete is modeled by an elastic-plastic theory, using a simple form of yield surface written in terms of the first two stress invariants. Associated flow and isotropic hardening are used. This model significantly simplifies

the actual behavior: the associated flow assumption generally overpredicts the inelastic volume strain; the simple yield surface used does not match all data very accurately (the third stress invariant would be needed to improve this aspect of the model); and, especially when the concrete is strained beyond the ultimate stress point, the assumption of constant elastic stiffness does not reproduce the observation that the unloading response is significantly weakened (the elastic response of the material appears to be damaged). In addition, when concrete is subjected to very high pressure stress, it exhibits inelastic response: no attempt has been made to build this behavior into the model. In spite of these limitations the model provides useful predictions for a variety of problems involving inelastic loading of concrete. The limitations are introduced for the sake of computational efficiency. In particular, assuming associated flow leads to enough symmetry in the Jacobian matrix of the constitutive model (the "material stiffness matrix") that the overall equilibrium equation solution usually does not require nonsymmetric equation solution for this reason. All of these limitations could be removed at some sacrifice in computational cost.

The cracking and compression responses of concrete that are incorporated in the model are illustrated by the uniaxial response of a specimen shown in Figure 4.5.1–1.

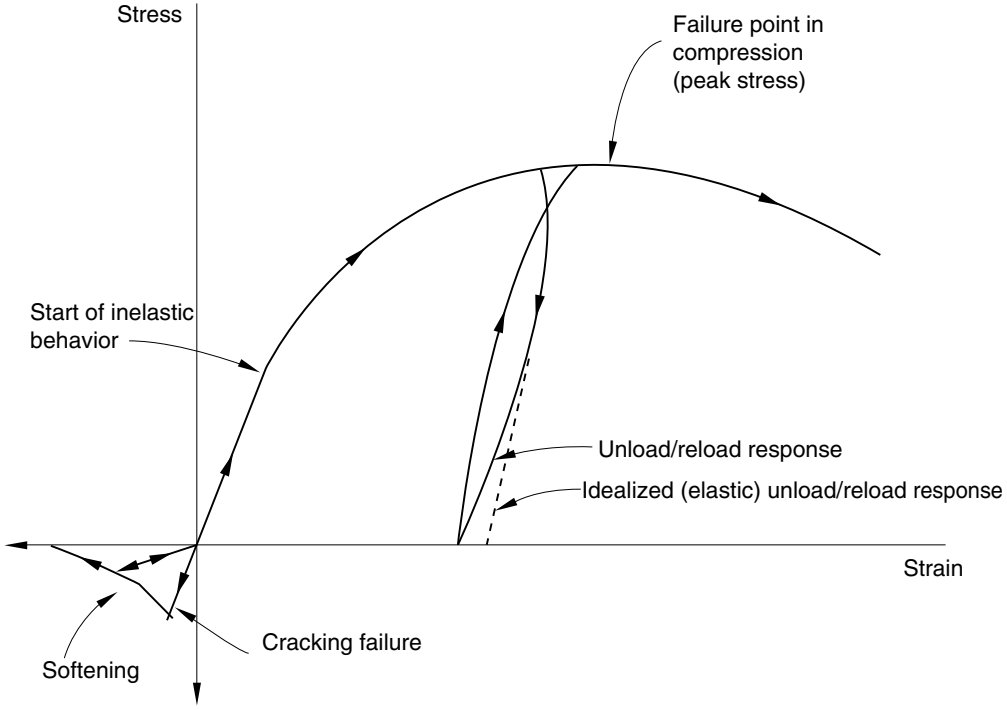


Figure 4.5.1–1 Uniaxial behavior of plain concrete.

When concrete is loaded in compression, it initially exhibits elastic response. As the stress is increased, some nonrecoverable (inelastic) straining occurs, and the response of the material softens. An ultimate stress is reached, after which the material softens until it can no longer carry any stress. If the load is removed at some

point after inelastic straining has occurred, the unloading response is softer than the initial elastic response: this effect is ignored in the model. When a uniaxial specimen is loaded into tension, it responds elastically until, at a stress that is typically 7–10% of the ultimate compressive stress, cracks form so quickly that—even on the stiffest testing machines available—it is very difficult to observe the actual behavior. For the purpose of developing the model, we assume that the material loses strength through a softening mechanism and that this is dominantly a damage effect, in the sense that open cracks can be represented by loss of elastic stiffness (as distinct from the nonrecoverable straining that is associated with classical plasticity effects, such as what we are using for the compressive behavior model). The model neglects any permanent strain associated with cracking; that is, we assume that the cracks can close completely when the stress across them becomes compressive.

In multiaxial stress states these observations can be generalized through the concept of surfaces of failure and of ultimate strength in stress space. These surfaces are defined below and are fitted to experimental data. Typical surfaces are shown in Figure 4.5.1–2 and Figure 4.5.1–3.

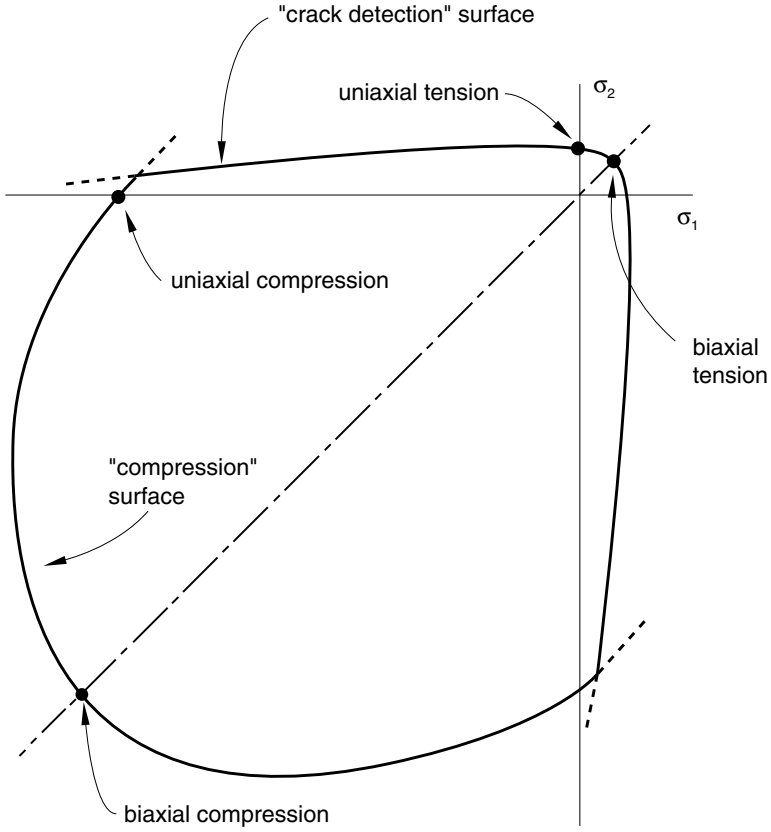


Figure 4.5.1–2 Concrete failure surfaces in plane stress.

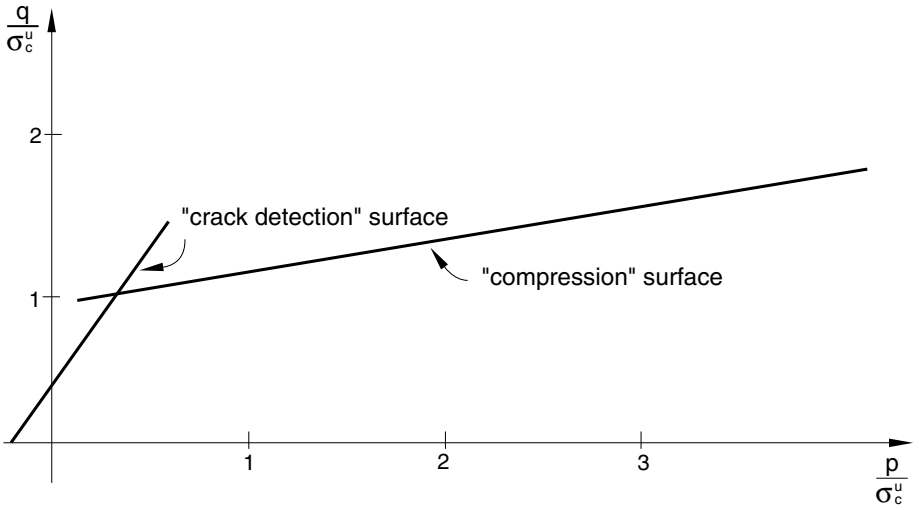


Figure 4.5.1–3 Concrete failure surfaces in the (p–q) plane.

This model makes no attempt to include prediction of cyclic response or of the reduction in the elastic stiffness caused by inelastic straining because the model is intended for application to relatively monotonic loading cases. Nevertheless, it is likely that—even in such cases—the stress trajectories will not be entirely radial and the model must predict the response in such cases in a reasonable way. An isotropically hardening "compressive" yield surface forms the basis of the model for the inelastic response when the principal stresses are dominantly compressive. In tension once cracking is defined to occur (by the "crack detection surface" of the model), the orientation of the cracks is stored and oriented, damaged elasticity is then used to model the existing cracks. Stress components associated with an open crack are not included in the definition of the crack detection surface for detecting additional cracks at the same point, and we only allow cracks to form in orthogonal directions at a point.

Since Abaqus/Standard is an implicit, stiffness method code and the material calculations used to define the behavior of the concrete are carried out independently at each integration point in that part of the model that is made of concrete, the solution is known at the start of the time increment. The constitutive calculations must provide values of stress and material stiffness at the end of the increment, based on the current estimate of the kinematic solution for the response at the spatial integration point during the increment that provides the (logarithmic) strain, ε, at the end of the increment.

Once cracks exist at a point, the component forms of all vector and tensor valued quantities are rotated so that they lie in the local system defined by the crack orientation vectors (the normals to the crack faces). The model ensures that these crack face normal vectors will be orthogonal, so that this local system is rectangular Cartesian. This use of a local system simplifies the computation of the damaged elasticity used for the components associated with existing cracks.

The model, thus, consists of a "compressive" yield/flow surface to model the concrete response in predominantly compressive states of stress, together with damaged elasticity to represent cracks that have occurred at a material calculation point, the occurrence of cracks being defined by a "crack detection" failure surface that is considered to be part of the elasticity. The details of this model are now presented.

Elastic-plastic model for concrete

The model uses the classical concepts of plasticity theory: a strain rate decomposition into elastic and inelastic strain rates, elasticity, yield, flow, and hardening.

Strain rate decomposition

We begin with a strain rate decomposition:

$$d\boldsymbol{\varepsilon} = d\boldsymbol{\varepsilon}^{el} + d\boldsymbol{\varepsilon}_c^{pl}, \tag{4.5.1–1}$$

where $d\boldsymbol{\varepsilon}$ is the total mechanical strain rate, $d\boldsymbol{\varepsilon}^{el}$ is the elastic strain rate (which includes crack detection strains—this elastic strain will be further decomposed when we describe the elasticity), and $d\boldsymbol{\varepsilon}_c^{pl}$ is the plastic strain rate associated with the "compression" surface.

We assume that the elastic part of the strain is always small, so this equation can be integrated as

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{el} + \boldsymbol{\varepsilon}_c^{pl}. \tag{4.5.1–2}$$

Compression yield

The "compression" surface is

$$f_c = q - \sqrt{3}a_0 p - \sqrt{3}\tau_c = 0, \tag{4.5.1–3}$$

where $\boldsymbol{p}$ is the effective pressure stress, defined as

$$p = -\frac{1}{3}\text{trace}(\boldsymbol{\sigma}),$$

and $\boldsymbol{q}$ is the Mises equivalent deviatoric stress:

$$q = \sqrt{\frac{3}{2}\mathbf{S} : \mathbf{S}},$$

where $\mathbf{S} = \boldsymbol{\sigma} + p\,\mathbf{I}$ are the deviatoric stress components; a_0 is a constant, which is chosen from the ratio of the ultimate stress reached in biaxial compression to the ultimate stress reached in uniaxial compression; and $\tau_c(\lambda_c)$ is a hardening parameter (τ_c is the size of the yield surface on the $\boldsymbol{q}$-axis at $p = 0$, so that τ_c is the yield stress in a state of pure shear stress when all components of $\boldsymbol{\sigma}$ are zero except $\sigma_{12} = \sigma_{21} = \tau_c$). The hardening is measured by the value of λ_c: the $\tau_c(\lambda_c)$ relationship is user-specified.

This simple surface is a straight line in ($\boldsymbol{p}$–$\boldsymbol{q}$) space and provides a good match to experimental data over a fairly wide range of pressure stress values (up to four to five times the maximum compressive stress that can be carried by the concrete in uniaxial compression). This form of the surface means that, as the hardening (λ_c) changes, the surfaces in ($\boldsymbol{p}$–$\boldsymbol{q}$) space are similar, so—for example—the ratio of flow stress in biaxial loading to flow stress in uniaxial loading is the same at all flow stress levels. This does not appear to be contradicted by any experimental data, and it means that only one constant (a_0) is needed to define the shape of the surface.

The value of a_0 is established from the user's data as follows. In uniaxial compression $p = \frac{1}{3}\sigma_c$ and $q = \sigma_c$, where σ_c is the stress magnitude. Therefore, on $f_c = 0$,

$$\frac{\tau_c}{\sigma_c} = \left(\frac{1}{\sqrt{3}} - \frac{a_0}{3} \right). \tag{4.5.1–4}$$

In biaxial compression $p = 2/3\ \sigma_{bc}$ and $q = \sigma_{bc}$, where σ_{bc} is the magnitude of each nonzero principal stress. Therefore, on $f_c = 0$,

$$\frac{\tau_c}{\sigma_{bc}} = \left(\frac{1}{\sqrt{3}} - \frac{2a_0}{3} \right). \tag{4.5.1–5}$$

The value of $\sigma_{bc}^u/\sigma_c^u = r_{bc}^\sigma$ is specified by the user as part of the failure surface data (typically $r_{bc}^\sigma \approx 1.16$). a_0 can be calculated from Equation 4.5.1–4 and Equation 4.5.1–5 as

$$a_0 = \sqrt{3}\,\frac{1 - r_{bc}^\sigma}{1 - 2r_{bc}^\sigma}.$$

The "compression" surface is shown in Figure 4.5.1–2 and Figure 4.5.1–3.

Hardening

The user defines the hardening by specifying the magnitude of the stress, $\sigma_{11}|$, in a uniaxial compression test as a function of the inelastic strain magnitude, $\varepsilon_{11}|$. These data are used to define the $\tau_c(\lambda_c)$ relationship, as follows.

In uniaxial compression $p = \frac{1}{3}\sigma_c$ and $q = \sigma_c$, where σ_c is the stress magnitude. During active plastic loading $f_c = 0$, so by using the definition of f_c (Equation 4.5.1–4), we obtain τ_c immediately as

$$\tau_c = \left(\frac{1}{\sqrt{3}} - \frac{a_0}{3} \right) \sigma_c. \tag{4.5.1–6}$$

Flow

The model uses associated flow, so if $f_c = 0$ and $d\lambda_c > 0$,

$$d\boldsymbol{\varepsilon}_c^{pl} = d\lambda_c \left(1 + c_0 \left(\frac{p}{\sigma_c} \right)^2 \right) \frac{\partial f_c}{\partial \boldsymbol{\sigma}}; \tag{4.5.1–7}$$

otherwise, $d\boldsymbol{\varepsilon}_c^{pl} = 0$.

In the definition of $d\boldsymbol{\varepsilon}_c^{pl}$, c_0 is a constant that is chosen so that the ratio of ε_{11}^{pl} in a monotonically loaded biaxial compression test to ε_{11}^{pl} in a monotonically loaded uniaxial compression test is r_{bc}^{ε}, a value specified by the user as part of the failure surface data (typically $r_{bc}^{\varepsilon} \approx 1.28$). The equation defining c_0 from r_{bc}^{ε} and the other constants in the compression surface is derived next.

The gradient of the flow potential for the compressive surface is

$$\frac{\partial f_c}{\partial \boldsymbol{\sigma}} = \frac{\partial q}{\partial \boldsymbol{\sigma}} - \sqrt{3}\, a_0 \frac{\partial p}{\partial \boldsymbol{\sigma}}.$$

Since

$$\frac{\partial p}{\partial \boldsymbol{\sigma}} = -\frac{1}{3}\mathbf{I}$$

and

$$\frac{\partial q}{\partial \boldsymbol{\sigma}} = \frac{3}{2}\frac{\mathbf{S}}{q},$$

then

$$\frac{\partial f_c}{\partial \boldsymbol{\sigma}} = \frac{3}{2}\frac{\mathbf{S}}{q} + \frac{a_0}{\sqrt{3}}\mathbf{I}.$$

In uniaxial compression $p = \frac{1}{3}\sigma_c$, $q = \sigma_c$, and $S_{11} = -\frac{2}{3}\sigma_c$, so Equation 4.5.1–7 defines

$$\left(d\varepsilon_c^{pl}\right)_{11}^{c} = d\lambda_c \left(1 + \frac{c_0}{9}\right)\left(\frac{a_0}{\sqrt{3}} - 1\right). \tag{4.5.1–8}$$

This equation can be integrated immediately to give

$$\left(\varepsilon_c^{pl}\right)_{11}^{c} = \lambda_c \left(1 + \frac{c_0}{9}\right)\left(\frac{a_0}{\sqrt{3}} - 1\right), \tag{4.5.1–9}$$

so λ_c is known from $\left(\varepsilon_c^{pl}\right)_{11}^{c}$ and the constants a_0 and c_0. Equation 4.5.1–6 and Equation 4.5.1–9, therefore, define the $\tau_c(\lambda_c)$ relationship from the concrete model input data once c_0 is known.

The constant c_0 is calculated from the user's definition of r_{bc}^{ε}, the ratio of $\left(\varepsilon_c^{pl}\right)_{11}^{bc}$ to $\left(\varepsilon_c^{pl}\right)_{11}^{c}$, the total plastic strain components that would occur in monotonically loaded biaxial and uniaxial compression tests. In biaxial compression when both nonzero principal stresses have the magnitude σ_{bc}, $p = \frac{2}{3}\sigma_{bc} = \frac{2}{3}\, r_{bc}^{\sigma}\, \sigma_c$, $q = \sigma_{bc} = r_{bc}^{\sigma}\, \sigma_c$, and $S_{11} = -\frac{1}{3}\, r_{bc}^{\sigma}\, \sigma_c$, so the flow rule gives

$$\left(d\varepsilon_c^{pl}\right)_{11}^{bc} = d\lambda_c \left(1 + \frac{4}{9}\left(r_{bc}^{\sigma}\right)^2 c_0\right)\left(\frac{a_0}{\sqrt{3}} - \frac{1}{2}\right).$$

Using this equation and Equation 4.5.1–8 then defines c_0 from r_{bc}^{ε} and the other constants as

$$c_0 = 9\,\frac{r_{bc}^{\varepsilon}(\sqrt{3}-a_0)+(a_0-\sqrt{3}/2)}{r_{bc}^{\varepsilon}(a_0-\sqrt{3})+(r_{bc}^{\sigma})^2(2\sqrt{3}-4a_0)}\ .$$

Crack detection and damaged elasticity

Cracking dominates the material behavior when the state of stress is predominantly tensile. The model uses a "crack detection" plasticity surface in stress space to determine when cracking takes place and the orientation of the cracking. Damaged elasticity is then used to describe the postfailure behavior of the concrete with open cracks.

Numerically we use the "crack detection" plasticity model for the increment in which cracking takes place and subsequently use damaged elasticity once the crack's presence and orientation have been detected. As a result there is at least one increment in which we calculate crack detection "plastic" strains. Since these are really just the outcome of a numerical device to treat cracking, they are recast as elastic strains in the direction of cracking and as plastic strains in the other directions. (This means that we retain the stresses calculated for equilibrium purposes, as well as the strain decomposition of Equation 4.5.1–1.)

The basis of the postcracked behavior is the brittle fracture concept of Hilleborg (1976). We assume that the fracture energy required to form a unit area of crack surface, G_f, is a material property. This value can be calculated from measuring the tensile stress as a function of the crack opening displacement (Figure 4.5.1–4), as

$$G_f = \int \sigma_t \, du.$$

Typical values of G_f range from 40 N/m (0.22 lb/in) for a typical construction concrete (with $\sigma_c^u \approx 20$ MPa, 2850 lb/in^2) to 120 N/m (0.67 lb/in) for a high strength concrete (with $\sigma_c^u \approx 40$ MPa, 5700 lb/in^2).

The implication of assuming that G_f is a material property is that, when the elastic part of the displacement, u^{el}, is eliminated, the relationship between the stress and the remaining part of the displacement, $u^{cr} = u - u^{el}$, is fixed, regardless of the specimen size. For example, consider a specimen developing a single crack across its section as tensile displacement is applied to it: u^{cr} is the displacement across the crack and is not changed by using a longer or shorter specimen in the test (so long as the specimen is significantly longer than the width of the crack band, which will typically be of the order of the aggregate size). Thus, one important part of the cracked concrete's tensile behavior is defined in terms of a stress/displacement relationship. In the finite element implementation of this model we must, therefore, compute the relative displacement at an integration point to provide u^{cr}. We do this in Abaqus by multiplying the strain by a characteristic length associated with the integration point. The characteristic crack length is based on the element geometry and formulation: it is a typical length of a line across an element for a first-order element; it is half of the same typical length for a second-order element. For beams and trusses it is a characteristic length along the element axis. For membranes and shells it is a characteristic length in the reference surface. For axisymmetric elements it is a characteristic length in the r–z plane only. For cohesive elements it is equal to the constitutive

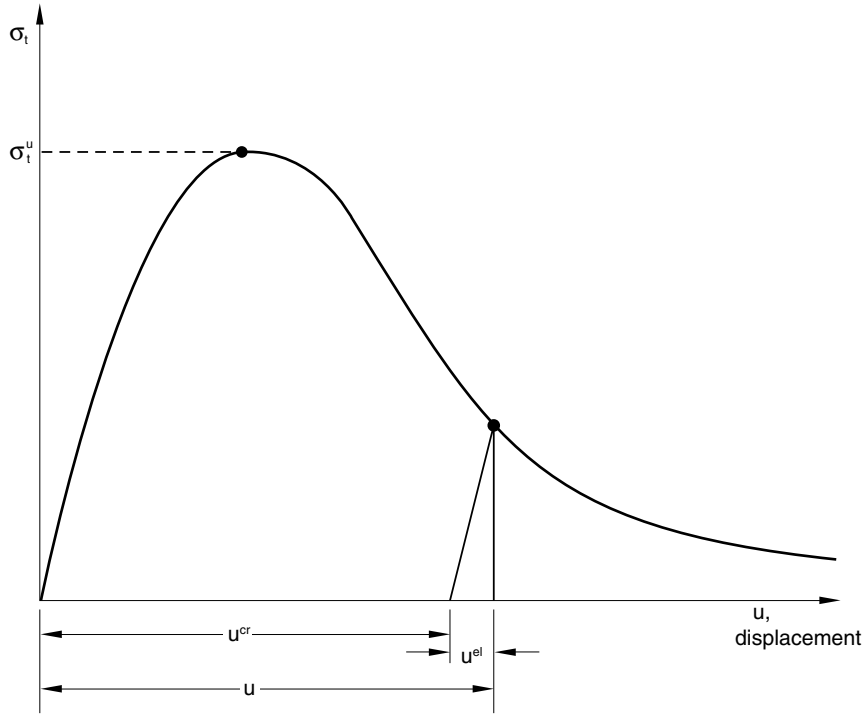


Figure 4.5.1–4 Cracking behavior based on fracture energy.

thickness. This definition of the characteristic length is used because we do not necessarily know in which direction the concrete will crack and so cannot choose the length measure in any particular direction. Thus, if there are elements in the model that have large aspect ratios, the model will likely provide different results if it is loaded in different directions and cracking occurs in such elements. This effect should be considered by the user in defining values for the material properties.

In reinforced concrete the σ–u^{cr} relationship must also represent the action of the bond between the concrete and the rebar as the concrete cracks. We assume this is accommodated by increasing the value of G_f based on comparisons with experiments on reinforced material.

We first describe the crack detection plasticity model and then discuss the damaged elasticity.

Strain rate decomposition

We decompose the elastic strain rate of Equation 4.5.1–1 as

$$d\boldsymbol{\varepsilon}^{el} = d\boldsymbol{\varepsilon}_d^{el} + d\boldsymbol{\varepsilon}_t^{pl}, \tag{4.5.1–10}$$

where $d\varepsilon^{el}$ is the total mechanical strain rate for the crack detection problem, $d\varepsilon_d^{el}$ is the elastic strain rate, and $d\varepsilon_t^{pl}$ is the plastic strain rate associated with the crack detection surface.

Yield

The crack detection surface is the Coulomb line

$$f_t = \hat{q} - \left(3 - b_0 \frac{\sigma_t}{\sigma_t^u}\right) \hat{p} - \left(2 - \frac{b_0}{3} \frac{\sigma_t}{\sigma_t^u}\right) \sigma_t = 0, \tag{4.5.1–11}$$

where σ_t^u is the failure stress in uniaxial tension and b_0 is a constant that is defined from the value of the tensile failure stress, σ_I, in a state of biaxial stress when the other nonzero principal stress, σ_{II}, is at the uniaxial compression ultimate stress value, σ_c^u. $\sigma_t(\lambda_t)$ is a hardening parameter (σ_t is the equivalent uniaxial tensile stress). The hardening is measured by λ_t, with the $\sigma_t(\lambda_t)$ relationship defined from the user-specified tension stiffening data (see Figure 4.5.1–5).

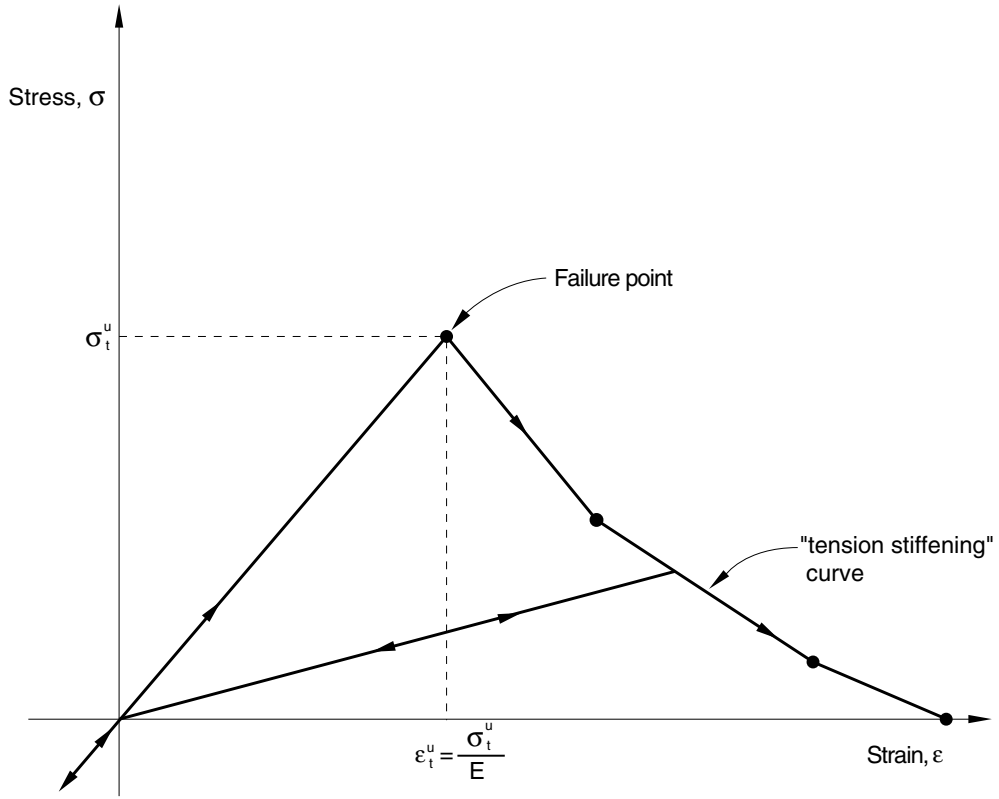


Figure 4.5.1–5 "Tension stiffening" model.

The stress measures $\hat{p}$ and $\hat{q}$ are defined in the same way as $\boldsymbol{p}$ and $\boldsymbol{q}$, except that all stress components $\sigma_{\alpha\beta}$ associated with open cracks (that is, if α or β is a crack direction in which the direct strain $\varepsilon^{el}_{\alpha\alpha} > 0$ or $\varepsilon^{el}_{\beta\beta} > 0$) are not included in these measures: they are invariants in subspaces of the stress space.

This surface has a simple mathematical form but matches plane stress data quite well. The hardening is introduced in the particular form shown in Equation 4.5.1–11 so that, as $\sigma_t \to 0$, the surface becomes $q - 3\,p = 0$, which in ($\boldsymbol{p}$–$\boldsymbol{q}$) space is the cone containing the principal axes of stress. This means that, as the tension stiffening is exhausted in a plane stress test, the stress point will drop back onto the nearest principal stress axis.

The value of b_0 is obtained as follows. The failure surface data include a definition of $\boldsymbol{f}$, a ratio that states that, in a plane stress test cracking would occur when one principal stress has the value $\sigma_I = -\sigma^u_c$ (σ^u_c is the magnitude of the ultimate stress in uniaxial compression) and the other nonzero principal stress has the value $\sigma_{II} = f\sigma^u_t$. Another value provided as part of the failure surface data is r^σ_t, which defines

$$\sigma^u_t = r^\sigma_t\,\sigma^u_c.$$

Cracking would, therefore, occur at the point with principal stresses $-\sigma^u_c$, $fr^\sigma_t\,\sigma^u_c$, and $\boldsymbol{0}$. For these values

$$p = \frac{1}{3}\,(1 - fr^\sigma_t)\,\sigma^u_c$$

and

$$q = \sigma^u_c\sqrt{1 + (fr^\sigma_t)^2} + fr^\sigma_t.$$

Therefore, with $\sigma_t = \sigma^u_t$,

$$f_t = \sigma^u_c\sqrt{1 + (fr^\sigma_t)^2} + fr^\sigma_t - \frac{1}{3}\,(3 - b_0)\,(1 - fr^\sigma_t)\,\sigma^u_c - \left(2 - \frac{b_0}{3}\right) r^\sigma_t\sigma^u_c = 0,$$

so

$$b_0 = 3\,\frac{1 + (2 - f)r^\sigma_t - \sqrt{1 + (fr^\sigma_t)^2} + fr^\sigma_t}{1 + r^\sigma_t(1 - f)}\ .$$

The crack detection surface is shown in Figure 4.5.1–2 and Figure 4.5.1–3.

Flow

The crack detection model uses the assumption of associated flow, so if $f_t = 0$ and $d\lambda_t > 0$,

$$d\boldsymbol{\varepsilon}^{pl}_t = d\lambda_t\,\frac{\partial f_t}{\partial \boldsymbol{\sigma}}; \tag{4.5.1–12}$$

otherwise, $d\boldsymbol{\varepsilon}^{pl}_t = 0$.

Hardening

The user defines the tension stiffening behavior by specifying the magnitude of the stress, σ_t, in a uniaxial tension test as a function of the inelastic strain. (When the fracture energy concept is used to define the postfailure behavior, "strain" is now defined as u^{cr}/c, where $\boldsymbol{c}$ is the characteristic length associated with the integration point.) The $\sigma_t(\lambda_t)$ relationship is defined as follows.

Using the definition of f_t, Equation 4.5.1–11, in the flow rule above we write

$$d\boldsymbol{\varepsilon}_t^{pl} = d\lambda_t \left(\frac{3}{2}\frac{\mathbf{S}}{q} + \left(1 - \frac{b_0}{3}\frac{\sigma_t}{\sigma_t^u}\right)\mathbf{I} \right).$$

In uniaxial tension $S_{11} = \frac{2}{3}\sigma_t$ and $q = \sigma_t$. Therefore, in uniaxial tension,

$$\left(d\varepsilon_t^{pl}\right)_{11} = d\lambda_t \left(2 - \frac{b_0}{3}\frac{\sigma_t}{\sigma_t^u}\right);$$

and, hence,

$$d\lambda_t = \left(d\varepsilon_t^{pl}\right)_{11} \Big/ \left(2 - \frac{b_0}{3}\frac{\sigma_t}{\sigma_t^u}\right) . \tag{4.5.1–13}$$

Upon integration Equation 4.5.1–13 gives λ_t from $\left(d\varepsilon_t^{pl}\right)_{11}$; and, therefore, the $\sigma_t(\lambda_t)$ relationship is obtained from the tension stiffening input data.

Damaged elasticity

Following crack detection we use damaged elasticity to model the failed material. The elasticity is written in the form

$$\boldsymbol{\sigma} = \mathbf{D} : \boldsymbol{\varepsilon}^{el}, \tag{4.5.1–14}$$

where $\mathbf{D}$ is the elastic stiffness matrix for the concrete.

Let α represent a cracked direction, with corresponding direct stress $\sigma_{\overline{\alpha\alpha}}$ and direct elastic strain $\varepsilon^{el}_{\overline{\alpha\alpha}}$. In these expressions and in the remainder of this section, no summation is implied by repeated indices with a bar over them. If the fracture energy concept is used, the strains are related to the user-specified stress/displacement definition for the tension stiffening behavior by $\varepsilon = u/c$, where $\boldsymbol{c}$ is the characteristic length associated with the integration point.

Then, in $\mathbf{D}$, $D_{\overline{\alpha\alpha}\beta\gamma}$ is the usual elasticity of the concrete if $\varepsilon_{\overline{\alpha\alpha}} \le 0$. If $\varepsilon^{\text{open}}_{\overline{\alpha\alpha}} > \varepsilon_{\overline{\alpha\alpha}} > 0$,

$$D_{\overline{\alpha\alpha\alpha\alpha}} = \sigma^{\text{open}}_{\overline{\alpha\alpha}} \Big/ \varepsilon^{\text{open}}_{\overline{\alpha\alpha}},$$

where $\sigma^{\text{open}}_{\overline{\alpha\alpha}}$ is the stress corresponding to $\varepsilon^{\text{open}}_{\overline{\alpha\alpha}}$ (as defined in the tension stiffening data), and

$$\varepsilon_{\overline{\alpha\alpha}}^{\text{open}} = \max_{\text{over history}} \left(\varepsilon_{\overline{\alpha\alpha}}^{el}\right).$$

If $\varepsilon_{\overline{\alpha\alpha}} = \varepsilon_{\overline{\alpha\alpha}}^{\text{open}}$,

$$D_{\overline{\alpha\alpha\alpha\alpha}} = d\sigma_{\overline{\alpha\alpha}}^{\text{open}} \Big/ d\varepsilon_{\overline{\alpha\alpha}}^{\text{open}},$$

which follows from the tension stiffening data.

We also assume no Poisson effect for open cracks: for $\varepsilon_{\overline{\alpha\alpha}}^{el} > 0$, $D_{\overline{\alpha\alpha}\beta\gamma} = 0$ for $\beta \neq \alpha, \gamma \neq \alpha$.

The shear terms in the elasticity associated with existing crack directions are

$$D_{\overline{\alpha\beta\alpha\beta}} = \hat{G}, \quad \beta \neq \alpha,$$

where $\hat{G} = \varrho^{\text{close}} G$ for $\varepsilon_{\overline{\alpha\alpha}} \leq 0$, and $\hat{G} = \varrho^{\text{open}} G$ for $\varepsilon_{\overline{\alpha\alpha}} > 0$. In these expressions G is the elastic shear modulus, ϱ^{close} is a constant defined by the user as part of the shear retention data (see Figure 4.5.1–6), and ϱ^{open} is a linear function of $\bar{\varepsilon}^{el}$, $\varrho^{\text{open}} = (1 - \bar{\varepsilon}_{\overline{\alpha\alpha}}^{el}/\varepsilon^{\max})$ and is also defined by the user in the shear retention data. Here $\bar{\varepsilon}^{el} = \langle \varepsilon_{\overline{\alpha\alpha}}^{el} \rangle + \langle \varepsilon_{\overline{\beta\beta}}^{el} \rangle$, where $\langle$ and $\rangle$ are Macauley brackets, defining

$$\langle f \rangle = \begin{cases} f & \text{if } f \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

for any function f.

Cracking

As soon as the crack detection surface (f_t) has been activated, we assume that cracking has occurred. The crack direction, $\mathbf{n}_\alpha$, is taken to be the direction of that part of the maximum principal plastic strain increment conjugate to the crack detection surface, $\Delta\boldsymbol{\varepsilon}_t^{pl}$, that is orthogonal to the directions of any existing cracks at the same point. This crack orientation is stored for subsequent calculations, which are done for convenience in a local coordinate system oriented so that one of the coordinate directions is the crack direction, $\mathbf{n}_\alpha$. Cracking is irrecoverable in the sense that, once a crack has occurred at a point, it remains throughout the rest of the calculation. Following crack detection, the crack affects the calculations by damaging the elasticity, as defined above. In addition, if the elastic strain across a crack is tensile, the invariants used in the crack detection surface are defined in the stress sub-space in which all stress components associated with the open crack direction are neglected, as described in the section above on yield. This implies that no more than three cracks can occur at any point (two in a plane stress case, one in a uniaxial stress case).

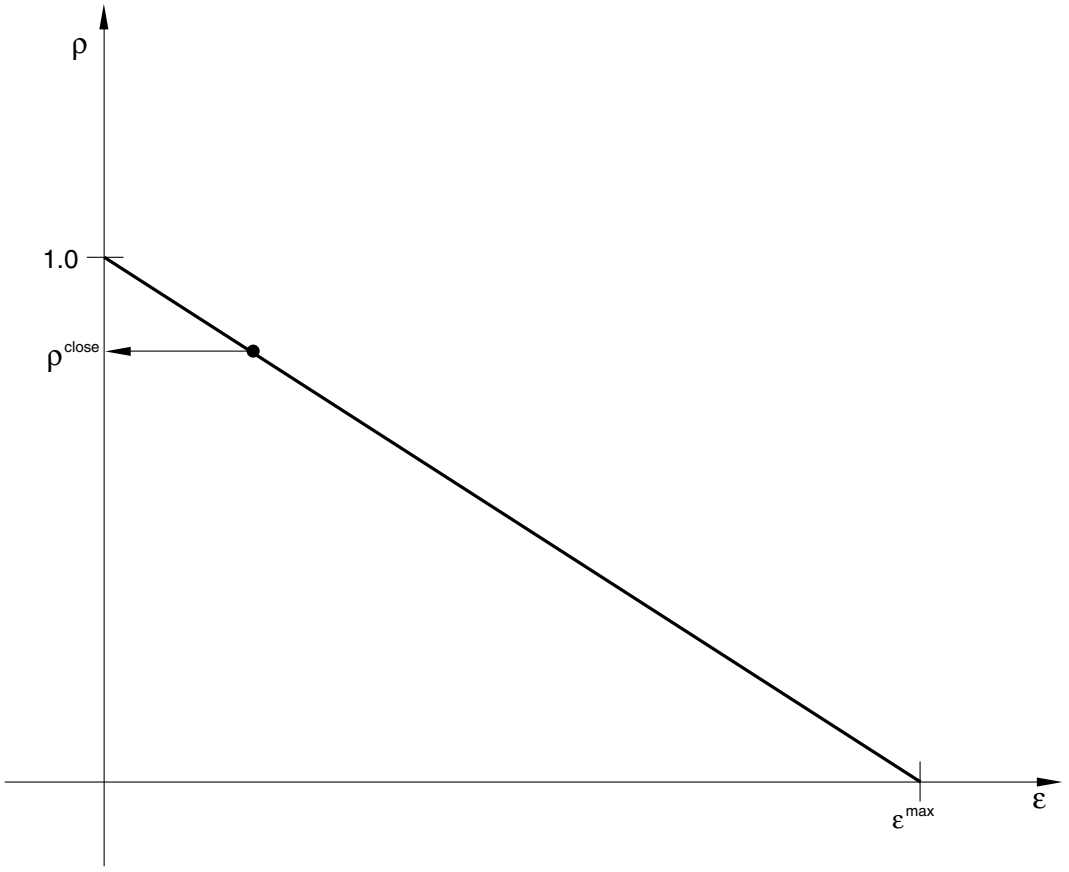


Figure 4.5.1–6 Shear retention.

Integration of the model

The model is integrated using the backward Euler method generally used with the plasticity models in Abaqus. A material Jacobian consistent with this integration operator is used for the equilibrium iterations.

Reference

- "Concrete smeared cracking," Section 23.6.1 of the Abaqus Analysis User's Guide

4.5.2 DAMAGED PLASTICITY MODEL FOR CONCRETE AND OTHER QUASI-BRITTLE MATERIALS

Products: Abaqus/Standard Abaqus/Explicit

This section describes the concrete damaged plasticity model provided in Abaqus for the analysis of concrete and other quasi-brittle materials. The material library in Abaqus also includes other constitutive models for concrete based on the smeared crack approach. These are the smeared crack model in Abaqus/Standard, described in "An inelastic constitutive model for concrete," Section 4.5.1, and the brittle cracking model in Abaqus/Explicit, described in "A cracking model for concrete and other brittle materials," Section 4.5.3.

The concrete damaged plasticity model is primarily intended to provide a general capability for the analysis of concrete structures under cyclic and/or dynamic loading. The model is also suitable for the analysis of other quasi-brittle materials, such as rock, mortar and ceramics; but it is the behavior of concrete that is used in the remainder of this section to motivate different aspects of the constitutive theory. Under low confining pressures, concrete behaves in a brittle manner; the main failure mechanisms are cracking in tension and crushing in compression. The brittle behavior of concrete disappears when the confining pressure is sufficiently large to prevent crack propagation. In these circumstances failure is driven by the consolidation and collapse of the concrete microporous microstructure, leading to a macroscopic response that resembles that of a ductile material with work hardening.

Modeling the behavior of concrete under large hydrostatic pressures is out of the scope of the plastic-damage model considered here. The constitutive theory in this section aims to capture the effects of irreversible damage associated with the failure mechanisms that occur in concrete and other quasi-brittle materials under fairly low confining pressures (less than four or five times the ultimate compressive stress in uniaxial compression loading). These effects manifest themselves in the following macroscopic properties:

- different yield strengths in tension and compression, with the initial yield stress in compression being a factor of 10 or more higher than the initial yield stress in tension;
- softening behavior in tension as opposed to initial hardening followed by softening in compression;
- different degradation of the elastic stiffness in tension and compression;
- stiffness recovery effects during cyclic loading; and
- rate sensitivity, especially an increase in the peak strength with strain rate.

The plastic-damage model in Abaqus is based on the models proposed by Lubliner et al. (1989) and by Lee and Fenves (1998). The model is described in the remainder of this section. An overview of the main ingredients of the model is given first, followed by a more detailed discussion of the different aspects of the constitutive model.

Overview

The main ingredients of the inviscid concrete damaged plasticity model are summarized below.

Strain rate decomposition

An additive strain rate decomposition is assumed for the rate-independent model:

$$\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}}^{el} + \dot{\boldsymbol{\varepsilon}}^{pl},$$

where $\dot{\boldsymbol{\varepsilon}}$ is the total strain rate, $\dot{\boldsymbol{\varepsilon}}^{el}$ is the elastic part of the strain rate, and $\dot{\boldsymbol{\varepsilon}}^{pl}$ is the plastic part of the strain rate.

Stress-strain relations

The stress-strain relations are governed by scalar damaged elasticity:

$$\boldsymbol{\sigma} = (1-d)\mathbf{D}_0^{el} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{pl}) = \mathbf{D}^{el} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{pl}),$$

where $\mathbf{D}_0^{el}$ is the initial (undamaged) elastic stiffness of the material; $\mathbf{D}^{el} = (1-d)\mathbf{D}_0^{el}$ is the degraded elastic stiffness; and d is the scalar stiffness degradation variable, which can take values in the range from zero (undamaged material) to one (fully damaged material). Damage associated with the failure mechanisms of the concrete (cracking and crushing) therefore results in a reduction in the elastic stiffness. Within the context of the scalar-damage theory, the stiffness degradation is isotropic and characterized by a single degradation variable, d. Following the usual notions of continuum damage mechanics, the effective stress is defined as

$$\bar{\boldsymbol{\sigma}} \overset{\text{def}}{=} \mathbf{D}_0^{el} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{pl}).$$

The Cauchy stress is related to the effective stress through the scalar degradation relation:

$$\boldsymbol{\sigma} = (1-d)\bar{\boldsymbol{\sigma}}.$$

For any given cross-section of the material, the factor $(1-d)$ represents the ratio of the effective load-carrying area (i.e., the overall area minus the damaged area) to the overall section area. In the absence of damage, $d = 0$, the effective stress $\bar{\boldsymbol{\sigma}}$ is equivalent to the Cauchy stress, $\boldsymbol{\sigma}$. When damage occurs, however, the effective stress is more representative than the Cauchy stress because it is the effective stress area that is resisting the external loads. It is, therefore, convenient to formulate the plasticity problem in terms of the effective stress. As discussed later, the evolution of the degradation variable is governed by a set of hardening variables, $\tilde{\boldsymbol{\varepsilon}}^{pl}$, and the effective stress; that is, $d = d(\bar{\boldsymbol{\sigma}}, \tilde{\boldsymbol{\varepsilon}}^{pl})$.

Hardening variables

Damaged states in tension and compression are characterized independently by two hardening variables, $\tilde{\varepsilon}_t^{pl}$ and $\tilde{\varepsilon}_c^{pl}$, which are referred to as equivalent plastic strains in tension and compression, respectively. The evolution of the hardening variables is given by an expression of the form

$$\tilde{\boldsymbol{\varepsilon}}^{pl} = \begin{bmatrix} \tilde{\varepsilon}_t^{pl} \\ \tilde{\varepsilon}_c^{pl} \end{bmatrix}; \quad \dot{\tilde{\boldsymbol{\varepsilon}}}^{pl} = \mathbf{h}(\bar{\boldsymbol{\sigma}}, \tilde{\boldsymbol{\varepsilon}}^{pl}) \cdot \dot{\boldsymbol{\varepsilon}}^{pl},$$

as described later in this section.

Microcracking and crushing in the concrete are represented by increasing values of the hardening variables. These variables control the evolution of the yield surface and the degradation of the elastic stiffness. They are also intimately related to the dissipated fracture energy required to generate microcracks.

Yield function

The yield function, $F(\bar{\boldsymbol{\sigma}}, \tilde{\varepsilon}^{pl})$, represents a surface in effective stress space, which determines the states of failure or damage. For the inviscid plastic-damage model

$$F(\bar{\boldsymbol{\sigma}}, \tilde{\varepsilon}^{pl}) \leq 0.$$

The specific form of the yield function is described later in this section.

Flow rule

Plastic flow is governed by a flow potential G according to the flow rule:

$$\gamma \dot{\boldsymbol{\varepsilon}}^{pl} = \dot{\lambda} \frac{\partial G(\bar{\boldsymbol{\sigma}})}{\partial \bar{\boldsymbol{\sigma}}},$$

where $\dot{\lambda}$ is the nonnegative plastic multiplier. The plastic potential is defined in the effective stress space. The specific form of the flow potential for the concrete damaged plasticity model is discussed later in this section. The model uses nonassociated plasticity, therefore requiring the solution of nonsymmetric equations.

Summary

In summary, the elastic-plastic response of the concrete damaged plasticity model is described in terms of the effective stress and the hardening variables:

$$\begin{aligned} \bar{\boldsymbol{\sigma}} &= \mathbf{D}_0^{el} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{pl}) \in \{\bar{\boldsymbol{\sigma}} | F(\bar{\boldsymbol{\sigma}}, \tilde{\varepsilon}^{pl}) \leq 0\}, \\ \dot{\tilde{\varepsilon}}^{pl} &= \mathbf{h}(\bar{\boldsymbol{\sigma}}, \tilde{\varepsilon}^{pl}) \cdot \dot{\boldsymbol{\varepsilon}}^{pl}, \\ \dot{\boldsymbol{\varepsilon}}^{pl} &= \dot{\lambda} \frac{\partial G(\bar{\boldsymbol{\sigma}})}{\partial \bar{\boldsymbol{\sigma}}}, \end{aligned} \tag{4.5.2–1}$$

where $\dot{\lambda}$ and F obey the Kuhn-Tucker conditions: $\dot{\lambda} F = 0;\ \dot{\lambda} \geq 0;\ F \leq 0$. The Cauchy stress is calculated in terms of the stiffness degradation variable, $d(\bar{\boldsymbol{\sigma}}, \tilde{\varepsilon}^{pl})$, and the effective stress as

$$\boldsymbol{\sigma} = (1 - d)\bar{\boldsymbol{\sigma}}. \tag{4.5.2–2}$$

The constitutive relations for the elastic-plastic response, Equation 4.5.2–1, are decoupled from the stiffness degradation response, Equation 4.5.2–2, which makes the model attractive for an effective numerical implementation. The inviscid model summarized here can be extended easily to account for

viscoplastic effects through the use of a viscoplastic regularization by permitting stresses to be outside the yield surface.

Damage and stiffness degradation

The evolution equations of the hardening variables $\tilde{\varepsilon}_t^{pl}$ and $\tilde{\varepsilon}_c^{pl}$ are conveniently formulated by considering uniaxial loading conditions first and then extended to multiaxial conditions.

Uniaxial conditions

It is assumed that the uniaxial stress-strain curves can be converted into stress versus plastic strain curves of the form

$$\begin{aligned} \sigma_t &= \sigma_t(\tilde{\varepsilon}_t^{pl}, \dot{\tilde{\varepsilon}}_t^{pl}, \theta, f_i), \\ \sigma_c &= \sigma_c(\tilde{\varepsilon}_c^{pl}, \dot{\tilde{\varepsilon}}_c^{pl}, \theta, f_i), \end{aligned} \tag{4.5.2–3}$$

where the subscripts *t* and *c* refer to tension and compression, respectively; $\dot{\tilde{\varepsilon}}_t^{pl}$ and $\dot{\tilde{\varepsilon}}_c^{pl}$ are the equivalent plastic strain rates, $\tilde{\varepsilon}_t^{pl} = \int_0^t \dot{\tilde{\varepsilon}}_t^{pl} dt$ and $\tilde{\varepsilon}_c^{pl} = \int_0^t \dot{\tilde{\varepsilon}}_c^{pl} dt$ are the equivalent plastic strains, θ is the temperature, and $f_i, (i = 1, 2, ...)$ are other predefined field variables.

Under uniaxial loading conditions the effective plastic strain rates are given as

$$\begin{aligned} \dot{\tilde{\varepsilon}}_t^{pl} &= \dot{\varepsilon}_{11}^{pl}, \text{ in uniaxial tension and} \\ \dot{\tilde{\varepsilon}}_c^{pl} &= -\dot{\varepsilon}_{11}^{pl}, \text{ in uniaxial compression.} \end{aligned} \tag{4.5.2–4}$$

In the remainder of this section we adopt the convention that σ_c is a positive quantity representing the magnitude of the uniaxial compression stress; that is, $\sigma_c = -\sigma_{11}$.

As shown in Figure 4.5.2–1, when the concrete specimen is unloaded from any point on the strain softening branch of the stress-strain curves, the unloading response is observed to be weakened: the elastic stiffness of the material appears to be damaged (or degraded). The degradation of the elastic stiffness is significantly different between tension and compression tests; in either case, the effect is more pronounced as the plastic strain increases. The degraded response of concrete is characterized by two independent uniaxial damage variables, d_t and d_c, which are assumed to be functions of the plastic strains, temperature, and field variables:

$$\begin{aligned} d_t &= d_t(\tilde{\varepsilon}_t^{pl}, \theta, f_i), \quad (0 \le d_t \le 1), \\ d_c &= d_c(\tilde{\varepsilon}_c^{pl}, \theta, f_i), \quad (0 \le d_c \le 1). \end{aligned} \tag{4.5.2–5}$$

The uniaxial degradation variables are increasing functions of the equivalent plastic strains. They can take values ranging from zero, for the undamaged material, to one, for the fully damaged material.

If E_0 is the initial (undamaged) elastic stiffness of the material, the stress-strain relations under uniaxial tension and compression loading are, respectively:

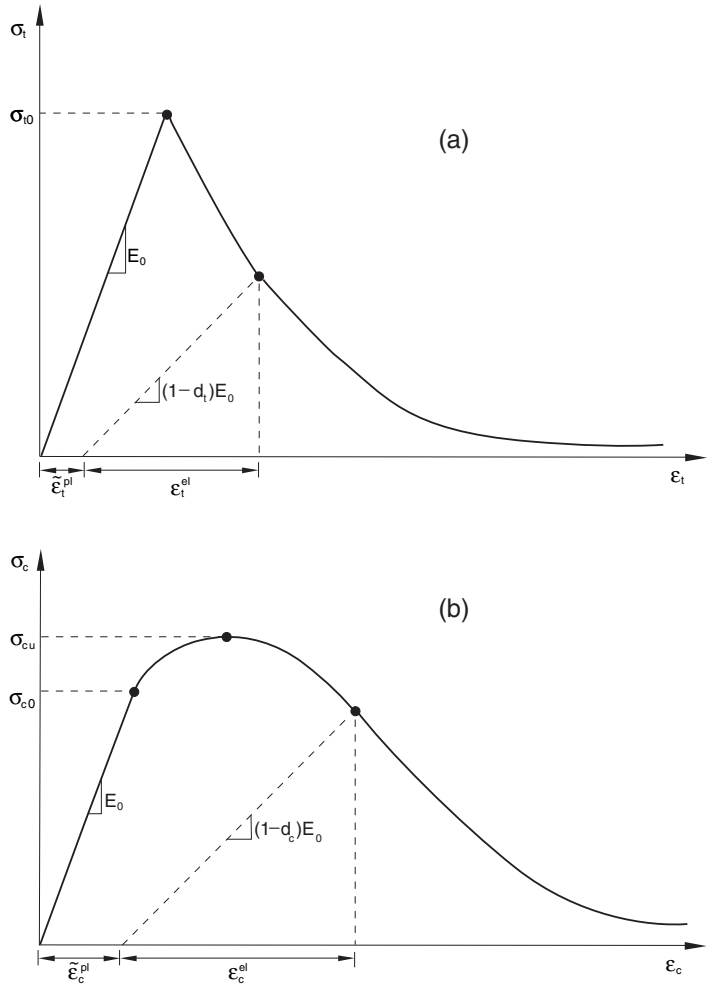


Figure 4.5.2–1 Response of concrete to uniaxial loading in tension (a) and compression (b).

$$\sigma_t = (1 - d_t)E_0(\varepsilon_t - \tilde{\varepsilon}_t^{pl}),$$
$$\sigma_c = (1 - d_c)E_0(\varepsilon_c - \tilde{\varepsilon}_c^{pl}).$$

Under uniaxial loading cracks propagate in a direction transverse to the stress direction. The nucleation and propagation of cracks, therefore, causes a reduction of the available load-carrying area, which in turn leads to an increase in the effective stress. The effect is less pronounced under compressive loading since cracks run parallel to the loading direction; however, after a significant amount of crushing, the effective

load-carrying area is also significantly reduced. The effective uniaxial cohesion stresses, $\bar{\sigma}_t$ and $\bar{\sigma}_c$, are given as

$$\bar{\sigma}_t = \frac{\sigma_t}{(1-d_t)} = E_0(\varepsilon_t - \tilde{\varepsilon}_t^{pl}),$$
$$\bar{\sigma}_c = \frac{\sigma_c}{(1-d_c)} = E_0(\varepsilon_c - \tilde{\varepsilon}_c^{pl}).$$

The effective uniaxial cohesion stresses determine the size of the yield (or failure) surface.

Uniaxial cyclic conditions

Under uniaxial cyclic loading conditions the degradation mechanisms are quite complex, involving the opening and closing of previously formed micro-cracks, as well as their interaction. Experimentally, it is observed that there is some recovery of the elastic stiffness as the load changes sign during a uniaxial cyclic test. The stiffness recovery effect, also known as the "unilateral effect," is an important aspect of the concrete behavior under cyclic loading. The effect is usually more pronounced as the load changes from tension to compression, causing tensile cracks to close, which results in the recovery of the compressive stiffness.

The concrete damaged plasticity model assumes that the reduction of the elastic modulus is given in terms of a scalar degradation variable, d, as

$$E = (1-d)E_0,$$

where E_0 is the initial (undamaged) modulus of the material.

This expression holds both in the tensile ($\sigma_{11} > 0$) and compressive ($\sigma_{11} < 0$) sides of the cycle. The stiffness reduction variable, d, is a function of the stress state and the uniaxial damage variables, d_t and d_c. For the uniaxial cyclic conditions, Abaqus assumes that

$$(1-d) = (1-s_t d_c)(1-s_c d_t), \quad 0 \le s_t,\ s_c \le 1, \tag{4.5.2–6}$$

where s_t and s_c are functions of the stress state that are introduced to represent stiffness recovery effects associated with stress reversals. They are defined according to

$$s_t = 1 - w_t r^*(\bar{\sigma}_{11}); \quad 0 \le w_t \le 1,$$
$$s_c = 1 - w_c(1 - r^*(\bar{\sigma}_{11})); \quad 0 \le w_c \le 1,$$

where

$$r^*(\bar{\sigma}_{11}) = H(\bar{\sigma}_{11}) = \begin{cases} 1 & \text{if} \quad \bar{\sigma}_{11} > 0 \\ 0 & \text{if} \quad \bar{\sigma}_{11} < 0 \end{cases}.$$

The weight factors w_t and w_c, which are assumed to be material properties, control the recovery of the tensile and compressive stiffness upon load reversal. To illustrate this, consider the example in

Figure 4.5.2–2, where the load changes from tension to compression. Assume that there was no previous compressive damage (crushing) in the material; that is, $\tilde{\varepsilon}_c^{pl} = 0$ and $d_c = 0$. Then

$$(1-d) = (1-s_c d_t) = (1-(1-w_c(1-r^*))d_t).$$

In tension ($\bar{\sigma}_{11} > 0$), $r^* = 1$; thus, $d = d_t$ as expected. In compression ($\bar{\sigma}_{11} < 0$), $r^* = 0$, and $d = (1-w_c)d_t$. If $w_c = 1$, then $d = 0$; therefore, the material fully recovers the compressive stiffness (which in this case is the initial undamaged stiffness, $E = E_0$). If, on the other hand, $w_c = 0$, then $d = d_t$ and there is no stiffness recovery. Intermediate values of w_c result in partial recovery of the stiffness.

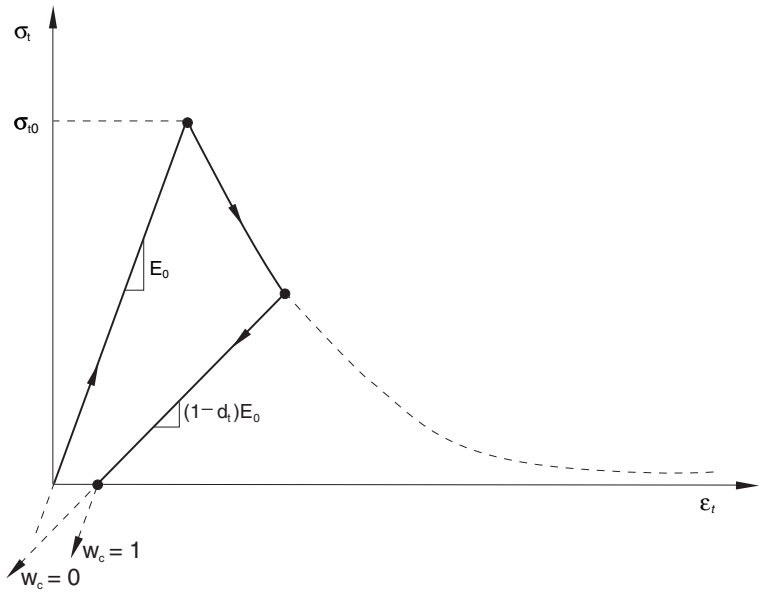


Figure 4.5.2–2 Illustration of the effect of the compression stiffness recovery parameter w_c.

The evolution equations of the equivalent plastic strains are also generalized to the uniaxial cyclic conditions as

$$\begin{aligned} \dot{\tilde{\varepsilon}}_t^{pl} &= r^* \dot{\varepsilon}_{11}^{pl}, \\ \dot{\tilde{\varepsilon}}_c^{pl} &= -(1-r^*)\dot{\varepsilon}_{11}^{pl}, \end{aligned} \tag{4.5.2–7}$$

which clearly reduces to Equation 4.5.2–4 during the tensile and compressive phases of the cycle.

Multiaxial conditions

The evolution equations for the hardening variables must be extended for the general multiaxial conditions. Based on Lee and Fenves (1998) we assume that the equivalent plastic strain rates are evaluated according to the expressions

$$\begin{aligned} \dot{\tilde{\varepsilon}}_t^{pl} &\overset{\text{def}}{=} r(\hat{\bar{\boldsymbol{\sigma}}})\dot{\hat{\varepsilon}}_{\max}^{pl}, \\ \dot{\tilde{\varepsilon}}_c^{pl} &\overset{\text{def}}{=} -(1-r(\hat{\bar{\boldsymbol{\sigma}}}))\dot{\hat{\varepsilon}}_{\min}^{pl}, \end{aligned} \tag{4.5.2–8}$$

where $\dot{\hat{\varepsilon}}_{\max}^{pl}$ and $\dot{\hat{\varepsilon}}_{\min}^{pl}$ are, respectively, the maximum and minimum eigenvalues of the plastic strain rate tensor $\dot{\boldsymbol{\varepsilon}}^{pl}$ and

$$r(\hat{\bar{\boldsymbol{\sigma}}}) \overset{\text{def}}{=} \frac{\sum_{i=1}^{3}\langle\hat{\bar{\sigma}}_i\rangle}{\sum_{i=1}^{3}|\hat{\bar{\sigma}}_i|}; \quad 0 \le r(\hat{\bar{\boldsymbol{\sigma}}}) \le 1$$

is a stress weight factor that is equal to one if all principal stresses $\hat{\bar{\sigma}}_i, (i = 1, 2, 3)$, are positive and equal to zero if they are negative. The Macauley bracket $\langle\cdot\rangle$ is defined by $\langle x\rangle = \frac{1}{2}(|x|+x)$. In uniaxial loading conditions Equation 4.5.2–8 reduces to the uniaxial definitions Equation 4.5.2–4 and Equation 4.5.2–7, since $\dot{\hat{\varepsilon}}_{\max}^{pl} = \dot{\varepsilon}_{11}^{pl}$ in tension, and $\dot{\hat{\varepsilon}}_{\min}^{pl} = \dot{\varepsilon}_{11}^{pl}$ in compression.

If the eigenvalues of the plastic strain rate tensor ($\dot{\hat{\varepsilon}}_i,\ i = 1, 2, 3$) are ordered such that $\dot{\hat{\varepsilon}}_{\max}^{pl} = \dot{\hat{\varepsilon}}_1 \ge \dot{\hat{\varepsilon}}_2 \ge \dot{\hat{\varepsilon}}_3 = \dot{\hat{\varepsilon}}_{\min}^{pl}$, the evolution equation for general multiaxial stress conditions can be expressed in the following matrix form:

$$\dot{\tilde{\boldsymbol{\varepsilon}}}^{pl} = \begin{bmatrix} \dot{\tilde{\varepsilon}}_t^{pl} \\ \dot{\tilde{\varepsilon}}_c^{pl} \end{bmatrix} = \hat{\mathbf{h}}(\hat{\bar{\boldsymbol{\sigma}}}, \tilde{\boldsymbol{\varepsilon}}^{pl}) \cdot \dot{\hat{\boldsymbol{\varepsilon}}}^{pl},$$

where

$$\hat{\mathbf{h}}(\hat{\bar{\boldsymbol{\sigma}}}, \tilde{\boldsymbol{\varepsilon}}^{pl}) = \begin{bmatrix} r(\hat{\bar{\boldsymbol{\sigma}}}) & 0 & 0 \\ 0 & 0 & -(1-r(\hat{\bar{\boldsymbol{\sigma}}})) \end{bmatrix},$$

and

$$\dot{\hat{\boldsymbol{\varepsilon}}}^{pl} = \begin{bmatrix} \dot{\hat{\varepsilon}}_1 \\ \dot{\hat{\varepsilon}}_2 \\ \dot{\hat{\varepsilon}}_3 \end{bmatrix}.$$

Elastic stiffness degradation

The plastic-damage concrete model assumes that the elastic stiffness degradation is isotropic and characterized by a single scalar variable, d:

$$\mathbf{D}^{el} = (1-d)\mathbf{D}_0^{el}; \quad 0 \le d \le 1. \tag{4.5.2–9}$$

The definition of the scalar degradation variable $\boldsymbol{d}$ must be consistent with the uniaxial monotonic responses (d_t and d_c), and it should also should capture the complexity associated with the degradation mechanisms under cyclic loading. For the general multiaxial stress conditions Abaqus assumes that

$$(1-d) = (1-s_t d_c)(1-s_c d_t), \quad 0 \le s_t,\ s_c \le 1, \tag{4.5.2–10}$$

similar to the uniaxial cyclic case, only that s_t and s_c are now given in terms of the function $r(\hat{\bar{\boldsymbol{\sigma}}})$ as

$$\begin{aligned} s_t &= 1 - w_t r(\hat{\bar{\boldsymbol{\sigma}}}); \quad 0 \le w_t \le 1, \\ s_c &= 1 - w_c(1 - r(\hat{\bar{\boldsymbol{\sigma}}})); \quad 0 \le w_c \le 1. \end{aligned}$$

It can be easily verified that Equation 4.5.2–10 for the scalar degradation variable is consistent with the uniaxial response.

The experimental observation in most quasi-brittle materials, including concrete, is that the compressive stiffness is recovered upon crack closure as the load changes from tension to compression. On the other hand, the tensile stiffness is not recovered as the load changes from compression to tension once crushing micro-cracks have developed. This behavior, which corresponds to $w_t = 0$ and $w_c = 1$, is the default used by Abaqus. Figure 4.5.2–3 illustrates a uniaxial load cycle assuming the default behavior.

Yield condition

The plastic-damage concrete model uses a yield condition based on the yield function proposed by Lubliner et al. (1989) and incorporates the modifications proposed by Lee and Fenves (1998) to account for different evolution of strength under tension and compression. In terms of effective stresses the yield function takes the form

$$F(\bar{\boldsymbol{\sigma}}, \tilde{\boldsymbol{\varepsilon}}^{pl}) = \frac{1}{1-\alpha}\left(\bar{q} - 3\alpha\bar{p} + \beta(\tilde{\boldsymbol{\varepsilon}}^{pl})\langle\hat{\bar{\sigma}}_{\max}\rangle - \gamma\langle-\hat{\bar{\sigma}}_{\max}\rangle\right) - \bar{\sigma}_c(\tilde{\varepsilon}_c^{pl}) \le 0, \tag{4.5.2–11}$$

where α and γ are dimensionless material constants;

$$\bar{p} = -\frac{1}{3}\bar{\boldsymbol{\sigma}} : \mathbf{I}$$

is the effective hydrostatic pressure;

$$\bar{q} = \sqrt{\frac{3}{2}\bar{\mathbf{S}} : \bar{\mathbf{S}}}$$

is the Mises equivalent effective stress;

$$\bar{\mathbf{S}} = \bar{p}\mathbf{I} + \bar{\boldsymbol{\sigma}},$$

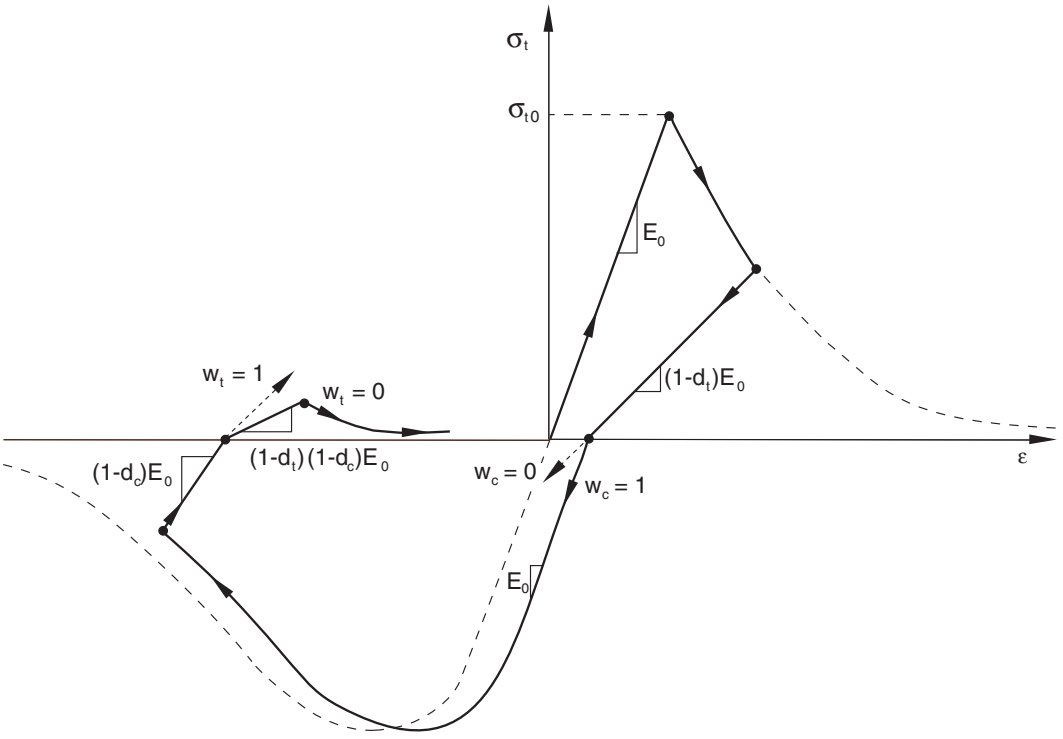


Figure 4.5.2–3 Uniaxial load cycle (tension-compression-tension) assuming default values for the stiffness recovery factors: $w_t = 0$ and $w_c = 1$.

is the deviatoric part of the effective stress tensor $\bar{\boldsymbol{\sigma}}$; and $\hat{\bar{\sigma}}_{\max}$ is the algebraically maximum eigenvalue of $\bar{\boldsymbol{\sigma}}$. The function $\beta(\tilde{\varepsilon}^{pl})$ is given as

$$\beta(\tilde{\varepsilon}^{pl}) = \frac{\bar{\sigma}_c(\tilde{\varepsilon}_c^{pl})}{\bar{\sigma}_t(\tilde{\varepsilon}_t^{pl})}(1-\alpha) - (1+\alpha),$$

where $\bar{\sigma}_t$ and $\bar{\sigma}_c$ are the effective tensile and compressive cohesion stresses, respectively.

In biaxial compression, with $\hat{\bar{\sigma}}_{\max} = 0$, Equation 4.5.2–11 reduces to the well-known Drucker-Prager yield condition. The coefficient α can be determined from the initial equibiaxial and uniaxial compressive yield stress, σ_{b0} and σ_{c0}, as

$$\alpha = \frac{\sigma_{b0} - \sigma_{c0}}{2\sigma_{b0} - \sigma_{c0}}.$$

Typical experimental values of the ratio σ_{b0}/σ_{c0} for concrete are in the range from 1.10 to 1.16, yielding values of α between 0.08 and 0.12 (Lubliner et al., 1989).

The coefficient γ enters the yield function only for stress states of triaxial compression, when $\hat{\bar{\sigma}}_{\max} < 0$. This coefficient can be determined by comparing the yield conditions along the tensile and compressive meridians. By definition, the *tensile meridian* (TM) is the locus of stress states satisfying the condition $\hat{\bar{\sigma}}_{\max} = \hat{\bar{\sigma}}_1 > \hat{\bar{\sigma}}_2 = \hat{\bar{\sigma}}_3$ and the *compressive meridian* (CM) is the locus of stress states such that $\hat{\bar{\sigma}}_{\max} = \hat{\bar{\sigma}}_1 = \hat{\bar{\sigma}}_2 > \hat{\bar{\sigma}}_3$, where $\hat{\bar{\sigma}}_1$, $\hat{\bar{\sigma}}_2$, and $\hat{\bar{\sigma}}_3$ are the eigenvalues of the effective stress tensor. It can be easily shown that $(\hat{\bar{\sigma}}_{\max})_{\mathrm{TM}} = \frac{2}{3}\bar{q} - \bar{p}$ and $(\hat{\bar{\sigma}}_{\max})_{\mathrm{CM}} = \frac{1}{3}\bar{q} - \bar{p}$, along the tensile and compressive meridians, respectively. With $\hat{\bar{\sigma}}_{\max} < 0$ the corresponding yield conditions are

$$\left(\frac{2}{3}\gamma + 1\right)\bar{q} - (\gamma + 3\alpha)\bar{p} = (1-\alpha)\bar{\sigma}_c, \quad \text{(TM)}$$

$$\left(\frac{1}{3}\gamma + 1\right)\bar{q} - (\gamma + 3\alpha)\bar{p} = (1-\alpha)\bar{\sigma}_c. \quad \text{(CM)}$$

Let $K_c = \bar{q}_{(\mathrm{TM})}/\bar{q}_{(\mathrm{CM})}$ for any given value of the hydrostatic pressure $\bar{p}$ with $\hat{\bar{\sigma}}_{\max} < 0$; then

$$K_c = \frac{\gamma + 3}{2\gamma + 3}.$$

The fact that K_c is constant does not seem to be contradicted by experimental evidence (Lubliner et al., 1989). The coefficient γ is, therefore, evaluated as

$$\gamma = \frac{3(1 - K_c)}{2K_c - 1}.$$

A value of $K_c = \frac{2}{3}$, which is typical for concrete, gives $\gamma = 3$.

If $\hat{\bar{\sigma}}_{\max} > 0$, the yield conditions along the tensile and compressive meridians reduce to

$$\left(\frac{2}{3}\beta + 1\right)\bar{q} - (\beta + 3\alpha)\bar{p} = (1-\alpha)\bar{\sigma}_c, \quad \text{(TM)}$$

$$\left(\frac{1}{3}\beta + 1\right)\bar{q} - (\beta + 3\alpha)\bar{p} = (1-\alpha)\bar{\sigma}_c. \quad \text{(CM)}$$

Let $K_t = \bar{q}_{(\mathrm{TM})}/\bar{q}_{(\mathrm{CM})}$ for any given value of the hydrostatic pressure $\bar{p}$ with $\hat{\bar{\sigma}}_{\max} > 0$; then

$$K_t = \frac{\beta + 3}{2\beta + 3}.$$

Typical yield surfaces are shown in Figure 4.5.2–4 in the deviatoric plane and in Figure 4.5.2–5 for plane-stress conditions.

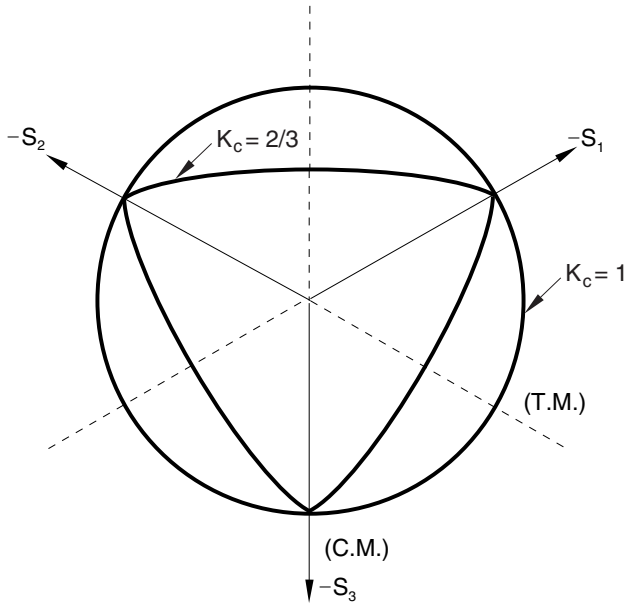


Figure 4.5.2–4 Yield surfaces in the deviatoric plane, corresponding to different values of K_c.

Flow rule

The plastic-damage model assumes nonassociated potential flow,

$$\dot{\boldsymbol{\varepsilon}}^{pl} = \dot{\lambda}\frac{\partial G(\bar{\boldsymbol{\sigma}})}{\partial \bar{\boldsymbol{\sigma}}}.$$

The flow potential G chosen for this model is the Drucker-Prager hyperbolic function:

$$G = \sqrt{(\epsilon\sigma_{t0}\tan\psi)^2 + \bar{q}^2} - \bar{p}\tan\psi, \tag{4.5.2–12}$$

where ψ is the dilation angle measured in the p–q plane at high confining pressure; σ_{t0} is the uniaxial tensile stress at failure; and ϵ is a parameter, referred to as the eccentricity, that defines the rate at which the function approaches the asymptote (the flow potential tends to a straight line as the eccentricity tends to zero). This flow potential, which is continuous and smooth, ensures that the flow direction is defined uniquely. The function asymptotically approaches the linear Drucker-Prager flow potential at high confining pressure stress and intersects the hydrostatic pressure axis at 90°. See "Models for granular or polymer behavior," Section 4.4.2, for further discussion of this potential.

Because plastic flow is nonassociated, the use of the plastic-damage concrete model requires the solution of nonsymmetric equations.

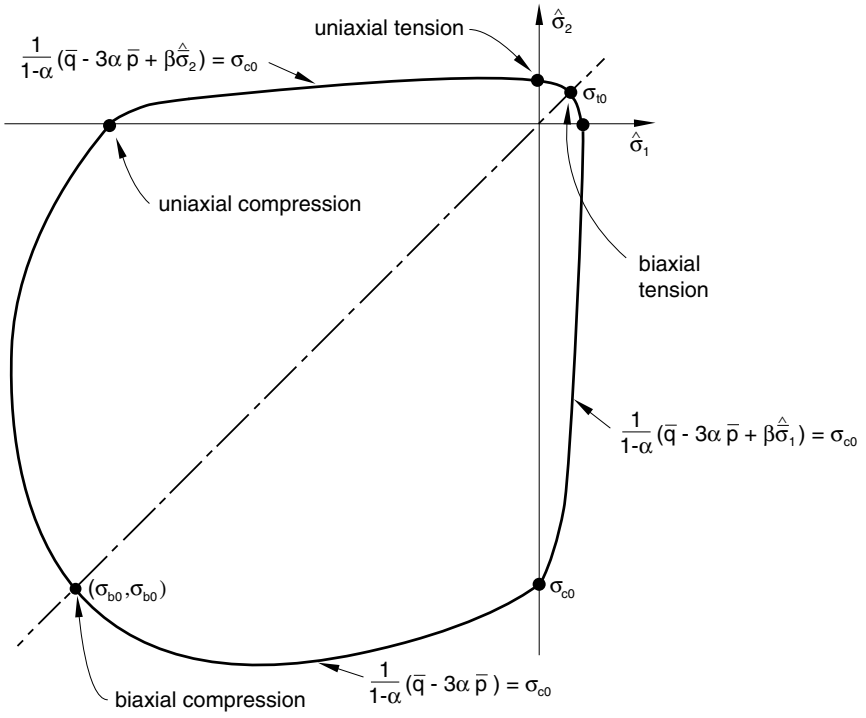


Figure 4.5.2–5 Yield surface in plane stress.

Viscoplastic regularization

Material models exhibiting softening behavior and stiffness degradation often lead to severe convergence difficulties in implicit analysis programs. Some of these convergence difficulties can be overcome by using a viscoplastic regularization of the constitutive equations. The concrete damaged plasticity model can be regularized using viscoplasticity, therefore permitting stresses to be outside of the yield surface. We use a generalization of the Duvaut-Lions regularization, according to which the viscoplastic strain rate tensor, $\dot{\varepsilon}_v^{pl}$, is defined as

$$\dot{\varepsilon}_v^{pl} = \frac{1}{\mu}(\varepsilon^{pl} - \varepsilon_v^{pl}). \tag{4.5.2–13}$$

Here μ is the viscosity parameter representing the relaxation time of the viscoplastic system and ε^{pl} is the plastic strain evaluated in the inviscid backbone model.

Similarly, a viscous stiffness degradation variable, d_v, for the viscoplastic system is defined as

$$\dot{d}_v = \frac{1}{\mu}(d - d_v), \tag{4.5.2–14}$$

where d is the degradation variable evaluated in the inviscid backbone model. The stress-strain relation of the viscoplastic model is given as

$$\boldsymbol{\sigma} = (1 - d_v)\mathbf{D}_0^{el} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_v^{pl}). \tag{4.5.2–15}$$

The solution of the viscoplastic system relaxes to that of the inviscid case as $t/\mu \to \infty$, where t represents time. Using the viscoplastic regularization with a small value for the viscosity parameter (small compared to the characteristic time increment) usually helps improve the rate of convergence of the model in the softening regime, without compromising results.

Integration of the model

The model is integrated using the backward Euler method generally used with the plasticity models in Abaqus. A material Jacobian consistent with this integration operator is used for the equilibrium iterations.

4.5.3 A CRACKING MODEL FOR CONCRETE AND OTHER BRITTLE MATERIALS

Product: Abaqus/Explicit

This section describes the cracking constitutive model provided in Abaqus/Explicit for concrete and other brittle materials. The material library in Abaqus also includes a constitutive model for concrete based on theories of scalar plastic damage, described in "Damaged plasticity model for concrete and other quasi-brittle materials," Section 4.5.2, which is available in Abaqus/Standard and Abaqus/Explicit. In Abaqus/Standard plain concrete can also be analyzed with the smeared crack concrete model described in "An inelastic constitutive model for concrete," Section 4.5.1. Although this brittle cracking model can also be useful for other materials, such as ceramics and brittle rocks, it is primarily intended to model plain concrete. Therefore, in the remainder of this section, the physical behavior of concrete is used to motivate the different aspects of the constitutive model.

Reinforced concrete modeling in Abaqus is accomplished by combining standard elements, using this plain concrete cracking model, with "rebar elements"—rods, defined singly or embedded in oriented surfaces, that use a one-dimensional strain theory and that can be used to model the reinforcing itself. The rebar elements are superposed on the mesh of plain concrete elements and are used with standard metal plasticity models that describe the behavior of the rebar material. This modeling approach allows the concrete behavior to be considered independently of the rebar, so this section discusses the plain concrete cracking model only. Effects associated with the rebar/concrete interface, such as bond slip and dowel action, cannot be considered in this approach except by modifying some aspects of the plain concrete behavior to mimic them (such as the use of "tension stiffening" to simulate load transfer across cracks through the rebar).

It is generally accepted that concrete exhibits two primary modes of behavior: a brittle mode in which microcracks coalesce to form discrete macrocracks representing regions of highly localized deformation, and a ductile mode where microcracks develop more or less uniformly throughout the material, leading to nonlocalized deformation. The brittle behavior is associated with cleavage, shear and mixed mode fracture mechanisms that are observed under tension and tension-compression states of stress. It almost always involves softening of the material. The ductile behavior is associated with distributed microcracking mechanisms that are primarily observed under compression states of stress. It almost always involves hardening of the material, although subsequent softening is possible at low confining pressures. The cracking model described here models only the brittle aspects of concrete behavior. Although this is a major simplification, there are many applications where only the brittle behavior of the concrete is significant; and, therefore, the assumption that the material is linear elastic in compression is justified in those cases.

Smeared cracking assumption

A smeared model is chosen to represent the discontinuous macrocrack brittle behavior. In this approach we do not track individual "macro" cracks: rather, the presence of cracks enters into the calculations by the way the cracks affect the stress and material stiffness associated with each material calculation point.

Here, for simplicity, the term "crack" is used to mean a direction in which cracking has been detected at the material calculation point in question. The closest physical concept is that there exists a continuum of microcracks at the point, oriented as determined by the model. The anisotropy introduced by cracking

is included in the model since it is assumed to be important in the simulations for which the model is intended.

Some objections have been raised against smeared crack models. The principal concern is that this modeling approach inherently introduces mesh sensitivity in the solutions, in the sense that the finite element results do not converge to a unique result. For example, since cracking is associated with strain softening, mesh refinement will lead to narrower crack bands. Many researchers have addressed this concern, and the general consensus is that Hillerborg's (1976) approach—based on brittle fracture concepts—is adequate to deal with this issue for practical purposes. A length scale, typically in the form of a "characteristic" length, is introduced to "regularize" the smeared continuum models and attenuate the sensitivity of the results to mesh density. This aspect of the model is discussed in detail later.

Crack direction assumptions

Various researchers have proposed three basic crack direction models (Rots and Blaauwendraad, 1989): fixed, orthogonal cracks; the rotating crack model; and fixed, multidirectional (nonorthogonal) cracks. In the fixed, orthogonal crack model the direction normal to the first crack is aligned with the direction of maximum tensile principal stress at the time of crack initiation. The model has memory of this crack direction, and subsequent cracks at the point under consideration can only form in directions orthogonal to the first crack. In the rotating crack concept only a single crack can form at any point (aligned with the direction of maximum tensile principal stress). Thus, the single crack direction rotates with the direction of the principal stress axes. This model has no memory of crack direction. Finally, the multidirectional crack model allows the formation of any number of cracks at a point as the direction of the principal stress axes changes with loading. In practice, some limitation is imposed on the number of cracks allowed to form at a point. The model has memory of all crack directions.

The multidirectional crack model is the least popular, mainly because the criterion used to decide when subsequent cracks form (to limit the number of cracks at a point) is somewhat arbitrary: the concept of a "threshold angle" is introduced to prevent new cracks from forming at angles less than this threshold value to existing cracks. The fixed orthogonal and rotating crack models have both been used extensively, even though objections can be raised against both. In the rotating crack model the concept of crack closing and reopening is not well-defined because the orientation of the crack can vary continuously. The fixed orthogonal crack model has been criticized mainly because the traditional treatment of "shear retention" employed in the model tends to make the response of the model too stiff. This problem can be resolved by formulating the shear retention in a way that ensures that the shear stresses tend to zero as deformation on the crack interfaces takes place (this is done in the Abaqus model, as described later). Finally, although the fixed orthogonal crack model has the orthogonality limitation, it is considered superior to the rotating crack model in cases where the effect of multiple cracks is important (the rotating crack model is restricted to a single crack at any point).

The fixed orthogonal cracks model is used in Abaqus so that the maximum number of cracks at a material point is limited by the number of direct stress components present at that material point of the finite element model (for example, a maximum of three cracks in three-dimensional, axisymmetric, and plane strain problems or a maximum of two cracks in plane stress problems). Once cracks exist at a point, the component forms of all vector and tensor valued quantities are rotated so that they lie in the local system defined by the crack orientation vectors (the normals to the crack faces). The model ensures

that these crack face normal vectors are orthogonal so that this local system is rectangular Cartesian. Crack closing and reopening can take place along the directions of the crack surface normals. The model neglects any permanent strain associated with cracking; that is, we assume that the cracks can close completely when the stress across them becomes compressive.

Elastic-cracking model for concrete

The main ingredients of the model are a strain rate decomposition into elastic (concrete) and cracking strain rates, elasticity, a set of cracking conditions, and a cracking relation (the evolution law for the cracking behavior). The main advantage of the strain decomposition is that it allows the eventual addition of other effects, such as plasticity and creep, in a consistent manner. The elastic-cracking strain decomposition also allows the separate identification of a cracking strain that represents the state of a crack; this contrasts with the classical smeared cracking models where a single strain quantity is used to represent the state of a cracked solid in a homogenized form leading to a modified (damaged) elasticity formulation.

Strain rate decomposition

We begin with a strain rate decomposition,

$$d\boldsymbol{\varepsilon} = d\boldsymbol{\varepsilon}^{el} + d\boldsymbol{\varepsilon}^{ck}, \tag{4.5.3–1}$$

where $d\boldsymbol{\varepsilon}$ is the total mechanical strain rate, $d\boldsymbol{\varepsilon}^{el}$ is the elastic strain rate representing the uncracked concrete (the continuum between the cracks), and $d\boldsymbol{\varepsilon}^{ck}$ is the cracking strain rate associated with any existing cracks.

Crack direction transformations

The strains in Equation 4.5.3–1 are referred to the global Cartesian coordinate system and can be written in vector form (in a three-dimensional setting) as

$$\boldsymbol{\varepsilon} = [\varepsilon_{11} \quad \varepsilon_{22} \quad \varepsilon_{33} \quad \gamma_{12} \quad \gamma_{13} \quad \gamma_{23}]^T.$$

For incorporating the cracking relations it is convenient to define a local Cartesian coordinate system n, t, s that is aligned with the crack directions. In the local system, shown in Figure 4.5.3–1, the strains are

$$\mathbf{e} = [e_{nn} \quad e_{tt} \quad e_{ss} \quad g_{nt} \quad g_{ns} \quad g_{ts}]^T.$$

The transformation between global and local strains is written in matrix form as

$$\boldsymbol{\varepsilon} = \mathbf{T}\,\mathbf{e}, \tag{4.5.3–2}$$

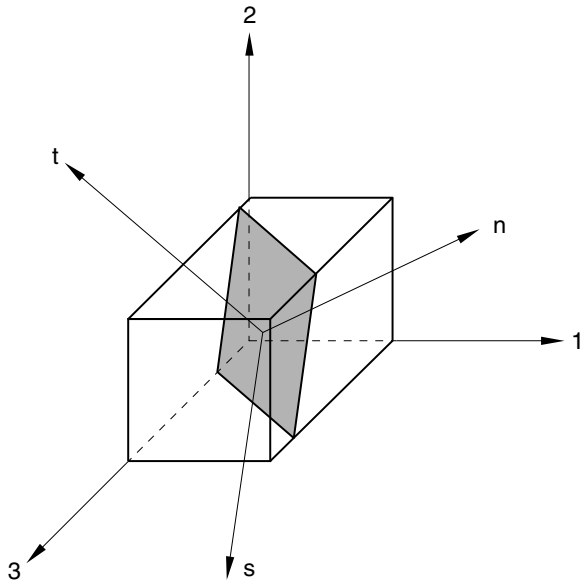


Figure 4.5.3–1 Global and local cracking coordinate systems.

where $\mathbf{T}$ is a transformation matrix constructed from the direction cosines of the local cracking coordinate system. $\mathbf{T}$ is constant in our fixed crack model.

The conjugate stress quantities can be written in the global coordinate system as

$$\boldsymbol{\sigma} = [\sigma_{11} \quad \sigma_{22} \quad \sigma_{33} \quad \sigma_{12} \quad \sigma_{13} \quad \sigma_{23}]^T$$

and in the local cracking system as

$$\mathbf{t} = [t_{nn} \quad t_{tt} \quad t_{ss} \quad t_{nt} \quad t_{ns} \quad t_{ts}]^T.$$

The transformation between local and global stresses is then

$$\mathbf{t} = \mathbf{T}^T \boldsymbol{\sigma}. \tag{4.5.3–3}$$

Elasticity

The intact continuum between the cracks is modeled with isotropic, linear elasticity. The orthotropic nature of the cracked material is introduced in the cracking component of the model. As stated earlier, the approach of decomposing the strains into elastic, intact concrete, strains, and cracking strains has the advantage that this smeared model can be generalized to include other effects such as plasticity and creep (although such generalizations are not yet included in Abaqus/Explicit).

Crack detection

A simple Rankine criterion is used to detect crack initiation. This states that a crack forms when the maximum principal tensile stress exceeds the tensile strength of the brittle material. The Rankine crack detection surface is shown in Figure 4.5.3–2 in the deviatoric plane, in Figure 4.5.3–3 in the meridional plane, and in Figure 4.5.3–4 in plane stress. Although crack detection is based purely on Mode I fracture considerations, ensuing cracked behavior includes both Mode I (tension softening) and Mode II (shear softening/retention) behavior, as described later.

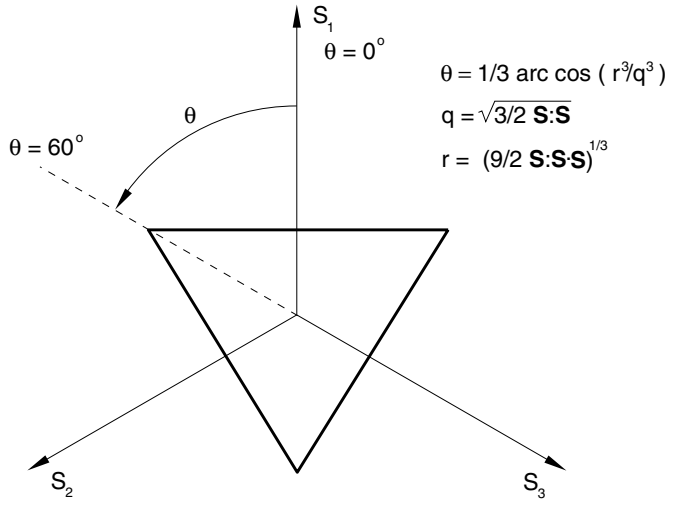


Figure 4.5.3–2 Rankine criterion in the deviatoric plane.

As soon as the Rankine criterion for crack formation has been met, we assume that a first crack has formed. The crack surface is taken to be normal to the direction of the maximum tensile principal stress. Subsequent cracks can form with crack surface normals in the direction of maximum principal tensile stress that is orthogonal to the directions of any existing crack surface normals at the same point.

The crack orientations are stored for subsequent calculations, which are done for convenience in a local coordinate system oriented in the crack directions. Cracking is irrecoverable in the sense that, once a crack has occurred at a point, it remains throughout the rest of the calculation. However, a crack may subsequently close and reopen.

Cracking conditions

We introduce a consistency condition for cracking (analogous to the yield condition in classical plasticity) written in the crack direction coordinate system in the form of the tensor

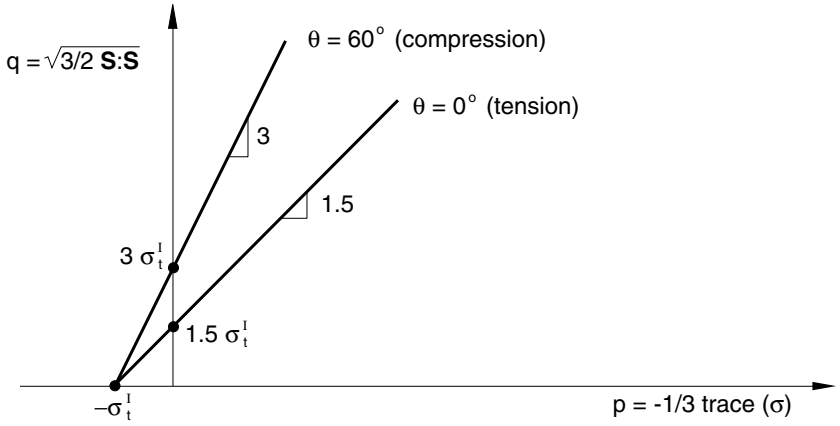


Figure 4.5.3–3 Rankine criterion in the meridional plane.

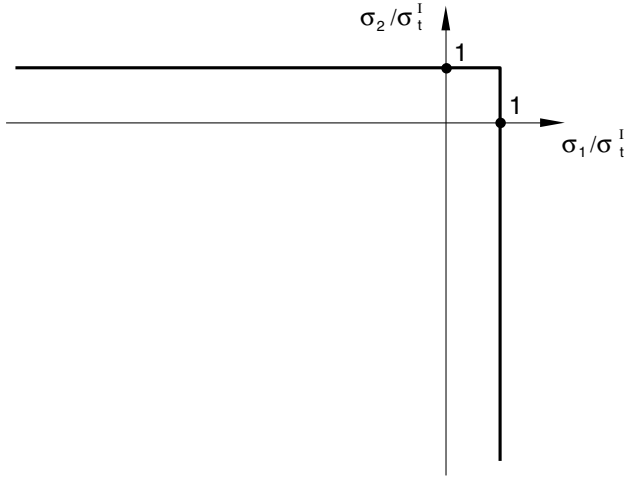


Figure 4.5.3–4 Rankine criterion in plane stress.

$$\mathbf{C} = \mathbf{C}(\mathbf{t}, \boldsymbol{\sigma}^{I,II}) = \mathbf{0}, \tag{4.5.3–4}$$

where

$$\mathbf{C} = [C_{nn} \quad C_{tt} \quad C_{ss} \quad C_{nt} \quad C_{ns} \quad C_{ts}]^T$$

and $\boldsymbol{\sigma}^{I,II}$ represents a tension softening model (Mode I fracture) in the case of the direct components of stress and a shear softening/retention model (Mode II fracture) in the case of the shear components of

stress. The matrices $\partial \mathbf{C}/\partial \mathbf{t}$ and $\partial \mathbf{C}/\partial \boldsymbol{\sigma}^{I,II}$ are assumed to be diagonal, implying the usual assumption that there is no coupling between cracks in the cracking conditions.

Each cracking condition is more complex than a classical yield condition in the sense that two cracking states are possible (an actively opening crack state and a closing/reopening crack state), contrasting with a single plastic state in classical plasticity. This can be illustrated by writing the cracking conditions for a particular crack normal direction $\boldsymbol{n}$ explicitly:

$$C_{nn} = C_{nn}(t_{nn}, \sigma_t^I) = t_{nn} - \sigma_t^I(e_{nn}^{ck}) = 0 \tag{4.5.3–5}$$

for an actively opening crack, where $\sigma_t^I(e_{nn}^{ck})$ is the tension softening evolution (defined by the user), and

$$C_{nn} = C_{nn}(t_{nn}, \sigma_c^I) = t_{nn} - \sigma_c^I(e_{nn}^{ck})\Big|_{e_{nn}^{\text{open}}} = 0 \tag{4.5.3–6}$$

for a closing/reopening crack, where $\sigma_c^I(e_{nn}^{ck})\Big|_{e_{nn}^{\text{open}}}$ is the crack closing/reopening evolution that depends on the maximum crack opening strain defined as

$$e_{nn}^{\text{open}} = \max_{\text{over history}} \left(e_{nn}^{ck}\right).$$

These conditions are illustrated in Figure 4.5.3–5 and represent the tension softening model adopted for the cracking behavior normal to crack surfaces. Similar conditions can be written for the other two possible crack normal directions, $\boldsymbol{s}$ and $\boldsymbol{t}$. It must be emphasized that, although the cracking condition of Equation 4.5.3–4 has been written for the most general case of all possible cracks existing, only the components of $\mathbf{C}$ that refer to existing cracks are considered in the computations with this model.

The cracking conditions for the shear components in the crack coordinate system are activated when the associated normal directions are cracked. We now present the shear cracking conditions by writing the conditions for shear component nt explicitly.

The crack opening dependent shear model (shear retention model) is written as

$$C_{nt} = C_{nt}(t_{nt}, \sigma_s^{II}) = t_{nt} - \sigma_s^{II}(g_{nt}^{ck}, e_{nn}^{ck}, e_{tt}^{ck}) = 0 \tag{4.5.3–7}$$

for shear loading or unloading of the crack, where $\sigma_s^{II}(g_{nt}^{ck}, e_{nn}^{ck}, e_{tt}^{ck})$ is the shear evolution that depends linearly on the shear strain and also depends on the crack opening strain (this dependency being defined by the user). Figure 4.5.3–6 illustrates the model. Although this model is inspired by the traditional shear retention models, it differs from those models in one important aspect: the shear stress tends to zero as the crack develops. This is discussed in more detail later.

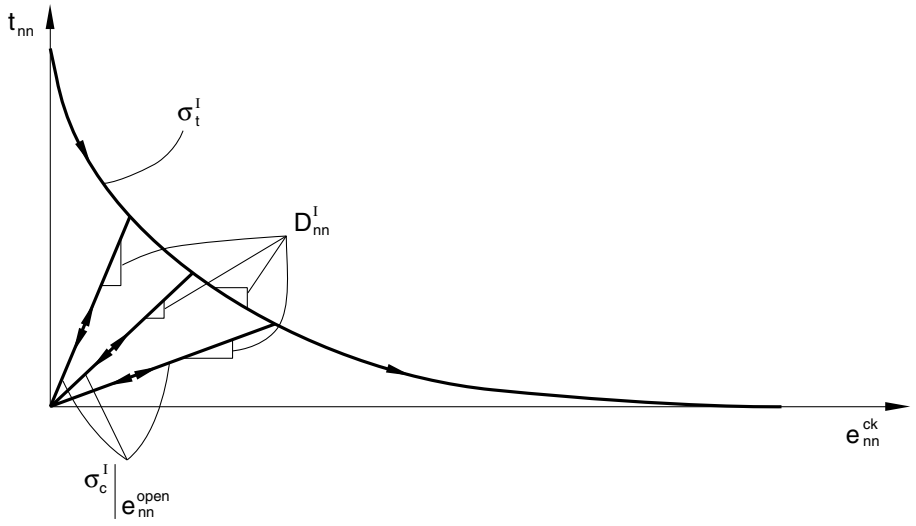


Figure 4.5.3–5 Cracking conditions for Mode I cracking.

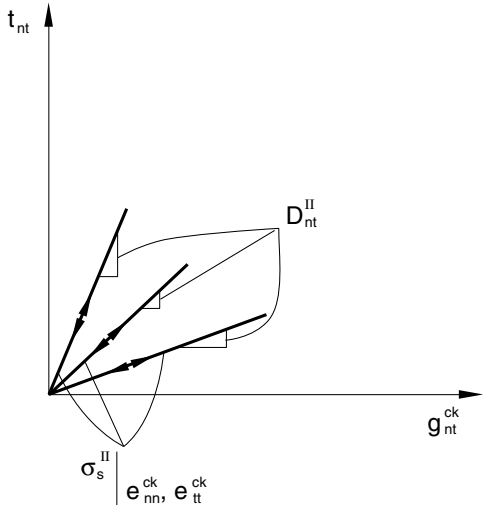


Figure 4.5.3–6 Cracking conditions for Mode II cracking (crack opening dependent model).

Cracking relation

The relation between the local stresses and the cracking strains at the crack interfaces is written in rate form as

$$d\mathbf{t} = \mathbf{D}^{ck}\, d\mathbf{e}^{ck}, \tag{4.5.3–8}$$

where $\mathbf{D}^{ck}$ is a diagonal cracking matrix that depends on the state of the existing cracks. The definition of these diagonal components ($D^I_{nn}, D^I_{tt}, D^I_{ss}, D^{II}_{nt}, D^{II}_{ns}, D^{II}_{ts}$) is given in Figure 4.5.3–5 and Figure 4.5.3–6.

Rate constitutive equations

Using the strain rate decomposition (Equation 4.5.3–3) and the elasticity relations, we can write the rate of stress as

$$d\boldsymbol{\sigma} = \mathbf{D}^{el}\left(d\boldsymbol{\varepsilon} - \mathbf{T}\, d\mathbf{e}^{ck}\right), \tag{4.5.3–9}$$

where $\mathbf{D}^{el}$ is the isotropic linear elasticity matrix.

Premultiplying Equation 4.5.3–9 by $\mathbf{T}^T$ and substituting Equation 4.5.3–5 and Equation 4.5.3–8 into the resulting left-hand side yields

$$d\mathbf{e}^{ck} = \left(\mathbf{D}^{ck} + \mathbf{T}^T\, \mathbf{D}^{el}\, \mathbf{T}\right)^{-1} \mathbf{T}^T\, \mathbf{D}^{el}\, d\boldsymbol{\varepsilon}. \tag{4.5.3–10}$$

Finally, substituting Equation 4.5.3–10 into Equation 4.5.3–9 results in the stress-strain rate equations:

$$d\boldsymbol{\sigma} = \left[\mathbf{D}^{el} - \mathbf{D}^{el}\, \mathbf{T}\left(\mathbf{D}^{ck} + \mathbf{T}^T\, \mathbf{D}^{el}\, \mathbf{T}\right)^{-1} \mathbf{T}^T\, \mathbf{D}^{el}\right] d\boldsymbol{\varepsilon} = \mathbf{D}^{ec}\, d\boldsymbol{\varepsilon}. \tag{4.5.3–11}$$

Tension softening models

The brittle fracture concept of Hilleborg (1976) forms the basis of the postcracked behavior in the direction normal to the crack surface (commonly referred to as tension softening). We assume that the fracture energy required to form a unit area of crack surface in Mode I, G^I_f, is a material property. This value can be calculated from measuring the tensile stress as a function of the crack opening displacement (Figure 4.5.3–7), as

$$G^I_f = \int \sigma^I_t\, du_n.$$

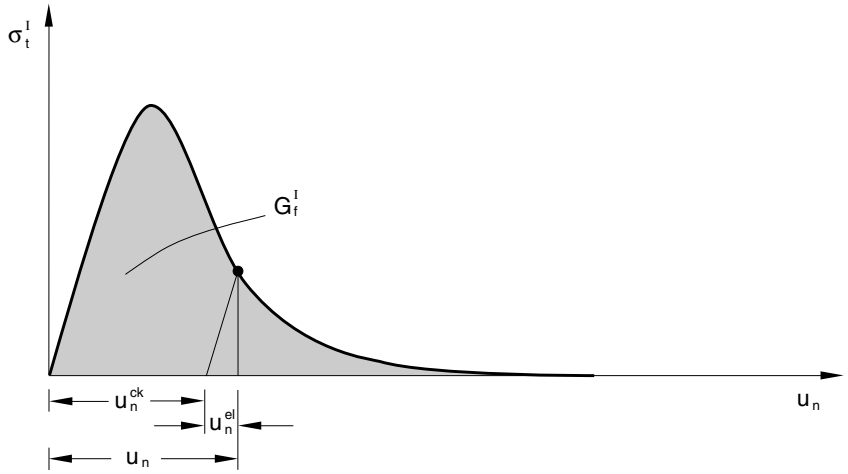


Figure 4.5.3–7 Mode I fracture energy based cracking behavior.

Typical values of G_f^I range from 40 N/m (0.22 lb/in) for a typical construction concrete (with a compressive strength of approximately 20 MPa, 2850 lb/in^2) to 120 N/m (0.67 lb/in) for a high strength concrete (with a compressive strength of approximately 40 MPa, 5700 lb/in^2).

The implication of assuming that G_f^I is a material property is that, when the elastic part of the displacement, u_n^{el}, is eliminated, the relationship between the stress and the remaining part of the displacement, $u_n^{ck} = u_n - u_n^{el}$, is fixed, regardless of the specimen size. For example, consider a specimen developing a single crack across its section as tensile displacement is applied to it: u_n^{ck} is the displacement across the crack and is not changed by using a longer or shorter specimen in the test (so long as the specimen is significantly longer than the width of the crack band, which will typically be of the order of the aggregate size). Thus, this important part of the cracked concrete's tensile behavior is defined in terms of a stress/displacement relationship.

In the finite element implementation of this model we must, therefore, compute the relative displacement at a material point to provide u_n^{ck}. We do this in Abaqus by multiplying the strain by a characteristic length associated with the material point (the cracking strain in local crack direction $\boldsymbol{n}$ is used as an example):

$$u_n^{ck} = e_{nn}^{ck}\, h,$$

where $\boldsymbol{h}$ is the characteristic length. This characteristic crack length is based on the element geometry and formulation: it is a typical length of a line across an element for a first-order element; it is half of the same typical length for a second-order element. For beams and trusses it is a characteristic length along the element axis. For membranes and shells it is a characteristic length in the reference surface. For axisymmetric elements it is a characteristic length in the $\boldsymbol{r}$–$\boldsymbol{z}$ plane only. For cohesive elements it is equal to the constitutive thickness. This definition of the characteristic length is used because we do not

necessarily know in which direction the concrete will crack; and, hence, we cannot choose the length measure *a priori* in any particular direction. These characteristic length estimates are appropriate only for well-shaped elements (elements that do not have large aspect ratios), which should be considered by the user in defining values for the material properties. Alternatively, this mesh dependency could be reduced by directly specifying the characteristic length as a function of element topology and material orientation in user subroutine **VUCHARLENGTH**, as described in "VUCHARLENGTH," Section 1.2.11 of the Abaqus User Subroutines Reference Guide.

For reinforced concrete, since Abaqus provides no direct modeling of the bond between rebar and concrete, the effect of this bond on the concrete cracks must be smeared into the plain concrete part of the model. This effect is generally accomplished by increasing the value of G_f^I based on comparisons with experiments on reinforced material. This increased ductility is commonly refered to as the "tension stiffening" effect.

In reinforced concrete applications the softening behavior of the concrete tends to have less influence on the overall response of the structure because of the stabilizing presence of the rebar. Therefore, it is often appropriate to define tension stiffening as a σ_t^I–e_{nn}^{ck} relationship directly. This option is also offered in Abaqus.

Cracked shear models

An important feature of the cracking model is that, whereas crack initiation is based on Mode I fracture only, postcracked behavior includes Mode II as well as Mode I. The Mode II shear behavior is described next.

The Mode II model is based on the common observation that the shear behavior is dependent on the amount of crack opening. Therefore, Abaqus offers a shear retention model in which the postcracked shear stiffness is dependent on crack opening. This model defines the total shear stress as a function of the total shear strain (shear direction nt is used as an example):

$$t_{nt} = D_{nt}^{II}(e_{nn}^{ck}, e_{tt}^{ck})\, g_{nt}^{ck}, \tag{4.5.3–12}$$

where $D_{nt}^{II}(e_{nn}^{ck}, e_{tt}^{ck})$ is a stiffness that depends on crack opening. D_{nt}^{II} can be expressed as

$$D_{nt}^{II} = \alpha(e_{nn}^{ck}, e_{tt}^{ck})\, G,$$

where G is the shear modulus of the uncracked concrete and $\alpha(e_{nn}^{ck}, e_{tt}^{ck})$ is a user-defined dependence of the form shown in Figure 4.5.3–8.

A commonly used mathematical form for this dependence when there is only one crack, associated with direction n, is the power law proposed by Rots and Blaauwendraad (1989):

$$\alpha(e_{nn}^{ck}) = \frac{\left(1 - \frac{e_{nn}^{ck}}{e_{max}^{ck}}\right)^p}{1 - \left(1 - \frac{e_{nn}^{ck}}{e_{max}^{ck}}\right)^p}, \tag{4.5.3–13}$$

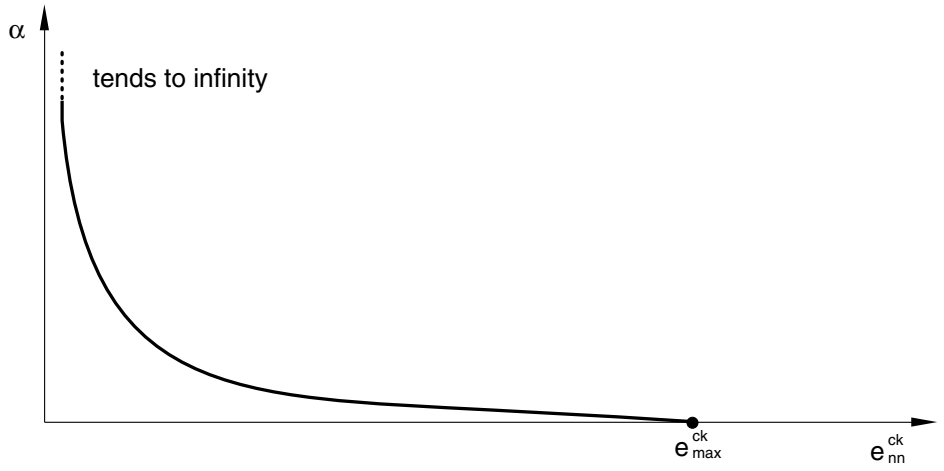


Figure 4.5.3–8 Shear retention factor dependence on crack opening.

where p and e^{ck}_{max} are material parameters. This form satisfies the requirements that $\alpha \to \infty$ as $e^{ck}_{nn} \to 0$ (corresponding to the state before crack initiation) and $\alpha \to 0$ as $e^{ck}_{nn} \to e^{ck}_{max}$ (corresponding to complete loss of aggregate interlock). Note that the bounds of α, as defined in our model using the elastic-cracking strain decomposition, are ∞ and zero. This contrasts with some of the traditional shear retention models where the intact concrete and cracking strains are not separated; the shear retention in these models is defined using a shear retention factor, ρ, which can have values between one and zero. The relationship between these two shear retention parameters is

$$\rho = \frac{\alpha}{(\alpha + 1)}. \tag{4.5.3–14}$$

The shear retention power law form given in Equation 4.5.3–13 can then be written in terms of ρ as

$$\rho(e^{ck}_{nn}) = \left(1 - \frac{e^{ck}_{nn}}{e^{ck}_{max}}\right)^{p}.$$

Since users are more accustomed to specifying shear retention factors in the traditional way (with values between one and zero), the Abaqus input requests ρ–e^{ck}_{nn} data. Using Equation 4.5.3–14, these data are then converted to α–e^{ck}_{nn} data for computation purposes.

When the shear component under consideration is associated with only one open crack direction (n or t), the crack opening dependence is obtained directly from Figure 4.5.3–8. However, when the shear direction is associated with two open crack directions (n and t), then

$$g^{ck}_{nt} = g^{ck,n}_{nt} + g^{ck,t}_{nt} = \frac{t_{nt}}{D^{II,n}_{nt}} + \frac{t_{nt}}{D^{II,t}_{nt}},$$

with

$$D_{nt}^{II,n} = \alpha(e_{nn}^{ck})\, G, \qquad D_{nt}^{II,t} = \alpha(e_{tt}^{ck})\, G,$$

and, therefore,

$$D_{nt}^{II} = \frac{t_{nt}}{g_{nt}^{ck}} = \frac{D_{nt}^{II,n}\, D_{nt}^{II,t}}{D_{nt}^{II,n} + D_{nt}^{II,t}}.$$

This total stress-strain shear retention model differs from the traditional shear retention models in which the stress-strain relations are written in incremental form (again, shear direction nt is used as an example):

$$\Delta t_{nt} = D_{nt}^{II}(e_{nn}^{ck}, e_{tt}^{ck})\, \Delta g_{nt}^{ck}, \tag{4.5.3–15}$$

where $D_{nt}^{II}(e_{nn}^{ck}, e_{tt}^{ck})$ is an incremental stiffness that depends on crack opening. The difference between the total model used in Abaqus (Equation 4.5.3–12) and the traditional incremental model (Equation 4.5.3–15) is best illustrated by considering the shear response of the two models in the case when a crack is simultaneously opening and shearing. This is shown in Figure 4.5.3–9 for the total model and in Figure 4.5.3–10 for the incremental model. It is apparent that, in the total model, the shear stress tends to zero as the crack opens and shears; whereas, in the incremental model the shear stress tends to a finite value. This may explain why overly stiff responses are usually obtained with the traditional shear retention models.

Reference

- "Cracking model for concrete," Section 23.6.2 of the Abaqus Analysis User's Guide

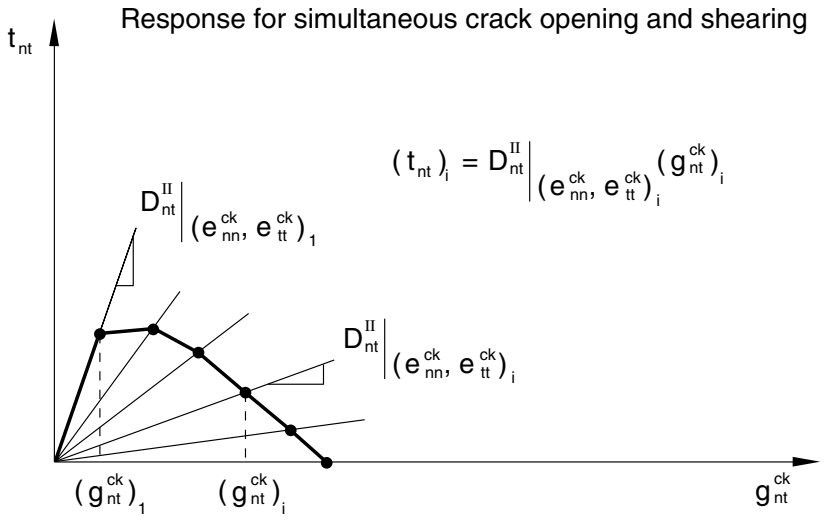


Figure 4.5.3–9 Abaqus crack opening–dependent shear retention (total) model.

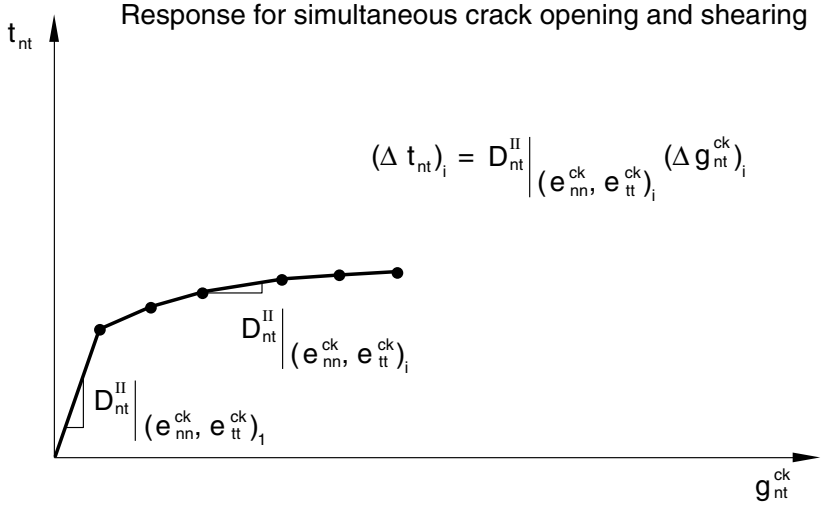


Figure 4.5.3–10 Traditional crack opening–dependent shear retention (incremental) model.

4.5.4 CONSTITUTIVE MODEL FOR JOINTED MATERIALS

Product: Abaqus/Standard

The jointed material model is intended to provide a simple, continuum model for materials containing a high density of parallel joint surfaces in different orientations. The spacing of the joints of a particular orientation is assumed to be sufficiently close compared to characteristic dimensions in the domain of the model that the joints can be smeared into a continuum of slip systems. An obvious application is the modeling of geotechnical problems where the medium of interest is composed of significantly faulted rock. In this context, models similar to the one described next have been proposed in the past; see, for example, the model formulated by Zienkiewicz and Pande (1977).

The model implemented in Abaqus/Standard provides for opening of the joints, or frictional sliding of the joints, in each of these systems (a "system" in this context is a joint orientation in a particular direction at a material calculation point). In addition to the joint systems, the model includes a bulk material failure mechanism. This is based on the Drucker-Prager failure criterion.

Joint system definitions

We consider a particular joint a oriented by the normal to the joint surface $\mathbf{n}_a$. We define $\mathbf{t}_{a\alpha}, \alpha = 1, 2$ as two unit, orthogonal vectors in the joint surface. The local stress components are the pressure stress across the joint

$$p_a \overset{\text{def}}{=} \mathbf{n}_a \cdot \boldsymbol{\sigma} \cdot \mathbf{n}_a,$$

and the shear stresses across the joint

$$\tau_{a\alpha} = \mathbf{n}_a \cdot \boldsymbol{\sigma} \cdot \mathbf{t}_{a\alpha},$$

where $\boldsymbol{\sigma}$ is the stress tensor. We define the shear stress magnitude as

$$\tau_a = \sqrt{\tau_{a\alpha}\tau_{a\alpha}}.$$

The local strain components are the normal strain across the joint

$$\varepsilon_{an} = \mathbf{n}_a \cdot \boldsymbol{\varepsilon} \cdot \mathbf{n}_a,$$

and the engineering shear strain in the α-direction in the joint surface

$$\gamma_{a\alpha} = \mathbf{n}_a \cdot \boldsymbol{\varepsilon} \cdot \mathbf{t}_{a\alpha} + \mathbf{t}_{a\alpha} \cdot \boldsymbol{\varepsilon} \cdot \mathbf{n}_a,$$

where $\boldsymbol{\varepsilon}$ is the strain tensor.

Strain rate decomposition

A linear strain rate decomposition is assumed, so that

$$d\boldsymbol{\varepsilon} = d\boldsymbol{\varepsilon}^{el} + d\boldsymbol{\varepsilon}^{pl}, \tag{4.5.4–1}$$

where $d\boldsymbol{\varepsilon}$ is the total strain rate, $d\boldsymbol{\varepsilon}^{el}$ is the elastic strain rate, and $d\boldsymbol{\varepsilon}^{pl}$ is the inelastic (plastic) strain rate. Supposing that several systems are active (we designate an active system by $\boldsymbol{i}$, where $i = b$ indicates the bulk material system and $i = a$ is a joint system $\boldsymbol{a}$), we write

$$d\boldsymbol{\varepsilon}^{pl} = \sum_i d\boldsymbol{\varepsilon}_i^{pl}. \tag{4.5.4–2}$$

Elasticity and joint opening/closing

When all joints at a point are closed, the elastic behavior of the material is assumed to be isotropic and linear. The material cannot be elastically incompressible (Poisson's ratio must be less than 0.5).

We use a stress-based joint opening criterion whereas joint closing is monitored based on strain. Joint system $\boldsymbol{a}$ opens when the estimated pressure stress across the joint (normal to the joint surface) is no longer positive:

$$p_a \leq 0.$$

In this case the material is assumed to have no elastic stiffness with respect to direct strain across the joint system. Open joints, thus, create anisotropic elastic response at a point. The joint system remains open as long as

$$\varepsilon_{an(ps)}^{el} \leq \varepsilon_{an}^{el},$$

where ε_{an}^{el} is the component of direct elastic strain across the joint and $\varepsilon_{an(ps)}^{el}$ is the component of direct elastic strain across the joint calculated in plane stress as

$$\varepsilon_{an(ps)}^{el} = -\frac{\nu}{E}(\sigma_{a1} + \sigma_{a2}),$$

where $\boldsymbol{E}$ is the Young's modulus of the material, ν is the Poisson's ratio, and

$$\sigma_{a\alpha} = \mathbf{t}_{a\alpha} \cdot \boldsymbol{\sigma} \cdot \mathbf{t}_{a\alpha},$$

are the direct stresses in the plane of the joint.

The shear response of open joints is governed by the shear retention parameter, f_{sr}, which represents the fraction of the elastic shear modulus retained when the joints are open (f_{sr}=0 means no shear stiffness

associated with open joints, while f_{sr}=1 corresponds to elastic shear stiffness in open joints; any value between these two extremes can be used).

Plastic behavior of joint systems

The failure surface for sliding on joint system a is defined by

$$f_a = \tau_a - p_a \tan\beta_a - d_a = 0, \tag{4.5.4–3}$$

where β_a is the friction angle for system a, and d_a is the cohesion for system a (see Figure 4.5.4–1).

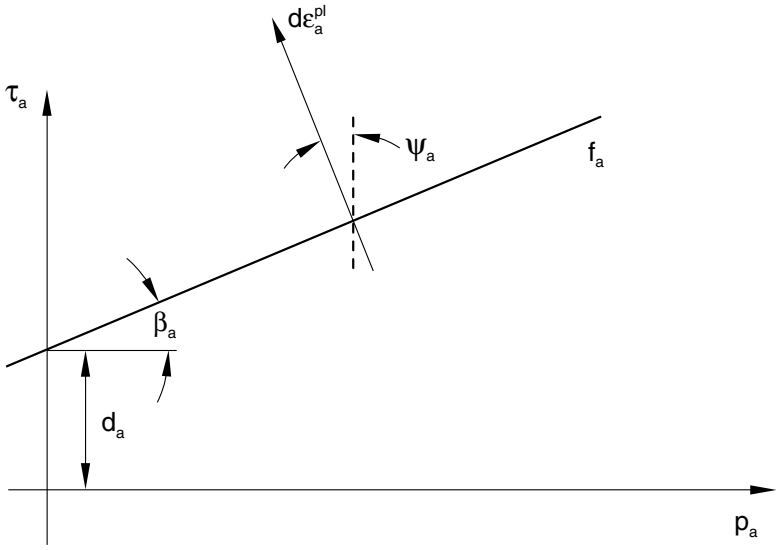


Figure 4.5.4–1 Joint system material model.

As long as $f_a < 0$, joint system a does not slip. When $f_a = 0$ joint system a slips. The inelastic ("plastic") strain on the system is then given by

$$d\gamma_{a\alpha}^{pl} = d\bar{\varepsilon}_a^{pl} \frac{\tau_{a\alpha}}{\tau_a} \cos\psi_a$$
$$d\varepsilon_{an}^{pl} = d\bar{\varepsilon}_a^{pl} \sin\psi_a,$$

where $d\gamma_{a\alpha}^{pl}$ is the rate of inelastic shear strain in direction α on the joint surface, $d\bar{\varepsilon}_a^{pl}$ is the magnitude of the inelastic strain rate, ψ_a is the dilation angle for this joint system (choosing $\psi_a = 0$ provides pure shear flow on the joint, while $\psi_a > 0$ causes dilation of the joint as it slips), and $d\varepsilon_{an}^{pl}$ is the inelastic strain normal to the joint surface. In order to add the plastic flow contributions from different systems we write the tensorial plastic strain rate for joint a as

$$d\boldsymbol{\varepsilon}_a^{pl} = d\varepsilon_{an}^{pl}\mathbf{n}_a\mathbf{n}_a + d\gamma_{a\alpha}^{pl}(\mathbf{n}_a\mathbf{t}_{a\alpha} + \mathbf{t}_{a\alpha}\mathbf{n}_a). \tag{4.5.4–4}$$

The sliding of the different joint systems at a point is independent, in the sense that sliding on one system does not change the failure criterion or the dilation angle for any other joint system at the same point. The model provides for up to three joint systems at a point.

Plastic behavior of bulk material

In addition to the joint systems, the model includes a bulk material failure mechanism. This is based on the Drucker-Prager failure criterion,

$$q - p\tan\beta_b - d_b = 0, \tag{4.5.4–5}$$

where $q \overset{\text{def}}{=} \sqrt{\frac{3}{2}\mathbf{S} : \mathbf{S}}$ is the Mises equivalent deviatoric stress (here $\mathbf{S}$ is the deviatoric stress $\mathbf{S} \overset{\text{def}}{=} \boldsymbol{\sigma} + p\mathbf{I}$), $p \overset{\text{def}}{=} -\frac{1}{3}\mathbf{I} : \boldsymbol{\sigma}$ is the equivalent pressure stress, β_b is the friction angle for the bulk material, and d_b is the cohesion for the bulk material (see Figure 4.5.4–2).

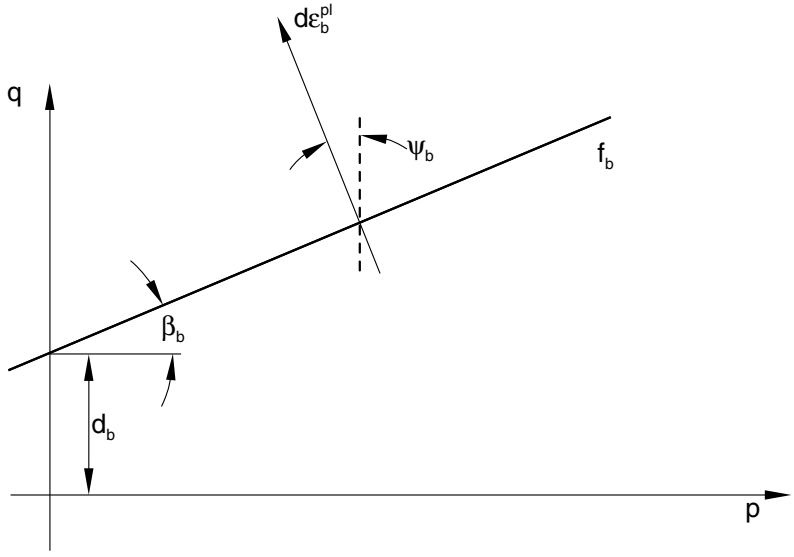


Figure 4.5.4–2 Bulk material model.

If this failure criterion is reached, the bulk inelastic flow is defined by

$$d\boldsymbol{\varepsilon}_b^{pl} = d\bar{\varepsilon}_b^{pl}\frac{1}{1 - \frac{1}{3}\tan\psi_b}\frac{\partial g_b}{\partial \boldsymbol{\sigma}}, \tag{4.5.4–6}$$

where

$$g_b = q - p \tan \psi_b \tag{4.5.4–7}$$

is the flow potential. Here, $d\bar{\varepsilon}_b^{pl}$ is the magnitude of the inelastic flow rate (chosen so that $d\bar{\varepsilon}_b^{pl} = |(d\varepsilon_b^{pl})_{11}|$ in uniaxial compression in the *1*-direction) and ψ_b is the dilation angle for the bulk material. This bulk failure model is a simplified version of the extended Drucker-Prager model described in "Models for granular or polymer behavior," Section 4.4.2. As with the joint systems, this bulk failure system is independent of the joint systems, in that bulk inelastic flow does not change the behavior of any joint system.

If $\psi \neq \beta$ in any system the flow in that system is "nonassociated." This implies that the material stiffness matrix is not symmetric, so that the unsymmetric matrix solution scheme should be invoked by the user. If the difference between ψ and β is not large, a symmetric approximation to the matrix can provide an acceptable rate of convergence of the equilibrium equations, and hence a lower overall solution cost. For this reason the unsymmetric solver is not automatically invoked for this material behavior. However, it is recommended for all cases where ψ and β are very different on any joint system.

Integration of the model

The constitutive equations described above are integrated using the backward Euler method generally used with the plasticity models in Abaqus. A material Jacobian consistent with this integration operator is used for the overall equilibrium iterations.

Reference

- "Jointed material model," Section 23.5.1 of the Abaqus Analysis User's Guide

4.6 Large-strain elasticity

- "Hyperelastic material behavior," Section 4.6.1
- "Fitting of hyperelastic and hyperfoam constants," Section 4.6.2
- "Anisotropic hyperelastic material behavior," Section 4.6.3

4.6.1 HYPERELASTIC MATERIAL BEHAVIOR

Products: Abaqus/Standard Abaqus/Explicit

The constitutive behavior of a hyperelastic material is defined as a total stress–total strain relationship, rather than as the rate formulation that has been discussed in the context of history-dependent materials in previous sections of this chapter. Therefore, the basic development of the formulation for hyperelasticity is somewhat different. Furthermore, hyperelastic materials are often incompressible or very nearly so; hence, mixed ("hybrid") formulations can be used effectively. In this section the hyperelastic model provided in Abaqus is defined, and the mixed variational principles used in Abaqus/Standard to treat the fully incompressible case and the almost incompressible case are introduced.

Definitions and basic kinematic results

We first introduce some definitions and basic kinematic results that will be used in this section. Some of these items have already been discussed in Chapter 1, "Introduction and Basic Equations": they are repeated here for convenience.

Writing the current position of a material point as $\mathbf{x}$ and the reference position of the same point as $\mathbf{X}$, the deformation gradient is

$$\mathbf{F} \stackrel{\text{def}}{=} \frac{\partial \mathbf{x}}{\partial \mathbf{X}}.$$

Then J, the total volume change at the point, is

$$J \stackrel{\text{def}}{=} \det(\mathbf{F}).$$

For simplicity, we define

$$\overline{\mathbf{F}} \stackrel{\text{def}}{=} J^{-\frac{1}{3}}\mathbf{F}$$

as the deformation gradient with the volume change eliminated.

We then introduce the deviatoric stretch matrix (the left Cauchy-Green strain tensor) of $\overline{\mathbf{F}}$ as

$$\overline{\mathbf{B}} \stackrel{\text{def}}{=} \overline{\mathbf{F}} \cdot \overline{\mathbf{F}}^T$$

so that we can define the first strain invariant as

$$\overline{I}_1 \stackrel{\text{def}}{=} \operatorname{trace} \overline{\mathbf{B}} = \mathbf{I} : \overline{\mathbf{B}}, \tag{4.6.1–1}$$

where $\mathbf{I}$ is a unit matrix, and the second strain invariant as

$$\overline{I}_2 \stackrel{\text{def}}{=} \frac{1}{2}\left(\overline{I}_1^2 - \operatorname{trace}\left(\overline{\mathbf{B}} \cdot \overline{\mathbf{B}}\right)\right) = \frac{1}{2}\left(\overline{I}_1^2 - \mathbf{I} : \overline{\mathbf{B}} \cdot \overline{\mathbf{B}}\right). \tag{4.6.1–2}$$

The variations of $\overline{\mathbf{B}}$, $\overline{\mathbf{B}} \cdot \overline{\mathbf{B}}$, $\overline{I}_1$, $\overline{I}_2$, and J will be required during the remainder of the development. We first define some variations of basic kinematic quantities that will be needed to write these results.

The gradient of the displacement variation with respect to current position is written as

$$\delta\mathbf{L} \stackrel{\text{def}}{=} \frac{\partial \delta\mathbf{u}}{\partial \mathbf{x}}.$$

The virtual rate of deformation is the symmetric part of $\delta\mathbf{L}$:

$$\delta\mathbf{D} \stackrel{\text{def}}{=} \text{sym}(\delta\mathbf{L}) = \frac{1}{2}\left(\delta\mathbf{L} + \delta\mathbf{L}^T\right),$$

which we decompose into the virtual rate of change of volume per current volume (the "virtual volumetric strain rate"),

$$\delta\varepsilon^{\text{vol}} \stackrel{\text{def}}{=} \mathbf{I} : \delta\mathbf{D},$$

and the virtual deviatoric strain rate,

$$\delta\mathbf{e} \stackrel{\text{def}}{=} \delta\mathbf{D} - \frac{1}{3}\delta\varepsilon^{\text{vol}}\mathbf{I}.$$

The virtual rate of spin of the material is the antisymmetric part of $\delta\mathbf{L}$:

$$\delta\mathbf{W} \stackrel{\text{def}}{=} \text{asym}(\delta\mathbf{L}) = \frac{1}{2}\left(\delta\mathbf{L} - \delta\mathbf{L}^T\right).$$

The variations of $\overline{\mathbf{B}}$, $\overline{\mathbf{B}} \cdot \overline{\mathbf{B}}$, $\overline{I}_1$, $\overline{I}_2$, and J are obtained directly from their definitions above in terms of these quantities as

$$\delta\overline{\mathbf{B}} = \delta\mathbf{e} \cdot \overline{\mathbf{B}} + \overline{\mathbf{B}} \cdot \delta\mathbf{e} + \delta\mathbf{W} \cdot \overline{\mathbf{B}} - \overline{\mathbf{B}} \cdot \delta\mathbf{W} = \mathbf{H}_1 : \delta\mathbf{e} + \delta\mathbf{W} \cdot \overline{\mathbf{B}} - \overline{\mathbf{B}} \cdot \delta\mathbf{W},$$

where

$$(H_1)_{ijkl} \stackrel{\text{def}}{=} \frac{1}{2}\left(\delta_{ik}\overline{B}_{jl} + \overline{B}_{ik}\delta_{jl} + \delta_{il}\overline{B}_{jk} + \overline{B}_{il}\delta_{jk}\right);$$

$$\begin{aligned}\delta(\overline{\mathbf{B}} \cdot \overline{\mathbf{B}}) &= \delta\mathbf{e} \cdot \overline{\mathbf{B}} \cdot \overline{\mathbf{B}} + \overline{\mathbf{B}} \cdot \overline{\mathbf{B}} \cdot \delta\mathbf{e} + 2\overline{\mathbf{B}} \cdot \delta\mathbf{e} \cdot \overline{\mathbf{B}} + \delta\mathbf{W} \cdot \overline{\mathbf{B}} \cdot \overline{\mathbf{B}} - \overline{\mathbf{B}} \cdot \overline{\mathbf{B}} \cdot \delta\mathbf{W}, \\ &= \mathbf{H}_2 : \delta\mathbf{e} + \delta\mathbf{W} \cdot \overline{\mathbf{B}} \cdot \overline{\mathbf{B}} - \overline{\mathbf{B}} \cdot \overline{\mathbf{B}} \cdot \delta\mathbf{W},\end{aligned}$$

where

$$(H_2)_{ijkl} \stackrel{\text{def}}{=} \frac{1}{2}\left(\delta_{ik}\overline{B}_{jp}\overline{B}_{pl} + \overline{B}_{ip}\overline{B}_{pk}\delta_{jl} + \delta_{il}\overline{B}_{jp}\overline{B}_{pk} + \overline{B}_{ip}\overline{B}_{pl}\delta_{jk}\right) + \overline{B}_{ik}\overline{B}_{jl} + \overline{B}_{il}\overline{B}_{jk};$$

$$\delta\overline{I}_1 = 2\overline{\mathbf{B}} : \delta\mathbf{e}; \tag{4.6.1–3}$$

$$\delta \overline{I}_2 = 2(\overline{I}_1 \overline{\mathbf{B}} - \overline{\mathbf{B}} \cdot \overline{\mathbf{B}}) : \delta \mathbf{e}; \tag{4.6.1–4}$$

and

$$\delta J = J \delta \varepsilon^{\text{vol}}. \tag{4.6.1–5}$$

The Cauchy ("true") stress components are defined from the strain energy potential as follows. From the virtual work principal the internal energy variation is

$$\delta W_I = \int_V \boldsymbol{\sigma} : \delta \mathbf{D} \, dV = \int_{V^0} J \boldsymbol{\sigma} : \delta \mathbf{D} \, dV^0,$$

where $\boldsymbol{\sigma}$ are the components of the Cauchy ("true") stress, V is the current volume, and V^0 is the reference volume.

We decompose the stress into the equivalent pressure stress,

$$p \overset{\text{def}}{=} -\frac{1}{3} \mathbf{I} : \boldsymbol{\sigma},$$

and the deviatoric stress,

$$\mathbf{S} \overset{\text{def}}{=} \boldsymbol{\sigma} + p \mathbf{I},$$

so that the internal energy variation can be written

$$\delta W_I = \int_{V^0} J(\mathbf{S} : \delta \mathbf{e} - p \, \delta \varepsilon^{\text{vol}}) \, dV^0.$$

For isotropic, compressible materials the strain energy, U, is a function of $\overline{I}_1$, $\overline{I}_2$, and J:

$$U = U(\overline{I}_1, \overline{I}_2, J),$$

so that

$$\delta U = \frac{\partial U}{\partial \overline{I}_1} \delta \overline{I}_1 + \frac{\partial U}{\partial \overline{I}_2} \delta \overline{I}_2 + \frac{\partial U}{\partial J} \delta J.$$

Hence, using Equation 4.6.1–3, Equation 4.6.1–4, and Equation 4.6.1–5,

$$\delta U = 2 \left[\left(\frac{\partial U}{\partial \overline{I}_1} + \overline{I}_1 \frac{\partial U}{\partial \overline{I}_2} \right) \overline{\mathbf{B}} - \frac{\partial U}{\partial \overline{I}_2} \overline{\mathbf{B}} \cdot \overline{\mathbf{B}} \right] : \delta \mathbf{e} + J \frac{\partial U}{\partial J} \, \delta \varepsilon^{\text{vol}}. \tag{4.6.1–6}$$

Since the variation of the strain energy potential is, by definition, the internal virtual work per reference volume, δW_I, we have

$$\delta W_I = \int_{V^0} J(\mathbf{S} : \delta\mathbf{e} - p\,\delta\varepsilon^{\text{vol}})\,dV^0 = \int_{V^0} \delta U\,dV^0. \tag{4.6.1–7}$$

For a compressible material the strain variations are arbitrary, so this equation defines the stress components for such a material as

$$\mathbf{S} = \frac{2}{J}\,\text{DEV}\left[\left(\frac{\partial U}{\partial \overline{I}_1} + \overline{I}_1\frac{\partial U}{\partial \overline{I}_2}\right)\overline{\mathbf{B}} - \frac{\partial U}{\partial \overline{I}_2}\overline{\mathbf{B}}\cdot\overline{\mathbf{B}}\right] \tag{4.6.1–8}$$

and

$$p = -\frac{\partial U}{\partial J}. \tag{4.6.1–9}$$

When the material response is almost incompressible, the pure displacement formulation, in which the strain invariants are computed from the kinematic variables of the finite element model, can behave poorly. One difficulty is that from a numerical point of view the stiffness matrix is almost singular because the effective bulk modulus of the material is so large compared to its effective shear modulus, thus causing difficulties with the solution of the discretized equilibrium equations. Similarly, in Abaqus/Explicit the high bulk modulus increases the dilatational wave speed, thus reducing the stable time increment substantially. Another problem is that, unless reduced-integration techniques are used, the stresses calculated at the numerical integration points show large oscillations in the pressure stress values, because—in general—the elements cannot respond accurately and still have small volume changes at all numerical integration points. To avoid such problems, Abaqus/Standard offers a "mixed" formulation for such cases. The concept is to introduce a variable, $\hat{J}$, that is used in place of the volume change, J, in the definition of the strain energy potential. The internal energy integral, W_I, is augmented with the constraint that $J - \hat{J} = 0$, imposed by the use of a Lagrange multiplier, $\hat{p}$, and integrated over the volume:

$$(W_I)^A \overset{\text{def}}{=} \int_{V^0}\left[U(\overline{I}_1, \overline{I}_2, \hat{J}) - \hat{p}(J - \hat{J})\right] dV^0.$$

Taking the variation of this definition,

$$\delta(W_I)^A = \int_{V^0}\left[J\mathbf{S} : \delta\mathbf{e} + \left(\frac{\partial U}{\partial \hat{J}} + \hat{p}\right)\delta\hat{J} - J\hat{p}\,\delta\varepsilon^{\text{vol}} - (J - \hat{J})\,\delta\hat{p}\right] dV^0.$$

Since $\delta\hat{J}$ is an independent variation in this expression, the Lagrange multiplier is

$$\hat{p} = -\frac{\partial U}{\partial \hat{J}},$$

and its variation is

$$\delta\hat{p} = -\hat{\mathbf{Q}} : \delta\mathbf{e} + \frac{\partial\hat{p}}{\partial\hat{J}}\,\delta\hat{J},$$

where

$$\hat{\mathbf{Q}} \overset{\text{def}}{=} J\frac{\partial\mathbf{S}}{\partial\hat{J}}.$$

These results allow us to write the augmented internal energy variation as

$$\delta(W_I)^A = \int_{V^0}\left[\left(J\mathbf{S} + (J-\hat{J})\hat{\mathbf{Q}}\right) : \delta\mathbf{e} - J\hat{p}\,\delta\varepsilon^{\text{vol}} - (J-\hat{J})\frac{\partial\hat{p}}{\partial\hat{J}}\,\delta\hat{J}\right] dV^0, \tag{4.6.1–10}$$

which implies that continuity of the $\hat{J}$ interpolation across elements is not required.

This augmented formulation can be used for any value of compressibility except fully incompressible behavior. For most element types $\hat{J}$ is interpolated independently in each element: Abaqus uses constant $\hat{J}$ in most first-order elements and linear variation of $\hat{J}$ with respect to position in second-order elements. The only element type where continuity of the $\hat{J}$ interpolation across elements is enforced is C3D4H: a first-order tetrahedron with a linear interpolation of $\hat{J}$ continuous across elements.

When the material is fully incompressible, U is a function of the first and second strain invariants—$\overline{I}_1$ and $\overline{I}_2$—only, and we write the internal energy in the augmented form,

$$(W_I)^A \overset{\text{def}}{=} \int_{V^0}\left[U - \hat{p}(J-1)\right] dV^0,$$

where $\hat{p}$ is again a Lagrange multiplier introduced to impose the constraint $J-1=0$ in such a way that the variation of $(W_I)^A$ can be taken with respect to all kinematic variables, thus giving

$$\delta(W_I)^A = \int_{V^0}\left[J\mathbf{S} : \delta\mathbf{e} - J\hat{p}\,\delta\varepsilon^{\text{vol}} - (J-1)\,\delta\hat{p}\right] dV^0. \tag{4.6.1–11}$$

The Lagrange multiplier $\hat{p}$ is interpolated in the same way as $\hat{J}$ is interpolated in the augmented formulation for almost incompressible behavior; that is, $\hat{p}$ is assumed to be constant in most first-order elements and to vary linearly with respect to position in second-order elements. The only exception is element type C3D4H, which is a first-order tetrahedron with a linear interpolation of $\hat{p}$ continuous across elements.

Rate of change of the internal virtual work

The rate of change of the internal virtual work is required for use in the Newton method, which is generally used in Abaqus/Standard to solve the nonlinear equilibrium equations (after discretization by finite elements). It will also be used when we extend the elasticity model to viscoelastic behavior for small (linearized) vibrations about a predeformed state.

When the pure displacement formulation is used for the compressible case, the deviatoric stress components, $\mathbf{S}$, are defined by Equation 4.6.1–8, from which we can show that

$$d(J\mathbf{S}) = J(\mathbf{C}^S : d\mathbf{e} + \mathbf{Q}\, d\varepsilon^{\mathrm{vol}} + d\mathbf{W}\cdot\mathbf{S} - \mathbf{S}\cdot d\mathbf{W}),$$

where the "effective deviatoric elasticity" of the material, $\mathbf{C}^S$, is defined as

$$\begin{aligned}\mathbf{C}^S \overset{\text{def}}{=}\ & \frac{2}{J}\left(\frac{\partial U}{\partial \overline{I}_1} + \overline{I}_1\frac{\partial U}{\partial \overline{I}_2}\right)\mathbf{H}_1 - \frac{2}{J}\frac{\partial U}{\partial \overline{I}_2}\,\mathbf{H}_2 + \frac{4}{J}\left(\frac{\partial^2 U}{\partial \overline{I}_1{}^2} + \frac{\partial U}{\partial \overline{I}_2} + 2\overline{I}_1\frac{\partial^2 U}{\partial \overline{I}_1\partial \overline{I}_2} + \overline{I}_1{}^2\frac{\partial^2 U}{\partial \overline{I}_2{}^2}\right)\overline{\mathbf{B}}\,\overline{\mathbf{B}} \\ & - \frac{4}{J}\left(\frac{\partial^2 U}{\partial \overline{I}_1\partial \overline{I}_2} + \overline{I}_1\frac{\partial^2 U}{\partial \overline{I}_2{}^2}\right)\left(\overline{\mathbf{B}}\cdot\overline{\mathbf{B}}\,\overline{\mathbf{B}} + \overline{\mathbf{B}}\,\overline{\mathbf{B}}\cdot\overline{\mathbf{B}}\right) + \frac{4}{J}\frac{\partial^2 U}{\partial \overline{I}_2{}^2}\,\overline{\mathbf{B}}\cdot\overline{\mathbf{B}}\,\overline{\mathbf{B}}\cdot\overline{\mathbf{B}} \\ & - \frac{4}{3J}\left[\frac{\partial U}{\partial \overline{I}_1} + 2\overline{I}_1\frac{\partial U}{\partial \overline{I}_2} + \overline{I}_1\frac{\partial^2 U}{\partial \overline{I}_1{}^2} + \left(\overline{I}_1{}^2 + 2\overline{I}_2\right)\frac{\partial^2 U}{\partial \overline{I}_1\partial \overline{I}_2} + 2\overline{I}_1\overline{I}_2\frac{\partial^2 U}{\partial \overline{I}_2{}^2}\right]\left(\mathbf{I}\,\overline{\mathbf{B}} + \overline{\mathbf{B}}\,\mathbf{I}\right) \\ & + \frac{4}{3J}\left(2\frac{\partial U}{\partial \overline{I}_2} + \overline{I}_1\frac{\partial^2 U}{\partial \overline{I}_1\partial \overline{I}_2} + 2\overline{I}_2\frac{\partial^2 U}{\partial \overline{I}_2{}^2}\right)\left(\mathbf{I}\,\overline{\mathbf{B}}\cdot\overline{\mathbf{B}} + \overline{\mathbf{B}}\cdot\overline{\mathbf{B}}\,\mathbf{I}\right),\end{aligned}$$

and the deviatoric stress rate-volumetric strain rate coupling term, $\mathbf{Q}$, is

$$\mathbf{Q} \overset{\text{def}}{=} \frac{\partial(J\mathbf{S})}{\partial J} = 2\left(\frac{\partial^2 U}{\partial \overline{I}_1\partial J} + \overline{I}_1\frac{\partial^2 U}{\partial \overline{I}_2\partial J}\right)\overline{\mathbf{B}} - 2\frac{\partial^2 U}{\partial \overline{I}_2\partial J}\overline{\mathbf{B}}\cdot\overline{\mathbf{B}} - \frac{2}{3}\left(\overline{I}_1\frac{\partial^2 U}{\partial \overline{I}_1\partial J} + 2\overline{I}_2\frac{\partial^2 U}{\partial \overline{I}_2\partial J}\right)\mathbf{I}.$$

From Equation 4.6.1–9 it can be shown that

$$d(Jp) = -J(\mathbf{Q} : d\mathbf{e} + K d\varepsilon^{\mathrm{vol}}),$$

where $\boldsymbol{K}$ is the effective bulk modulus of the material,

$$K \overset{\text{def}}{=} -\left(J\frac{\partial p}{\partial J} + p\right) = J\frac{\partial^2 U}{\partial J^2} + \frac{\partial U}{\partial J}.$$

Thus,

$$d\delta W_I = \int_V \left[\begin{array}{cc}\lfloor\, \delta\mathbf{e} & \delta\varepsilon^{\mathrm{vol}}\,\rfloor\end{array} : \begin{bmatrix}\mathbf{C}^S & \mathbf{Q}\\ \mathbf{Q} & K\end{bmatrix} : \begin{Bmatrix} d\mathbf{e}\\ d\varepsilon^{\mathrm{vol}}\end{Bmatrix} - \boldsymbol{\sigma} : (2\delta\boldsymbol{\varepsilon}\cdot d\boldsymbol{\varepsilon} - \delta\mathbf{L}^T\cdot d\mathbf{L})\right] dV, \qquad (4.6.1\text{–}12)$$

since

$$\delta\mathbf{e} : (d\mathbf{W}\cdot\mathbf{S} - \mathbf{S}\cdot d\mathbf{W}) + \mathbf{S} : d\delta\mathbf{e} - p d\delta\varepsilon^{\mathrm{vol}} = -\boldsymbol{\sigma} : (2\delta\boldsymbol{\varepsilon}\cdot d\boldsymbol{\varepsilon} - \delta\mathbf{L}^T\cdot d\mathbf{L}).$$

For the mixed formulation introduced for almost incompressible materials, the rate of change of the augmented variation of internal energy, Equation 4.6.1–10, is

$$d\delta(W_I)^A = \int_V \Bigg[\lfloor\, \delta\mathbf{e} \quad \delta\varepsilon^{\text{vol}} \quad \delta\hat{J} \,\rfloor \begin{bmatrix} \tilde{\mathbf{C}}^S & \hat{\mathbf{Q}} & \hat{\mathbf{A}} \\ \hat{\mathbf{Q}} & -\hat{p} & -\partial\hat{p}/\partial\hat{J} \\ \hat{\mathbf{A}} & -\partial\hat{p}/\partial\hat{J} & -\tilde{K} \end{bmatrix} \begin{Bmatrix} d\mathbf{e} \\ d\varepsilon^{\text{vol}} \\ d\hat{J} \end{Bmatrix} - \left(\boldsymbol{\sigma} + \frac{1}{J}(J-\hat{J})\hat{\mathbf{Q}} \right) : (2\delta\boldsymbol{\varepsilon}\cdot d\boldsymbol{\varepsilon} - \delta\mathbf{L}^T \cdot d\mathbf{L}) \Bigg] dV,$$

where

$$\tilde{\mathbf{C}}^S \overset{\text{def}}{=} \mathbf{C}^S + (J-\hat{J})\frac{\partial \mathbf{C}^S}{\partial \hat{J}},$$

$$\hat{\mathbf{Q}} \overset{\text{def}}{=} J\frac{\partial \mathbf{S}}{\partial \hat{J}},$$

$$\hat{\mathbf{A}} \overset{\text{def}}{=} \frac{1}{J}(J-\hat{J})\frac{\partial \hat{\mathbf{Q}}}{\partial \hat{J}},$$

and

$$\tilde{K} \overset{\text{def}}{=} \frac{1}{J^2}\left(\hat{K} - (J-\hat{J})\frac{\partial \hat{K}}{\partial \hat{J}} \right),$$

in which

$$\hat{K} \overset{\text{def}}{=} -J\frac{\partial \hat{p}}{\partial \hat{J}}.$$

For the case of incompressible materials the rate of change of the augmented variation of internal energy is similarly obtained from Equation 4.6.1–11 as

$$d\delta(W_I)^A = \int_V \Bigg[\lfloor\, \delta\mathbf{e} \quad \delta\varepsilon^{\text{vol}} \quad \delta\hat{p} \,\rfloor \begin{bmatrix} \mathbf{C}^S & 0 & 0 \\ 0 & -\hat{p} & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{Bmatrix} d\mathbf{e} \\ d\varepsilon^{\text{vol}} \\ d\hat{p} \end{Bmatrix} - \boldsymbol{\sigma} : (2\delta\boldsymbol{\varepsilon}\cdot d\boldsymbol{\varepsilon} - \delta\mathbf{L}^T \cdot d\mathbf{L}) \Bigg] dV.$$

Particular forms of the strain energy potential

Several particular forms of the strain energy potential are available in Abaqus. The incompressible or almost incompressible models make up:

- the polynomial form and its particular cases—the reduced polynomial form, the neo-Hookean form, the Mooney-Rivlin form, and the Yeoh form;
- the Ogden form;

- the Arruda-Boyce form; and
- the Van der Waals form.

In addition, a hyperelastic model for highly compressible, elastic materials is offered.

Polynomial form and particular cases

Given isotropy and additive decomposition of the deviatoric and volumetric strain energy contributions in the presence of incompressible or almost incompressible behavior, we can write the potential, in general, as

$$U = f(\overline{I}_1 - 3, \overline{I}_2 - 3) + g(J_{e\ell} - 1).$$

Setting $g = \sum_{i=1}^{N} \frac{1}{D_i}(J_{e\ell} - 1)^{2i}$ and expanding $f(\overline{I}_1 - 3, \overline{I}_2 - 3)$ in a Taylor series, we arrive at

$$U \overset{\text{def}}{=} \sum_{i+j=1}^{N} C_{ij}(\overline{I}_1 - 3)^i(\overline{I}_2 - 3)^j + \sum_{i=1}^{N} \frac{1}{D_i}(J_{e\ell} - 1)^{2i}. \tag{4.6.1–13}$$

This form is the polynomial representation of the strain energy in Abaqus. The parameter N can take values up to six; however, values of N greater than 2 are rarely used when both the first and second invariants are taken into account. C_{ij} and D_i are temperature-dependent material parameters. The value of N and tables giving the C_{ij} and D_i values as functions of temperature are specified by the user. The elastic volume strain, $J_{e\ell}$, follows from the total volume strain, J, and the thermal volume strain, J_{th}, with the relation

$$J_{e\ell} = \frac{J}{J_{th}},$$

and J_{th} follows from the linear thermal expansion, ε_{th}, with

$$J_{th} = (1 + \varepsilon_{th})^3,$$

where ε_{th} follows from the temperature and the isotropic thermal expansion coefficient defined by the user.

The D_i values determine the compressibility of the material: if all the D_i are zero, the material is taken as fully incompressible. If $D_1 = 0$, all D_i must be zero.

Regardless of the value of N, the initial shear modulus, μ_0, and the bulk modulus, k_0, depend only on the polynomial coefficients of order $N = 1$:

$$\mu_0 = 2(C_{10} + C_{01}), \qquad k_0 = \frac{2}{D_1}.$$

If $N = 1$, so that only the linear terms in the deviatoric strain energy are retained, the Mooney-Rivlin form is recovered:

$$U = C_{10}(\overline{I}_1 - 3) + C_{01}(\overline{I}_2 - 3) + \frac{1}{D_1}(J_{e\ell} - 1)^2.$$

The Mooney-Rivlin form can be viewed as an extension of the neo-Hookean form (discussed below) in that it adds a term that depends on the second invariant of the left Cauchy-Green tensor. In some cases this form will give a more accurate fit to the experimental data than the neo-Hookean form; in general, however, both models give similar accuracy since they use only linear functions of the invariants. These functions do not allow representation of the "upturn" at higher strain levels in the stress-strain curve.

Particular forms of the polynomial model can also be obtained by setting specific coefficients to zero. If all C_{ij} with $j \neq 0$ are set to zero, the reduced polynomial form is obtained:

$$U = \sum_{i=1}^{N} C_{i0}(\overline{I}_1 - 3)^i + \sum_{i=1}^{N} \frac{1}{D_i}(J_{e\ell} - 1)^{2i}.$$

Following Yeoh (1993) the justification for reducing the general polynomial series expansion by omitting the dependence on the second invariant arises from the following observations. The sensitivity of the strain energy function to changes in the second invariant is generally much smaller than the sensitivity to changes in the first invariant. In addition, the $\overline{I}_2$-dependence is difficult to measure, so it might be preferable to neglect it rather than to calculate it based on potentially inaccurate measurements. Finally, it appears that omitting the dependence on the second invariant if data for only a particular mode of deformation are known might enhance the prediction for other deformation states. This conjecture is supported by investigating the so-called reduced stresses in the presence of almost incompressible behavior:

$$\begin{aligned}
\frac{\sigma_1 - \sigma_2}{\lambda_1^2 - \lambda_2^2} &= 2\left(\frac{\partial U}{\partial \overline{I}_1} + \lambda_3^2 \frac{\partial U}{\partial \overline{I}_2}\right), \\
\frac{\sigma_1 - \sigma_3}{\lambda_1^2 - \lambda_3^2} &= 2\left(\frac{\partial U}{\partial \overline{I}_1} + \lambda_2^2 \frac{\partial U}{\partial \overline{I}_2}\right), \\
\frac{\sigma_2 - \sigma_3}{\lambda_2^2 - \lambda_3^2} &= 2\left(\frac{\partial U}{\partial \overline{I}_1} + \lambda_1^2 \frac{\partial U}{\partial \overline{I}_2}\right),
\end{aligned}$$

where the σ_i, $i = 1 \ldots 3$ represent the principal Cauchy ("true") stresses. If the derivatives with respect to $\overline{I}_2$ are omitted and different stress states—uniaxial, biaxial, and planar—are considered, the reduced stresses have the same form regardless of the stress state.

Measurements of the $\overline{I}_2$-dependence of carbon-black reinforced rubber vulcanizates confirming these findings can be found in Kawabata, Yamashita, et al. (1995). The paper of Kaliske and Rothert (1997) also supports the notion that often the prediction of general deformation states based on a uniaxial measurement can be enhanced only by ignoring the $\overline{I}_2$-dependence.

In this context it is worth noting that the mathematical structure of the Arruda-Boyce model can be viewed as a fifth-order reduced polynomial, where the five coefficients $C_{10} \ldots C_{50}$ are implicit nonlinear functions of the two parameters μ and λ_m in the Arruda-Boyce form. However, the Arruda-Boyce model

offers a physical interpretation of the parameters, which the general fifth-order reduced polynomial fails to provide.

The Yeoh form (Yeoh, 1993) can be viewed as a special case of the reduced polynomial with $N = 3$:

$$U = \sum_{i=1}^{3} C_{i0}(\overline{I}_1 - 3)^i + \sum_{i=1}^{3} \frac{1}{D_i}(J_{e\ell} - 1)^{2i}.$$

Typically, if $C_{10} = O(1)$, the second coefficient will be negative and one to two orders of magnitude smaller [i.e., C_{20} is $-O(0.1)$ to $-O(0.01)$], while the third coefficient C_{30} is again one to two orders of magnitude smaller but positive [i.e., C_{30} is $+O(1.E-2)$ to $+O(1.E-4)$]. These magnitudes will create the typical S-shape of the stress-strain behavior of rubber; at low strains C_{10} represents the initial shear modulus, which softens at moderate strains due to the effect of the negative second coefficient C_{20} and is followed by an upturn at large strains due to the positive third coefficient C_{30}. Thus, the Yeoh model often provides an accurate fit over a large strain range.

If the reduced-polynomial strain-energy function is simplified further by setting $N = 1$, the neo-Hookean form is obtained:

$$U = C_{10}(\overline{I}_1 - 3) + \frac{1}{D_1}(J_{e\ell} - 1)^2.$$

This form is the simplest hyperelastic model and often serves as a prototype for elastomeric materials in the absence of accurate material data. It also has some theoretical relevance since the mathematical representation is analogous to that of an ideal gas: the neo-Hookean potential represents the Helmholtz free energy of a molecular network with Gaussian chain-length distribution (see Treloar, 1975).

The user can request that Abaqus calculate the C_{ij} and D_i values from measurements of nominal stress and strain in simple experiments. The basis of this calculation is described in "Fitting of hyperelastic and hyperfoam constants," Section 4.6.2.

Ogden form

The Ogden strain energy potential is expressed in terms of the principal stretches. In Abaqus the following formulation is used:

$$U \overset{\text{def}}{=} \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i^2}(\overline{\lambda}_1^{\alpha_i} + \overline{\lambda}_2^{\alpha_i} + \overline{\lambda}_3^{\alpha_i} - 3) + \sum_{i=1}^{N} \frac{1}{D_i}(J_{e\ell} - 1)^{2i}, \tag{4.6.1–14}$$

where

$$\overline{\lambda}_i = J^{-\frac{1}{3}}\lambda_i \rightarrow \overline{\lambda}_1\overline{\lambda}_2\overline{\lambda}_3 = 1.$$

Hence, the first part of Ogden's strain energy function depends only on $\overline{I}_1$ and $\overline{I}_2$. Ogden's energy function cannot be written explicitly in terms of $\overline{I}_1$ and $\overline{I}_2$. It is, however, possible to obtain closed-form expressions for the derivatives of U with respect to $\overline{I}_1$ and $\overline{I}_2$.

The value of N and tables giving the μ_i and α_i values as functions of temperature are specified by the user. If $N = 2$, $\alpha_1 = 2$, and $\alpha_2 = -2$, the Mooney-Rivlin model is obtained. If $N = 1$ and $\alpha_1 = 2$, Ogden's model degenerates to the neo-Hookean material model. In the Ogden form the initial shear modulus, μ_0, depends on all coefficients:

$$\mu_0 = \sum_{i=1}^{N} \mu_i,$$

and the initial bulk modulus, k_0, depends on D_1 as before. The user can request that Abaqus calculate the μ_i and α_i values from measurements of nominal stress and strain.

Arruda-Boyce form

The hyperelastic Arruda-Boyce potential has the following form:

$$U = \mu \sum_{i=1}^{5} \frac{C_i}{\lambda_m^{2i-2}} \left(\overline{I}_1^i - 3^i \right) + \frac{1}{D} \left(\frac{J_{e\ell}^2 - 1}{2} - \ln J_{e\ell} \right),$$

where

$$C_1 = \frac{1}{2}, \qquad C_2 = \frac{1}{20}, \qquad C_3 = \frac{11}{1050}, \qquad C_4 = \frac{19}{7000}, \quad \text{and} \quad C_5 = \frac{519}{673750}.$$

The deviatoric part of the strain energy density comes from Arruda and Boyce (1993). This model is also known as the eight-chain model, since it was developed starting out from a representative volume element where eight springs emanate from the center of a cube to its corners. The values of the coefficients $C_1 \ldots C_5$ arise from a series expansion of the inverse Langevin function, which arises in the statistical treatment of non-Gaussian chains. The series expansion is truncated after the fifth term. The coefficient λ_m is referred to as the locking stretch. Approximately at this stretch the slope of the stress-strain curve will rise significantly. The initial shear modulus, μ_0, is related to μ with the expression

$$\mu_0 = \mu(1 + \frac{3}{5\lambda_m^2} + \frac{99}{175\lambda_m^4} + \frac{513}{875\lambda_m^6} + \frac{42039}{67375\lambda_m^8}).$$

A typical value of λ_m is 7, for which $\mu_0 = 1.0125\mu$.

The initial bulk modulus is obtained as $K_0 = 2/D$. To the deviatoric part of the strain energy density we add a simplified representation of the volumetric strain energy density, which requires only one material parameter, so that all material parameters can be estimated easily even with limited knowledge of the material behavior. This volumetric representation has been used successfully by Kaliske and Rothert (1997) and provides sufficient accuracy for most engineering elastomeric materials.

The Arruda-Boyce potential depends on the first invariant only. The physical interpretation is that the eight chains are stretched equally under the action of a general deformation state. The first invariant, $\overline{I}_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2$, directly represents this elongation.

When the Arruda-Boyce form is chosen, the user can specify the coefficients as functions of temperature; alternatively, Abaqus can perform a fit of the test data specified by the user to determine the coefficients.

Van der Waals form

The hyperelastic Van der Waals potential, also known as the Kilian model, has the following form:

$$U = \mu\left\{-(\lambda_m^2 - 3)\left[\ln(1-\eta) + \eta\right] - \frac{2}{3}a\left(\frac{\tilde{I}-3}{2}\right)^{\frac{3}{2}}\right\} + \frac{1}{D}\left(\frac{J_{e\ell}^2 - 1}{2} - \ln J_{e\ell}\right),$$

where

$$\tilde{I} = (1-\beta)\overline{I}_1 + \beta\overline{I}_2 \quad \text{and} \quad \eta = \sqrt{\frac{\tilde{I}-3}{\lambda_m^2 - 3}}.$$

The name "Van der Waals" draws on the analogy in the thermodynamic interpretation of the equations of state for rubber and gas. While the neo-Hookean model can be compared with an ideal gas in that it starts out from a Gaussian network with no mutual interaction between the "quasi-particles" (Kilian, 1981), the Van der Waals strain energy potential is analogous to the equations of state of a real gas. This introduces two additional material parameters: the locking stretch, λ_m, and the global interaction parameter, a. (Similarly, the Van der Waals equation for a real gas introduces two parameters to account for excluded volume and modified exchange of momentum between the particles.)

The locking stretch, λ_m, accounts for finite extendability of the non-Gaussian chain network. In contrast to the Arruda-Boyce model the mathematical structure of the Van der Waals potential is such that the strain energy tends to infinity as the locking stretch, λ_m, is reached; more precisely, as $\tilde{I} \to \lambda_m^2$. Thus, the Van der Waals potential cannot be used at stretches larger than the locking stretch.

The global interaction parameter, a, models the interaction between the chains; it is difficult to estimate. Kilian et al. (1986) point out that, given Mooney-Rivlin coefficients and a locking stretch λ_m, a suitable value for the global interaction parameter is

$$a = \frac{2C_{01}}{3\mu} + \frac{\lambda_m^2}{\lambda_m^3 - 1},$$

where μ is the initial shear modulus at low strains and C_{01} is the second Mooney-Rivlin parameter. Given a positive initial shear modulus, μ, and locking stretch, λ_m, too large a positive interaction parameter, a, will lead to Drucker instability in the tensile range. Realistic values of the global interaction parameter, a, will contribute to the characteristic S-shape of tensile stress-strain curves of rubber in the middle strain range before the final upturn as the locking stretch is approached, without causing instability.

The parameter β represents a linear mixture parameter combining both invariants $\overline{I}_1$ and $\overline{I}_2$ into $\tilde{I}$; for $\beta = 0.0$, the Van der Waals potential will be dependent on the first invariant only. Admissible values for this parameter are $0.0 \le \beta \le 1.0$.

When the Van der Waals potential is chosen, the user can specify the coefficients as functions of temperature; alternatively, the parameters can be fitted from user-defined test data. The data fitting

procedure will not necessarily yield a value of β between zero and one. If during the curve fitting procedure the parameter β leaves the admissible range, the curve fitting procedure is aborted and restarted with a fixed value of $\beta = 0.0$. Alternatively, the curve fitting procedure can be used with a user-defined value of β.

Strain energy potential for highly compressible elastomers

While the previous forms are intended for incompressible or almost incompressible materials, the elastic foam energy function is designed for describing highly compressible elastomers (Storåkers, 1986). This energy function has the form

$$U \stackrel{\text{def}}{=} \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i^2} \left[\hat{\lambda}_1^{\alpha_i} + \hat{\lambda}_2^{\alpha_i} + \hat{\lambda}_3^{\alpha_i} - 3 + \frac{1}{\beta_i}(J_{e\ell}^{-\alpha_i\beta_i} - 1) \right], \tag{4.6.1–15}$$

where

$$\hat{\lambda}_i = J_{th}^{-\frac{1}{3}} \lambda_i \rightarrow \hat{\lambda}_1 \hat{\lambda}_2 \hat{\lambda}_3 = J_{e\ell}.$$

The volumetric and the deviatoric contributions are coupled in this expression, which can be demonstrated clearly by writing the expression in the form

$$U = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i^2} \left[J_{e\ell}^{\frac{1}{3}\alpha_i} (\overline{\lambda}_1^{\alpha_i} + \overline{\lambda}_2^{\alpha_i} + \overline{\lambda}_3^{\alpha_i} - 3) + 3(J_{e\ell}^{\frac{1}{3}\alpha_i} - 1) + \frac{1}{\beta_i}(J_{e\ell}^{-\alpha_i\beta_i} - 1) \right]. \tag{4.6.1–16}$$

Series expansion of the last two terms in terms of $J_{e\ell} - 1$ shows that the first-order terms vanish and that the coefficients of the second-order terms are equal to $\frac{1}{2}\alpha_i^2(\frac{1}{3} + \beta_i)(J_{e\ell} - 1)^2$. Hence, a stable material is obtained if $\beta_i > -\frac{1}{3}$. For each term in the energy function, the coefficient β_i determines the degree of compressibility. β_i is related to the Poisson's ratio, ν_i, by the expressions

$$\beta_i = \frac{\nu_i}{1 - 2\nu_i}, \qquad \nu_i = \frac{\beta_i}{1 + 2\beta_i}.$$

Thus, if β_i is the same for all terms, we have a single effective Poisson's ratio, ν. The value of N and tables giving the μ_i, α_i, and ν_i values as functions of temperature are specified by the user for the hyperfoam material model. The Poisson's ratio is valid for finite values of the logarithmic principal strains $e_1,\ e_2,\ e_3$; $e_2 = e_3 = -\nu e_1$ in uniaxial tension. For $\beta_i = \nu_i = 0$ there is no Poisson's effect. The initial shear modulus, μ_0, again follows from

$$\mu_0 = \sum_{i=1}^{N} \mu_i,$$

and the initial bulk modulus follows from

$$k_0 = \sum_{i=1}^{N} 2\mu_i(\frac{1}{3} + \beta_i).$$

If Poisson's ratio is constant and known, Abaqus can calculate the μ_i and α_i from measurements of nominal stress and stretch as before. If Poisson's ratio depends on the level of straining, Abaqus can also calculate the β_i from the nominal lateral strains.

Subroutine `UHYPER`

Abaqus/Standard also allows other forms of strain energy potentials to be defined for isotropic materials via user subroutine **`UHYPER`** by programming the first and second derivatives of U with respect to $\overline{I}_1$, $\overline{I}_2$, and J in that subroutine.

References

- "Hyperelastic behavior of rubberlike materials," Section 22.5.1 of the Abaqus Analysis User's Guide
- "Hyperelastic behavior in elastomeric foams," Section 22.5.2 of the Abaqus Analysis User's Guide

4.6.2 FITTING OF HYPERELASTIC AND HYPERFOAM CONSTANTS

Products: Abaqus/Standard Abaqus/Explicit

In this section we will derive the equations needed for fitting the hyperelastic (polynomial, Ogden, Arruda-Boyce, and Van der Waals form) and hyperfoam constants to experimental test data. In addition, the procedures for checking the material stability using the Drucker criterion will be described.

For the hyperelastic models full incompressibility is assumed in fitting the hyperelastic constants to the test data, except in the volumetric test.

Stress-strain relations for the polynomial strain energy potential

The hyperelastic polynomial form can be fitted by Abaqus up to order $N = 2$. Since the Mooney-Rivlin potential corresponds to the case $N = 1$, these remarks also apply by setting the higher-order coefficients to zero. The energy potential is as follows:

$$\begin{aligned} U =& C_{10}(\overline{I}_1 - 3) + C_{01}(\overline{I}_2 - 3) + C_{20}(\overline{I}_1 - 3)^2 + C_{11}(\overline{I}_1 - 3)(\overline{I}_2 - 3) + C_{02}(\overline{I}_2 - 3)^2 \\ &+ \sum_{i=1}^{2} \frac{1}{D_i}(J_{e\ell} - 1)^{2i}. \end{aligned}$$

The deformation modes are characterized in terms of the principal stretches. The nominal stress-strain relations are now derived for the polynomial form with $N = 2$.

Uniaxial mode

$$\lambda_1 = \lambda_U, \quad \lambda_2 = \lambda_3 = \lambda_U^{-\frac{1}{2}}, \quad \lambda_U = 1 + \epsilon_U$$

The deviatoric strain invariants are

$$\overline{I}_1 = \lambda_U^2 + 2\lambda_U^{-1}, \qquad \overline{I}_2 = \lambda_U^{-2} + 2\lambda_U.$$

We invoke the principle of virtual work to derive the nominal stress-strain relationship,

$$\delta U = T_U \delta\lambda_U = \frac{\partial U}{\partial \lambda_U}\delta\lambda_U,$$

and it follows that

$$
\begin{aligned}
T_U =& \frac{\partial U}{\partial \lambda_U} = \frac{\partial U}{\partial \overline{I}_1}\frac{\partial \overline{I}_1}{\partial \lambda_U} + \frac{\partial U}{\partial \overline{I}_2}\frac{\partial \overline{I}_2}{\partial \lambda_U} \\
=& 2(1-\lambda_U^{-3})\left(\lambda_U \frac{\partial U}{\partial \overline{I}_1} + \frac{\partial U}{\partial \overline{I}_2}\right) \\
=& 2(1-\lambda_U^{-3})\Big[C_{10}\lambda_U + C_{01} + 2C_{20}\lambda_U(\overline{I}_1 - 3) + C_{11}\big(\overline{I}_1 - 3 + \lambda_U(\overline{I}_2 - 3)\big) + 2C_{02}(\overline{I}_2 - 3)\Big].
\end{aligned}
$$

Equibiaxial mode

$$\lambda_1 = \lambda_2 = \lambda_B,\ \lambda_3 = \lambda_B^{-2}, \quad \lambda_B = 1 + \epsilon_B$$

The deviatoric strain invariants are

$$\overline{I}_1 = 2\lambda_B^2 + \lambda_B^{-4}, \qquad \overline{I}_2 = 2\lambda_B^{-2} + \lambda_B^4.$$

From virtual work

$$\delta U = 2T_B \delta\lambda_B = \frac{\partial U}{\partial \lambda_B}\delta\lambda_B,$$

and it follows that,

$$
\begin{aligned}
T_B =& \frac{1}{2}\frac{\partial U}{\partial \lambda_B} = 2\left(\lambda_B - \lambda_B^{-5}\right)\left(\frac{\partial U}{\partial \overline{I}_1} + \lambda_B^2 \frac{\partial U}{\partial \overline{I}_2}\right) \\
=& 2(\lambda_B - \lambda_B^{-5})\Big[C_{10} + C_{01}\lambda_B^2 + 2C_{20}(\overline{I}_1 - 3) + C_{11}\Big(\lambda_B^2(\overline{I}_1 - 3) + (\overline{I}_2 - 3)\Big) \\
& + 2C_{02}\lambda_B^2(\overline{I}_2 - 3)\Big].
\end{aligned}
$$

Planar (pure shear) mode

$$\lambda_1 = \lambda_S,\ \lambda_2 = 1,\ \lambda_3 = \lambda_S^{-1}, \quad \lambda_S = 1 + \epsilon_S$$

The deviatoric strain invariants are

$$\overline{I}_1 = \overline{I}_2 = \lambda_S^2 + \lambda_S^{-2} + 1.$$

From virtual work

$$\delta U = T_S \delta\lambda_S = \frac{\partial U}{\partial \lambda_S}\delta\lambda_S,$$

and it follows that,

$$T_S = \frac{\partial U}{\partial \lambda_S} = 2(\lambda_S - \lambda_S^{-3})\left(\frac{\partial U}{\partial \overline{I}_1} + \frac{\partial U}{\partial \overline{I}_2}\right)$$
$$= 2(\lambda_S - \lambda_S^{-3})\Big[C_{10} + C_{01} + 2(C_{20} + C_{11} + C_{02})(\overline{I}_1 - 3)\Big].$$

Volumetric mode

$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_V, \quad J = \lambda_V^3$$

From virtual work

$$\delta U = -p\,\delta J = \frac{\partial U}{\partial J}\delta J,$$

and it follows that,

$$p = -\frac{\partial U}{\partial J},$$

$$p = -\sum_{i=1}^{N} 2i\frac{1}{D_i}(J-1)^{2i-1}.$$

Stress-strain relations for the reduced polynomial strain energy potential

The hyperelastic reduced polynomial form can be fitted by Abaqus up to order $N = 6$. For $N = 3$ the reduced polynomial is identical to the Yeoh model, and for $N = 1$ the neo-Hookean model is retained; hence, the following also applies to these forms. The reduced polynomial energy potential is as follows:

$$U = \sum_{i=1}^{N} C_{i0}(\overline{I}_1 - 3)^i + \sum_{i=1}^{N} \frac{1}{D_i}(J_{e\ell} - 1)^{2i}.$$

Following the arguments in the previous section, we derive the nominal stress-strain relations for the reduced polynomial.

Uniaxial mode

$$\lambda_1 = \lambda_U, \quad \lambda_2 = \lambda_3 = \lambda_U^{-\frac{1}{2}}, \quad \lambda_U = 1 + \epsilon_U$$

$$T_U = 2(\lambda_U - \lambda_U^{-2})\sum_{i=1}^{N} iC_{i0}(\overline{I}_1 - 3)^{i-1}.$$

Equibiaxial mode

$$\lambda_1 = \lambda_2 = \lambda_B,\ \lambda_3 = \lambda_B^{-2}, \quad \lambda_B = 1 + \epsilon_B$$

$$T_B = 2(\lambda_B - \lambda_B^{-5}) \sum_{i=1}^{N} iC_{i0}(\overline{I}_1 - 3)^{i-1}.$$

Planar (pure shear) mode

$$\lambda_1 = \lambda_S,\ \lambda_2 = 1,\ \lambda_3 = \lambda_S^{-1}, \quad \lambda_S = 1 + \epsilon_S$$

$$T_S = 2(\lambda_S - \lambda_S^{-3}) \sum_{i=1}^{N} iC_{i0}(\overline{I}_1 - 3)^{i-1}.$$

Volumetric mode

$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_V, \quad J = \lambda_V^3$$

$$p = -\sum_{i=1}^{N} 2i \frac{1}{D_i}(J-1)^{2i-1}.$$

Stress-strain relations for the hyperelastic Ogden strain energy potential

The hyperelastic Ogden form can be fitted up to order $N = 6$:

$$U = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i^2}(\overline{\lambda}_1^{\alpha_i} + \overline{\lambda}_2^{\alpha_i} + \overline{\lambda}_3^{\alpha_i} - 3) + \sum_{i=1}^{N} \frac{1}{D_i}(J_{e\ell} - 1)^{2i}.$$

Following the same approach as for the polynomial form, we can derive the nominal stress-strain equations for the Ogden form.

Uniaxial mode

$$\lambda_1 = \lambda_U, \quad \lambda_2 = \lambda_3 = \lambda_U^{-\frac{1}{2}}, \quad \lambda_U = 1 + \epsilon_U$$

$$T_U = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i}(\lambda_U^{\alpha_i - 1} - \lambda_U^{-\frac{1}{2}\alpha_i - 1}).$$

Equibiaxial mode

$$\lambda_1 = \lambda_2 = \lambda_B, \quad \lambda_3 = \lambda_B^{-2}, \quad \lambda_B = 1 + \epsilon_B$$

$$T_B = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i}(\lambda_B^{\alpha_i - 1} - \lambda_B^{-2\alpha_i - 1}).$$

Planar (pure shear) mode

$$\lambda_1 = \lambda_S,\ \lambda_2 = 1,\ \lambda_3 = \lambda_S^{-1}, \quad \lambda_S = 1 + \epsilon_S$$

$$T_S = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i}(\lambda_S^{\alpha_i - 1} - \lambda_S^{-\alpha_i - 1}).$$

Volumetric mode

$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_V, \quad J = \lambda_V^3$$

$$p = -\sum_{i=1}^{N} 2i \frac{1}{D_i}(J-1)^{2i-1}.$$

Stress-strain relations for the hyperelastic Arruda-Boyce strain energy potential

The hyperelastic Arruda-Boyce potential has the following form:

$$U = \mu \sum_{i=1}^{5} \frac{C_i}{\lambda_m^{2i-2}} \left(\overline{I}_1^i - 3^i \right) + \frac{1}{D} \left(\frac{J_{e\ell}^2 - 1}{2} - \ln J_{e\ell} \right),$$

where

$$C_1 = \frac{1}{2}, \qquad C_2 = \frac{1}{20}, \qquad C_3 = \frac{11}{1050}, \qquad C_4 = \frac{19}{7000} \quad \text{and} \quad C_5 = \frac{519}{673750}.$$

Following the same approach as for the polynomial form, we can derive the nominal stress-strain equations for the Arruda-Boyce potential.

Uniaxial mode

$$\lambda_1 = \lambda_U, \quad \lambda_2 = \lambda_3 = \lambda_U^{-\frac{1}{2}}, \quad \lambda_U = 1 + \epsilon_U$$

$$T_U = 2\mu(\lambda_U - \lambda_U^{-2}) \sum_{i=1}^{5} \frac{iC_i}{\lambda_m^{2i-2}} \overline{I}_1^{i-1}.$$

Equibiaxial mode

$$\lambda_1 = \lambda_2 = \lambda_B, \; \lambda_3 = \lambda_B^{-2}, \quad \lambda_B = 1 + \epsilon_B$$

$$T_B = 2\mu(\lambda_B - \lambda_B^{-5}) \sum_{i=1}^{5} \frac{iC_i}{\lambda_m^{2i-2}} \overline{I}_1^{i-1}.$$

Planar (pure shear) mode

$$\lambda_1 = \lambda_S, \; \lambda_2 = 1, \; \lambda_3 = \lambda_S^{-1}, \quad \lambda_S = 1 + \epsilon_S$$

$$T_S = 2\mu(\lambda_S - \lambda_S^{-3}) \sum_{i=1}^{5} \frac{iC_i}{\lambda_m^{2i-2}} \overline{I}_1^{i-1}.$$

Volumetric mode

$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_V, \quad J = \lambda_V^3$$

$$p = -\frac{1}{D}\left(J - \frac{1}{J}\right).$$

Stress-strain relations for the hyperelastic Van der Waals energy potential

The hyperelastic Van der Waals potential, also known as the Kilian model, has the following form:

$$U = \mu\left\{-(\lambda_m^2 - 3)\left[\ln(1-\eta) + \eta\right] - \frac{2}{3}a\left(\frac{\tilde{I} - 3}{2}\right)^{\frac{3}{2}}\right\} + \frac{1}{D}\left(\frac{J_{e\ell}^2 - 1}{2} - \ln J_{e\ell}\right),$$

where

$$\tilde{I} = (1-\beta)\overline{I}_1 + \beta\overline{I}_2 \quad \text{and} \quad \eta = \sqrt{\frac{\tilde{I}-3}{\lambda_m^2-3}}.$$

Following the same approach as for the polynomial form, we can derive the nominal stress-strain relations for the Van der Waals form.

Uniaxial mode

$$\lambda_1 = \lambda_U, \quad \lambda_2 = \lambda_3 = \lambda_U^{-\frac{1}{2}}, \quad \lambda_U = 1 + \epsilon_U$$

$$T_U = \mu(1-\lambda_U^{-3})\left(\frac{1}{1-\eta} - a\sqrt{\frac{\tilde{I}-3}{2}}\right)\left[\lambda_U(1-\beta)+\beta\right].$$

Equibiaxial mode

$$\lambda_1 = \lambda_2 = \lambda_B,\ \lambda_3 = \lambda_B^{-2}, \quad \lambda_B = 1 + \epsilon_B$$

$$T_B = \mu(\lambda_B - \lambda_B^{-5})\left(\frac{1}{1-\eta} - a\sqrt{\frac{\tilde{I}-3}{2}}\right)\left(1-\beta+\beta\lambda_B^2\right).$$

Planar (pure shear) mode

$$\lambda_1 = \lambda_S,\ \lambda_2 = 1,\ \lambda_3 = \lambda_S^{-1}, \quad \lambda_S = 1 + \epsilon_S$$

$$T_S = \mu(\lambda_S - \lambda_S^{-3})\left(\frac{1}{1-\eta} - a\sqrt{\frac{\tilde{I}-3}{2}}\right).$$

Volumetric mode

$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_V, \quad J = \lambda_V^3$$

$$p = -\frac{1}{D}\left(J - \frac{1}{J}\right).$$

Stress-strain relations for the hyperfoam strain energy potential

The hyperfoam potential is a modified form of the Hill strain energy potential and can be fitted up to order $N = 6$:

$$U = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i^2}\left[\hat{\lambda}_1^{\alpha_i} + \hat{\lambda}_2^{\alpha_i} + \hat{\lambda}_3^{\alpha_i} - 3 + \frac{1}{\beta_i}(J_{e\ell}^{-\alpha_i\beta_i} - 1)\right].$$

The deformation modes are characterized in terms of the principal stretches and the volume ratio J. The elastomeric foams are not incompressible: $J = \lambda_1\lambda_2\lambda_3 \neq 1$. The transverse stretches λ_2 and/or λ_3 are independently specified in the test data either as individual values depending on the lateral deformations or through the definition of an effective Poisson's ratio.

Uniaxial mode

$$\lambda_1 = \lambda_U,\ \lambda_2 = \lambda_3,\ J = \lambda_U\lambda_2^2, \quad \lambda_U = 1 + \epsilon_U$$

Equibiaxial mode

$$\lambda_1 = \lambda_2 = \lambda_B,\ J = \lambda_B^2\lambda_3^2, \quad \lambda_B = 1 + \epsilon_B$$

Planar mode

$$\lambda_1 = \lambda_P,\ \lambda_2 = 1,\ J = \lambda_P\lambda_3, \quad \lambda_P = 1 + \epsilon_P$$

The common nominal stress-strain relation for the three deformation modes above is

$$T_j = \frac{\partial U}{\partial \lambda_j} = \frac{2}{\lambda_j}\sum_{i=1}^{N}\frac{\mu_i}{\alpha_i}(\lambda_j^{\alpha_i} - J^{-\alpha_i\beta_i}),$$

where T_j is the nominal stress and λ_j is the stretch in the direction of loading.

Simple shear mode

The simple shear deformation is described in terms of the deformation gradient,

$$\mathbf{F} = \begin{bmatrix} 1 & \gamma & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where γ is the shear strain. Note also that for this deformation, $J = \det(\mathbf{F}) = 1$. The nominal shear stress T_S is

$$T_S = \frac{\partial U}{\partial \gamma} = \sum_{j=1}^{2} \left\{ \frac{2\gamma}{2(\lambda_j^2 - 1) - \gamma^2} \sum_{i=1}^{N} \frac{\mu_i}{\alpha_i} (\lambda_j^{\alpha_i} - 1) \right\},$$

where λ_j are the principal stretches in the plane of shearing, related to the shear strain γ as follows:

$$\lambda_{1,2} = \sqrt{1 + \frac{\gamma^2}{2} \pm \gamma \sqrt{1 + \frac{\gamma^2}{4}}}.$$

The stretch in the direction perpendicular to the plane of shearing is $\lambda_3 = 1$.

The transverse stress T_T developed during simple shear deformation (as a result of the Poynting effect) is

$$T_T = \frac{\partial U}{\partial \varepsilon} = \sum_{j=1}^{2} \left\{ \frac{2(\lambda_j^2 - 1)}{2\lambda_j^4 - \lambda_j^2(\gamma^2 + 2)} \sum_{i=1}^{N} \frac{\mu_i}{\alpha_i} (\lambda_j^{\alpha_i} - 1) \right\}.$$

Volumetric mode

The volumetric deformation is described as

$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_V, \; J = \lambda_V^3.$$

The pressure $\boldsymbol{p}$ is related to volume ratio $\boldsymbol{J}$ through

$$p = -\frac{\partial U}{\partial J} = -\frac{2}{J} \sum_{i=1}^{N} \frac{\mu_i}{\alpha_i} (J^{\frac{1}{3}\alpha_i} - J^{-\alpha_i \beta_i}).$$

Least squares fit

Given experimental data, the material constants are determined through a least-squares-fit procedure, which minimizes the relative error in stress. For the $\boldsymbol{n}$ nominal stress–nominal strain data pairs, the relative error measure $\boldsymbol{E}$ is minimized,

$$E = \sum_{i=1}^{n} \left(1 - T_i^{\text{th}} / T_i^{\text{test}} \right)^2,$$

where T_i^{test} is a stress value from the test data and T_i^{th} comes from one of the nominal stress expressions derived above.

The polynomial potential is linear in terms of the constants C_{ij}; therefore, a linear least-squares procedure can be used. The Ogden, the Arruda-Boyce, and the Van der Waals potential are nonlinear in some of their coefficients, thus necessitating the use of a nonlinear least-squares procedure.

Linear least squares fit for the polynomial model

For the full polynomial model we can rewrite the expressions for the T_k^{th} derived above as

$$T_k^{\text{th}} = \sum_{i+j=1}^{N} C_{ij} X_{ij}(\lambda_k), \quad k = 1 \ldots n,$$

where the $X_{ij}(\lambda_k)$ are functions that depend on the stress state (uniaxial, biaxial, or planar), as explained above. $N = 1$ for the first-order polynomial (or Mooney-Rivlin form), and $N = 2$ for the second-order polynomial. To minimize the relative error, we need to set

$$\frac{\partial E}{\partial C_{ij}} = 0,$$

which leads to the following set of $M = \frac{1}{2}N(N+3)$ equations:

$$\sum_{k=1}^{n} \sum_{i+j=1}^{N} \frac{X_{ij}(\lambda_k) X_{lm}(\lambda_k)}{(T_k^{\text{test}})^2} C_{ij} = \sum_{k=1}^{n} \frac{X_{lm}(\lambda_k)}{T_k^{\text{test}}}, \quad l + m = 1 \ldots N.$$

This linear set of $\boldsymbol{M}$ equations can be solved readily to define the coefficients C_{ij}.

To fit the volumetric coefficients, one needs to solve the system of $\boldsymbol{N}$ equations

$$\sum_{k=1}^{n} \sum_{i=1}^{N} \frac{X_i(J_k) X_j(J_k)}{(p_k^{\text{test}})^2} \frac{1}{D_i} = \sum_{k=1}^{n} \frac{X_j(J_k)}{p_k^{\text{test}}}, \quad j = 1 \ldots N,$$

where

$$X_i(J_k) = -2i(J_k - 1)^{2i-1},$$

and

$$J_k = \frac{V_k}{V_0}$$

is given by the user. This system of equations can be solved readily for D_i.

Linear least squares fit for the reduced polynomial model

For the reduced polynomial model we can rewrite the expressions for T_k^{th} derived above as follows:

$$T_k^{\text{th}} = \sum_{i=1}^{N} C_{i0} X_i(\lambda_k),$$

where again the functions $X_i(\lambda_k)$ depend on the stress state and the stretch, as outlined above, and N is the order of the reduced polynomial, which can take values up to $N = 6$. The following also applies to the Yeoh and neo-Hookean forms since these models are special cases of the reduced polynomial, with $N = 3$ and $N = 1$, respectively.

Following the same arguments as for the full polynomial, we arrive at the system of N equations

$$\sum_{k=1}^{n}\sum_{i=1}^{N}\frac{X_i(\lambda_k)X_j(\lambda_k)}{(T_k^{\text{test}})^2}C_{i0} = \sum_{k=1}^{n}\frac{X_j(\lambda_k)}{T_k^{\text{test}}}, \quad j = 1\ldots N.$$

This system of equations can be solved readily for the coefficients C_{i0}. The volumetric coefficients are fitted with the same procedure as used for the general polynomial models.

Nonlinear least squares fit

The Ogden, Arruda-Boyce, and Van der Waals potentials are nonlinear in some of their coefficients; hence, a nonlinear least-squares-fit procedure is required. We use the Marquard-Levenberg algorithm in the formulation by Twizell and Ogden (1986). Let $a_i, i = 1\ldots m$ be the coefficients of these hyperelastic models, where m is the number of coefficients contributing to the deviatoric behavior. Specifically, $m = 2N$ for the Ogden model, $m = 2$ for the Arruda-Boyce model, and $m = 4$ for the Van der Waals model. The coefficients are found by iterating the equation

$$a_i^{(r+1)} = a_i^{(r)} - \sum_{j=1}^{m}\sum_{k=1}^{n}\left[P_{ik}^{(r)}P_{jk}^{(r)} + \gamma\delta_{ij}\right]^{-1}P_{jk}^{(r)}E_k^{(r)},$$

where r is the iteration count, n is the number of data points,

$$E_k = \frac{T_k^{\text{test}} - T_k^{\text{th}}}{T_k^{\text{test}}}$$

is the vector of relative errors, and

$$P_{ik} = \frac{\partial E_k}{\partial a_i} = -\frac{1}{T_k^{\text{test}}}\frac{\partial T_k^{\text{th}}}{\partial a_i}$$

is the derivative of the vector of relative errors with respect to the coefficients a_i.

For $\gamma = 0$, the Newton algorithm is obtained; for very large values of γ, the steepest descent method is obtained. Thus, the Marquard-Levenberg algorithm represents a compromise between these two approaches: the value of γ is increased if the error grows and is reduced otherwise.

Nonlinear least squares fit for the Ogden model

After initializing the $\alpha_i^{(0)}$, the parameters $\mu_i^{(0)}$ are found with a linear least squares fit. In the iterative procedure outlined above, the following derivatives are used:

$$\frac{\partial T_k^{\rm th}}{\partial \mu_i} = \frac{2}{\alpha_i}(\lambda^{\alpha_i - 1} - \lambda^{c\alpha_i - 1}),$$
$$\frac{\partial T_k^{\rm th}}{\partial \alpha_i} = -\frac{2\mu_i}{\alpha_i^2}(\lambda^{\alpha_i - 1} - \lambda^{c\alpha_i - 1}) - \frac{2\mu_i}{\alpha_i}(\lambda^{\alpha_i - 1} - c\lambda^{c\alpha_i - 1}) \ln \lambda,$$

where

$$c = \begin{cases} -\frac{1}{2}, & \text{if uniaxial;} \\ -2, & \text{if biaxial;} \\ -1, & \text{if planar.} \end{cases}$$

Nonlinear least squares fit for the Arruda-Boyce model

The Arruda-Boyce model is linear in the shear modulus μ but nonlinear in the locking stretch λ_m. The locking stretch is initialized as $\lambda_m^{(0)} = \max(7.0, 3.0 \times \lambda_{\max})$, where $\lambda_{\max}$ is the maximum stretch in the user-specified test data. Given this locking stretch, the initial shear modulus, $\mu^{(0)}$, is obtained with a linear least squares fit.

In the iterative procedure outlined above, the following derivatives are used:

$$\frac{\partial T_k^{\rm th}}{\partial \mu} = \begin{cases} 2(\lambda_U - \lambda_U^{-2}) \sum_{i=1}^{5} \frac{iC_i}{\lambda_m^{2i-2} \overline{I}_1^{i-1}}, & \text{if uniaxial;} \\ 2(\lambda_B - \lambda_B^{-5}) \sum_{i=1}^{5} \frac{iC_i}{\lambda_m^{2i-2} \overline{I}_1^{i-1}}, & \text{if biaxial;} \\ 2(\lambda_S - \lambda_S^{-3}) \sum_{i=1}^{5} \frac{iC_i}{\lambda_m^{2i-2} \overline{I}_1^{i-1}}, & \text{if planar.} \end{cases}$$

$$\frac{\partial T_k^{\rm th}}{\partial \lambda_m} = \begin{cases} 2\mu(\lambda_U - \lambda_U^{-2}) \sum_{i=2}^{5} (2-2i) \frac{iC_i}{\lambda_m^{2i-1} \overline{I}_1^{i-1}}, & \text{if uniaxial;} \\ 2\mu(\lambda_B - \lambda_B^{-5}) \sum_{i=2}^{5} (2-2i) \frac{iC_i}{\lambda_m^{2i-1} \overline{I}_1^{i-1}}, & \text{if biaxial;} \\ 2\mu(\lambda_S - \lambda_S^{-3}) \sum_{i=2}^{5} (2-2i) \frac{iC_i}{\lambda_m^{2i-1} \overline{I}_1^{i-1}}, & \text{if planar.} \end{cases}$$

Nonlinear least squares fit for the Van der Waals model

The Van der Waals model is linear in the shear modulus μ but nonlinear in the locking stretch λ_m, the global interaction parameter $\boldsymbol{a}$, and the mixture parameter β. The locking stretch is initialized as $\lambda_m^{(0)} = \max(10.0, 3.0 \times \lambda_{\max})$, where $\lambda_{\max}$ is the maximum stretch in the user-specified test data. Given this guess for the locking stretch, we make use of an expression proposed by Kilian et al. (1986) to initialize the global interaction parameter

$$a^{(0)} = \frac{(\lambda_m^{(0)})^2}{(\lambda_m^{(0)})^3 - 1}.$$

The invariant mixture parameter is initialized to $\beta^{(0)} = 0$. Given these initial values, the shear modulus, $\mu^{(0)}$, is initialized using a linear least-squares-fit procedure.

In the iterative procedure outlined above, the following derivatives are used:

$$\frac{\partial T_k^{\text{th}}}{\partial \mu} = \begin{cases} (1-\lambda_U^{-3})\left(\frac{1}{1-\eta} - a\sqrt{\frac{\tilde{I}-3}{2}}\right)\left[\lambda_U(1-\beta)+\beta\right], & \text{if uniaxial;} \\ (\lambda_B-\lambda_B^{-5})\left(\frac{1}{1-\eta} - a\sqrt{\frac{\tilde{I}-3}{2}}\right)\left(1-\beta+\beta\lambda_B^2\right), & \text{if biaxial;} \\ (\lambda_S-\lambda_S^{-3})\left(\frac{1}{1-\eta} - a\sqrt{\frac{\tilde{I}-3}{2}}\right), & \text{if planar.} \end{cases}$$

$$\frac{\partial T_k^{\text{th}}}{\partial \lambda_m} = \begin{cases} -\mu(1-\lambda_U^{-3})\frac{\eta\lambda_m}{(1-\eta)^2(\lambda_m^2-3)}\left[\lambda_U(1-\beta)+\beta\right], & \text{if uniaxial;} \\ -\mu(\lambda_B-\lambda_B^{-5})\frac{\eta\lambda_m}{(1-\eta)^2(\lambda_m^2-3)}\left(1-\beta+\beta\lambda_B^2\right), & \text{if biaxial;} \\ -\mu(\lambda_S-\lambda_S^{-3})\frac{\eta\lambda_m}{(1-\eta)^2(\lambda_m^2-3)}, & \text{if planar.} \end{cases}$$

$$\frac{\partial T_k^{\text{th}}}{\partial a} = \begin{cases} -\mu(1-\lambda_U^{-3})\sqrt{\frac{\tilde{I}-3}{2}}\left[\lambda_U(1-\beta)+\beta\right], & \text{if uniaxial;} \\ -\mu(\lambda_B-\lambda_B^{-5})\sqrt{\frac{\tilde{I}-3}{2}}\left(1-\beta+\beta\lambda_B^2\right), & \text{if biaxial;} \\ -\mu(\lambda_S-\lambda_S^{-3})\sqrt{\frac{\tilde{I}-3}{2}}, & \text{if planar.} \end{cases}$$

$$\frac{\partial T_k^{\text{th}}}{\partial \beta} = \begin{cases} 2(1-\lambda_U^{-3})\left[\frac{\partial U'}{\partial \beta}\left(\lambda_U(1-\beta)+\beta\right) + U'(1-\lambda_U)\right], & \text{if uniaxial;} \\ 2(\lambda_B-\lambda_B^{-5})\left[\frac{\partial U'}{\partial \beta}\left(1-\beta+\beta\lambda_B^2\right) + U'(\lambda_B^2-1)\right], & \text{if biaxial;} \\ 0, & \text{if planar.} \end{cases}$$

In the derivatives of $\partial T_k^{\text{th}}/\partial\beta$

$$U' = \frac{\partial U}{\partial \tilde{I}} = \frac{\mu}{2}\left(\frac{1}{1-\eta} - a\sqrt{\frac{\tilde{I}-3}{2}}\right),$$

and

$$\frac{\partial U'}{\partial \beta} = \mu\left[\frac{1}{4}\frac{1}{y-3}\frac{1}{\eta(1-\eta^2)} - \frac{a}{8}\sqrt{\frac{2}{\tilde{I}-3}}\right](\overline{I}_1 - \overline{I}_2).$$

In the planar case $\overline{I}_1 = \overline{I}_2$; hence, $\partial T_k^{\text{th}}/\partial\beta$ vanishes.

Drucker stability check

Abaqus checks the Drucker stability of the material for the first three modes of deformation described above. The Drucker stability condition requires that the change in the Kirchhoff stress $d\boldsymbol{\tau}$ following from an infinitesimal change in the logarithmic strain $d\boldsymbol{\varepsilon}$ satisfies the inequality

$$d\boldsymbol{\tau} : d\boldsymbol{\varepsilon} > 0.$$

Using $d\boldsymbol{\tau} = \mathbf{D} \cdot d\boldsymbol{\varepsilon}$, the inequality becomes

$$d\boldsymbol{\varepsilon} \cdot \mathbf{D} : d\boldsymbol{\varepsilon} > 0,$$

thus requiring the tangential material stiffness $\mathbf{D}$ to be positive definite for material stability to be satisfied.

For the isotropic elastic formulation considered here, the inequality can be represented in terms of the principal stresses and strains:

$$d\tau_1 d\varepsilon_1 + d\tau_2 d\varepsilon_2 + d\tau_3 d\varepsilon_3 > 0.$$

Polynomial form

With the incompressibility assumption for the two hyperelastic models, the Kirchhoff stress is equal to the Cauchy stress: $\boldsymbol{\tau} = \boldsymbol{\sigma}$ and, thus,

$$d\boldsymbol{\sigma} : d\boldsymbol{\varepsilon} > 0.$$

In addition, we can choose any value for the hydrostatic pressure without affecting the strains. For the stability calculation a convenient choice is $\sigma_3 = d\sigma_3 = 0$, which gives us

$$d\sigma_1 d\varepsilon_1 + d\sigma_2 d\varepsilon_2 > 0.$$

The infinitesimal strain changes are related to the changes in stretch ratios by the equations

$$d\varepsilon_1 = \frac{d\lambda_1}{\lambda_1}, \quad d\varepsilon_2 = \frac{d\lambda_2}{\lambda_2}.$$

The stresses follow from the strain energy, which in turn follow from the changes in the strain invariants or in the stretches.

The relation between changes in the stress and changes in strain are described by the matrix equation

$$\begin{Bmatrix} d\sigma_1 \\ d\sigma_2 \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \begin{Bmatrix} d\varepsilon_1 \\ d\varepsilon_2 \end{Bmatrix},$$

where

$$D_{11} = 4(\lambda_1^2 + \lambda_3^2)\left(\frac{\partial U}{\partial \bar{I}_1} + \lambda_2^2 \frac{\partial U}{\partial \bar{I}_2}\right) + 4(\lambda_1^2 - \lambda_3^2)^2 \left(\frac{\partial^2 U}{\partial \bar{I}_1^2} + 2\lambda_2^2 \frac{\partial^2 U}{\partial \bar{I}_1 \partial \bar{I}_2} + \lambda_2^4 \frac{\partial^2 U}{\partial \bar{I}_2^2}\right),$$

$$D_{22} = 4(\lambda_2^2 + \lambda_3^2)\left(\frac{\partial U}{\partial \bar{I}_1} + \lambda_1^2 \frac{\partial U}{\partial \bar{I}_2}\right) + 4(\lambda_2^2 - \lambda_3^2)^2 \left(\frac{\partial^2 U}{\partial \bar{I}_1^2} + 2\lambda_1^2 \frac{\partial^2 U}{\partial \bar{I}_1 \partial \bar{I}_2} + \lambda_1^4 \frac{\partial^2 U}{\partial \bar{I}_2^2}\right),$$

$$D_{12} = D_{21} = 4\lambda_3^2 \frac{\partial U}{\partial \bar{I}_1} + 4\lambda_3^{-2} \frac{\partial U}{\partial \bar{I}_2} + 4(\lambda_1^2 - \lambda_3^2)(\lambda_2^2 - \lambda_3^2)\left(\frac{\partial^2 U}{\partial \bar{I}_1^2} + (\lambda_1^2 + \lambda_2^2)\frac{\partial^2 U}{\partial \bar{I}_1 \partial \bar{I}_2} + \lambda_1^2 \lambda_2^2 \frac{\partial^2 U}{\partial \bar{I}_2^2}\right).$$

For material stability $\mathbf{D}$ must be positive definite; thus, it is necessary that

$$D_{11} + D_{22} > 0,$$

$$D_{11} D_{22} - D_{12} D_{21} > 0,$$

for all relevant values of λ_1, λ_2, and λ_3.

Ogden form

For the Ogden form we follow the same approach as the polynomial form. Using $\lambda_3 = \lambda_1^{-1}\lambda_2^{-1}$, we have

$$U = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i^2}(\lambda_1^{\alpha_i} + \lambda_2^{\alpha_i} + \lambda_1^{-\alpha_i}\lambda_2^{-\alpha_i} - 3),$$

$$\sigma_1 = \lambda_1 \frac{\partial U}{\partial \lambda_1} = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i}(\lambda_1^{\alpha_i} - \lambda_1^{-\alpha_i}\lambda_2^{-\alpha_i}),$$

$$\sigma_2 = \lambda_2 \frac{\partial U}{\partial \lambda_2} = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i}(\lambda_2^{\alpha_i} - \lambda_1^{-\alpha_i}\lambda_2^{-\alpha_i}),$$

and the material stiffness $\mathbf{D}$ that we check for positive definiteness is

$$\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} = \sum_{i=1}^{N} 2\mu_i \lambda_1^{-\alpha_i}\lambda_2^{-\alpha_i} \begin{bmatrix} \lambda_1^{2\alpha_i}\lambda_2^{\alpha_i} + 1 & 1 \\ 1 & \lambda_1^{\alpha_i}\lambda_2^{2\alpha_i} + 1 \end{bmatrix}.$$

Arruda-Boyce form

For positive values of the shear modulus, μ, and the locking stretch, λ_m, the Arruda-Boyce form is always stable. Hence, it suffices to check the coefficients to determine whether the material satisfies Drucker stability.

Van der Waals form

When the Van der Waals model is employed in its admissible stretch range given by $\tilde{I} < \lambda_m^2$, its stability depends on the global interaction parameter, a, for positive values of the initial shear modulus, μ, and the locking stretch, λ_m. To verify the Drucker stability of the Van der Waals model, we can employ the equations derived for the polynomial models by making use of the fact that

$$\frac{\partial U}{\partial \bar{I}_1} = (1-\beta)\frac{\partial U}{\partial \tilde{I}} \quad \text{and} \quad \frac{\partial U}{\partial \bar{I}_2} = \beta \frac{\partial U}{\partial \tilde{I}}.$$

To determine the admissible stretch range, we need to find the two positive real-valued roots neighboring $\lambda = 1$ of the equation

$$\tilde{I}(\lambda) - \lambda_m^2 = 0$$

for each of the three stress states—uniaxial, biaxial, and planar—by using a simple bisection method.

Hyperfoam

The Kirchhoff stress-strain relation for the uniaxial, biaxial, planar, and volumetric deformation modes is

$$\tau_j = \lambda_j \frac{\partial U}{\partial \lambda_j} = 2\sum_{i=1}^{N} \frac{\mu_i}{\alpha_i}\left[\lambda_j^{\alpha_i} - J^{-\alpha_i \beta_i}\right].$$

Taking the total differential of τ_j and using $d\varepsilon_j = d\lambda_j/\lambda_j$,

$$d\tau_j = 2\sum_{i=1}^{N} \mu_i \left[\lambda_j^{\alpha_i} d\varepsilon_j + \beta_i J^{-\alpha_i \beta_i}(d\varepsilon_1 + d\varepsilon_2 + d\varepsilon_3)\right], \quad j = 1,2,3.$$

Since we cannot use the incompressibility assumption, we have to use all three principal stress and strain components and a 3×3 $\mathbf{D}$ matrix,

$$\begin{Bmatrix} d\tau_1 \\ d\tau_2 \\ d\tau_3 \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \begin{Bmatrix} d\varepsilon_1 \\ d\varepsilon_2 \\ d\varepsilon_3 \end{Bmatrix}.$$

Specifically,

$$\begin{Bmatrix} d\tau_1 \\ d\tau_2 \\ d\tau_3 \end{Bmatrix} = 2\sum_{i=1}^{N} \mu_i \begin{bmatrix} \lambda_1^{\alpha_i} + A_i & A_i & A_i \\ A_i & \lambda_2^{\alpha_i} + A_i & A_i \\ A_i & A_i & \lambda_3^{\alpha_i} + A_i \end{bmatrix} \begin{Bmatrix} d\varepsilon_1 \\ d\varepsilon_2 \\ d\varepsilon_3 \end{Bmatrix},$$

where $A_i = \beta_i J^{-\alpha_i \beta_i}$,

For the simple shear case the principal stretches λ_1 and λ_2 are computed from the shear strain γ (as given in an earlier expression). Thus, the same form of equations is used in checking material stability during simple shear deformation.

For material stability (i.e., for $\mathbf{D}$ to be positive definite) the following conditions must be satisfied:

$$D_{11} + D_{22} + D_{33} > 0,$$

$$D_{11}D_{22} + D_{22}D_{33} + D_{33}D_{11} - D_{23}^2 - D_{13}^2 - D_{12}^2 > 0,$$

$$\det(\mathbf{D}) > 0.$$

References

- "Hyperelastic behavior of rubberlike materials," Section 22.5.1 of the Abaqus Analysis User's Guide
- "Hyperelastic behavior in elastomeric foams," Section 22.5.2 of the Abaqus Analysis User's Guide

4.6.3 ANISOTROPIC HYPERELASTIC MATERIAL BEHAVIOR

Products: Abaqus/Standard Abaqus/Explicit

The constitutive behavior of hyperelastic materials is discussed in "Hyperelastic material behavior," Section 4.6.1, in the context of isotropic response. However, many materials of industrial and technological interest exhibit anisotropic elastic behavior due to the presence of preferred directions in their microstructure. Examples of such materials include common engineering materials (such as fiber-reinforced composites, reinforced rubber, and wood) as well as soft biological tissues (such as those found in arterial walls and heart tissues). Under large deformations these materials exhibit highly anisotropic and nonlinear elastic behavior due to rearrangements in their microstructure, such as reorientation of the fiber directions with deformation. The simulation of these nonlinear effects requires constitutive models formulated within the framework of anisotropic hyperelasticity.

Hyperelastic materials are described in terms of a "strain energy potential," which defines the strain energy stored in the material per unit of reference volume (volume in the initial configuration) as a function of the deformation at that point in the material. Two distinct formulations are used for the representation of the strain energy potential of anisotropic hyperelastic materials: strain-based and invariant-based.

Strain-based formulation

In this case the strain energy function is expressed directly in terms of the components of a suitable strain tensor, such as the Green strain tensor (see "Strain measures," Section 1.4.2):

$$U = U(\boldsymbol{\varepsilon}^G),$$

where $\boldsymbol{\varepsilon}^G = \frac{1}{2}(\mathbf{C} - \mathbf{I})$ is Green's strain, $\mathbf{C} = \mathbf{F}^T \cdot \mathbf{F}$ is the right Cauchy-Green strain tensor, $\mathbf{F}$ is the deformation gradient, and $\mathbf{I}$ is the identity matrix. Without loss of generality, the strain energy function can be written in the form

$$U = U(\overline{\boldsymbol{\varepsilon}}^G, J), \tag{4.6.3–1}$$

where $\overline{\boldsymbol{\varepsilon}}^G = \frac{1}{2}(\overline{\mathbf{C}} - \mathbf{I})$ is the modified Green strain tensor, $\overline{C} = J^{-\frac{2}{3}}\mathbf{C}$ is the distortional part of the right Cauchy-Green strain, and $J = \det(\mathbf{F})$ is the volume change.

The underlying assumption in models based on the strain-based formulation is that the preferred materials directions are initially aligned with an orthogonal coordinate system in the reference (stress-free) configuration. These directions may become nonorthogonal only after deformation. Examples of this form of strain energy function include the generalized Fung-type form (see "Generalized Fung form," below).

From Equation 4.6.3–1 the variation of U is given as

$$\delta U = \frac{\partial U}{\partial \overline{\boldsymbol{\varepsilon}}^G} : \delta\overline{\boldsymbol{\varepsilon}}^G + \frac{\partial U}{\partial J}\delta J = \overline{\mathbf{F}} \cdot \frac{\partial U}{\partial \overline{\boldsymbol{\varepsilon}}^G} \cdot \overline{\mathbf{F}}^T : \delta\mathbf{e} + \frac{\partial U}{\partial J}\delta J. \tag{4.6.3–2}$$

Using the principle of virtual work, the variation of the strain energy potential can be written as

$$\delta W_I = \int_{V^0} J(\mathbf{S} : \delta\mathbf{e} - p\,\delta\varepsilon^{\mathrm{vol}})\,dV^0 = \int_{V^0} \delta U\,dV^0 \tag{4.6.3–3}$$

(see Equation 4.6.1–7).

For a compressible material the strain variations are arbitrary, so this equation defines the stress components for such a material as

$$\mathbf{S} = \frac{1}{J}\,\mathrm{DEV}\left[\overline{\mathbf{F}}\cdot\frac{\partial U}{\partial\overline{\boldsymbol{\varepsilon}}^G}\cdot\overline{\mathbf{F}}^T\right]$$

and

$$p = -\frac{\partial U}{\partial J}.$$

When the material response is almost incompressible, the pure displacement formulation, in which the strain invariants are computed from the kinematic variables of the finite element model, can behave poorly. One difficulty is that from a numerical point of view the stiffness matrix is almost singular because the effective bulk modulus of the material is so large compared to its effective shear modulus, thus causing difficulties with the solution of the discretized equilibrium equations. Similarly, in Abaqus/Explicit the high bulk modulus increases the dilatational wave speed, thus reducing the stable time increment substantially. To avoid such problems, Abaqus/Standard offers a "mixed" formulation for such cases (refer to "Hyperelastic material behavior," Section 4.6.1).

Invariant-based formulation

Using the continuum theory of fiber-reinforced composites (Spencer, 1984), the strain energy function can be expressed directly in terms of the invariants of the deformation tensor and fiber directions. For example, consider a composite material that consists of an isotropic hyperelastic matrix reinforced with N families of fibers. The directions of the fibers in the reference configuration are characterized by a set of unit vectors $\mathbf{A}_\alpha$, $(\alpha = 1, ..., N)$. Assuming that the strain energy depends not only on deformation, but also on the fiber directions, the following form is postulated:

$$U = U(\mathbf{C}, \mathbf{A}_\alpha); \qquad \alpha = 1, ..., N.$$

The strain energy of the material must remain unchanged if both matrix and fibers in the reference configuration undergo a rigid body rotation. Then, following Spencer (1984), the strain energy can be expressed as an isotropic function of an irreducible set of scalar invariants that form the integrity basis of the tensor $\mathbf{C}$ and the vectors $\mathbf{A}_\alpha$:

$$U = U\big(\overline{I}_1, \overline{I}_2, J, \overline{I}_{4(\alpha\beta)}, \overline{I}_{5(\alpha\beta)}, \zeta_{\alpha\beta}\big), \qquad \alpha = 1, ..., N;\ \beta = 1, ..., \alpha, \tag{4.6.3–4}$$

where $\overline{I}_1$ and $\overline{I}_2$ are the first and second deviatoric strain invariants; J is the volume ratio (or third strain invariant); and $\overline{I}_{4(\alpha\beta)}$ and $\overline{I}_{5(\alpha\beta)}$ are the *pseudo-invariants* of $\overline{\mathbf{C}}$, $\mathbf{A}_\alpha$, and $\mathbf{A}_\beta$, defined as

$$\overline{I}_{4(\alpha\beta)} = \mathbf{A}_\alpha \cdot \overline{\mathbf{C}} \cdot \mathbf{A}_\beta, \qquad \text{and} \qquad \overline{I}_{5(\alpha\beta)} = \mathbf{A}_\alpha \cdot \overline{\mathbf{C}}^2 \cdot \mathbf{A}_\beta, \qquad \alpha = 1, ..., N;\ \beta = 1, ..., \alpha.$$

The terms $\zeta_{\alpha\beta}$ are geometrical constants (independent of deformation) equal to the cosine of the angle between the directions of any two families of fibers in the reference configuration,

$$\zeta_{\alpha\beta} = \mathbf{A}_\alpha \cdot \mathbf{A}_\beta, \qquad \alpha = 1, ..., N;\ \beta = 1, ..., \alpha.$$

Unlike in the case of the strain-based formulation, in the invariant-based formulation the fiber directions need not be orthogonal in the initial configuration. An example of the invariant-based energy function is the form proposed by Holzapfel, Gasser, and Ogden (2000) for arterial walls (see "Holzapfel-Gasser-Ogden form," below).

From Equation 4.6.3–4 the variation of U is given as

$$\delta U = \frac{\partial U}{\partial \overline{I}_1}\delta\overline{I}_1 + \frac{\partial U}{\partial \overline{I}_2}\delta\overline{I}_2 + \frac{\partial U}{\partial J}\delta J + \sum_{\alpha=1}^{N}\sum_{\beta=1}^{\alpha}\left(\frac{\partial U}{\partial \overline{I}_{4(\alpha\beta)}}\delta\overline{I}_{4(\alpha\beta)} + \frac{\partial U}{\partial \overline{I}_{5(\alpha\beta)}}\delta\overline{I}_{5(\alpha\beta)}\right). \qquad (4.6.3\text{–}5)$$

Using the principle of virtual work (Equation 4.6.3–3) and after some lengthy derivations, the stress components for a compressible material are found to be given as

$$\begin{aligned}\mathbf{S} = {} & \frac{2}{J}\,\mathrm{DEV}\left[\left(\frac{\partial U}{\partial \overline{I}_1} + \overline{I}_1\frac{\partial U}{\partial \overline{I}_2}\right)\overline{\mathbf{B}} - \frac{\partial U}{\partial \overline{I}_2}\overline{\mathbf{B}}\cdot\overline{\mathbf{B}}\right] \\ & + \sum_{\alpha=1}^{N}\sum_{\beta=1}^{\alpha}\frac{\partial U}{\partial \overline{I}_{4(\alpha\beta)}}\mathrm{DEV}(\overline{\mathbf{a}}_\alpha\overline{\mathbf{a}}_\beta + \overline{\mathbf{a}}_\beta\overline{\mathbf{a}}_\alpha) \\ & + \sum_{\alpha=1}^{N}\sum_{\beta=1}^{\alpha}\frac{\partial U}{\partial \overline{I}_{5(\alpha\beta)}}\mathrm{DEV}(\overline{\mathbf{a}}_\alpha\overline{\mathbf{a}}'_\beta + \overline{\mathbf{a}}'_\alpha\overline{\mathbf{a}}_\beta + \overline{\mathbf{a}}_\beta\overline{\mathbf{a}}'_\alpha + \overline{\mathbf{a}}'_\beta\overline{\mathbf{a}}_\alpha),\end{aligned}$$

and

$$p = -\frac{\partial U}{\partial J};$$

where $\overline{\mathbf{a}}_\alpha \overset{\text{def}}{=} \overline{\mathbf{F}}\cdot\mathbf{A}_\alpha$ and $\overline{\mathbf{a}}'_\alpha \overset{\text{def}}{=} \overline{\mathbf{B}}\cdot\mathbf{a}_\alpha$.

Particular forms of the strain energy potential

Several particular forms of the strain energy potential are available in Abaqus. The incompressible or almost incompressible models make up:

- the generalized Fung form and
- the Holzapfel-Gasser-Ogden form.

In addition, Abaqus provides a general capability to support user-defined forms of the strain energy potential via two sets of user subroutines: one for strain-based and one for invariant-based formulations.

Generalized Fung form

The generalized Fung strain energy potential in Abaqus is based on the two-dimensional exponential form proposed by Fung et al. (1979), suitably generalized to arbitrary three-dimensional states following Humphrey (1995). It has the following form:

$$U = \frac{c}{2}(e^Q - 1) + \frac{1}{D}\left(\frac{(J^{e\ell})^2 - 1}{2} - \ln J^{e\ell}\right),$$

where U is the strain energy per unit of reference volume, c and D are temperature-dependent material parameters, $J^{e\ell}$ is the elastic volume ratio, and Q is defined as

$$Q = \overline{\boldsymbol{\varepsilon}}^G : \mathbf{b} : \overline{\boldsymbol{\varepsilon}}^G = \overline{\varepsilon}^G_{ij} b_{ijkl} \overline{\varepsilon}^G_{kl},$$

where b_{ijlk} is a dimensionless symmetric fourth-order tensor of anisotropic material constants that can be temperature dependent and $\overline{\varepsilon}^G_{ij}$ are the components of the modified Green strain tensor.

The elastic volume ratio, $J^{e\ell}$, relates the total volume ratio, J, and the thermal volume ratio, J^{th}:

$$J^{e\ell} = \frac{J}{J^{th}}.$$

J^{th} is given by

$$J^{th} = (1 + \varepsilon_1^{th})(1 + \varepsilon_2^{th})(1 + \varepsilon_3^{th}),$$

where ε_i^{th} are the principal thermal expansion strains that are obtained from the temperature and the thermal expansion coefficients.

The initial deviatoric elasticity tensor, $\overline{\mathbf{D}}_0$, and bulk modulus, K_0, are given by

$$\overline{\mathbf{D}}_0 = c\mathbf{b}, \qquad K_0 = \frac{2}{D}.$$

Abaqus supports two forms of the generalized Fung model: anisotropic and orthotropic. The number of independent components b_{ijlk} that must be specified depends on the level of anisotropy of the material: 21 for the fully anisotropic case and 9 for the orthotropic case.

Holzapfel-Gasser-Ogden form

The form of the strain energy potential is based on that proposed by Holzapfel, Gasser, and Ogden (2000) and Gasser, Ogden, and Holzapfel (2006) for modeling arterial layers with distributed collagen fiber orientations:

$$U = C_{10}(\overline{I}_1 - 3) + \frac{1}{D}\left(\frac{(J^{e\ell})^2 - 1}{2} - \ln J^{e\ell}\right) + \frac{k_1}{2k_2}\sum_{\alpha=1}^{N}\left\{\exp[k_2\langle\overline{E}_\alpha\rangle^2] - 1\right\},$$

with

$$\overline{E}_\alpha \stackrel{\text{def}}{=} \kappa(\overline{I}_1 - 3) + (1 - 3\kappa)(\overline{I}_{4(\alpha\alpha)} - 1),$$

where U is the strain energy per unit of reference volume; C_{10}, D, k_1, k_2, and κ are temperature-dependent material parameters; N is the number of families of fibers ($N \leq 3$); $\overline{I}_1$ is the first invariant of $\overline{\mathbf{C}}$; $J^{e\ell}$ is the elastic volume ratio, as defined previously; and $\overline{I}_{4(\alpha\alpha)}$ are pseudo-invariants of $\overline{\mathbf{C}}$ and $\mathbf{A}_\alpha$.

The model assumes that the directions of the collagen fibers within each family are dispersed (with rotational symmetry) about a mean preferred direction. The parameter κ ($0 \leq \kappa \leq \frac{1}{3}$) describes the level of dispersion in the fiber directions. If $\rho(\Theta)$ is the orientation density function that characterizes the distribution (it represents the normalized number of fibers with orientations in the range $[\Theta, \Theta + d\Theta]$ with respect to the mean direction), the parameter κ is defined as

$$\kappa = \frac{1}{4}\int_0^\pi \rho(\Theta)\sin^3\Theta d\Theta.$$

It is also assumed that all families of fibers have the same mechanical properties and the same dispersion. When $\kappa = 0$, the fibers are perfectly aligned (no dispersion). When $\kappa = \frac{1}{3}$, the fibers are randomly distributed and the material becomes isotropic; this corresponds to a spherical orientation density function.

The strain-like quantity $\overline{E}_\alpha$ characterizes the deformation of the family of fibers with mean direction $\mathbf{A}_\alpha$. $\overline{E}_\alpha = \overline{I}_{4(\alpha\alpha)} - 1$ for perfectly aligned fibers ($\kappa = 0$), and $\overline{E}_\alpha = (\overline{I}_1 - 3)/3$ for randomly distributed fibers ($\kappa = \frac{1}{3}$).

The first two terms in the expression of the strain energy function represent the distortional and volumetric contributions of the noncollagenous isotropic ground material; and the third term represents the contributions from the different families of collagen fibers, taking into account the effects of dispersion. A basic assumption of the model is that collagen fibers can support only tension, as they would buckle under compressive loading. Thus, the anisotropic contribution in the strain energy function appears only when the strain of the fibers is positive or, equivalently, when $\overline{E}_\alpha > 0$. This condition is enforced by the term $\langle\overline{E}_\alpha\rangle$, where the operator $\langle\cdot\rangle$ stands for the Macauley bracket and is defined as $\langle x\rangle = \frac{1}{2}(|x| + x)$.

The initial deviatoric elasticity tensor, $\overline{\mathbf{D}}_0$, and bulk modulus, K_0, are given by

$$\overline{\mathbf{D}}_0 = 4C_{10}\mathfrak{I} + 2(1 - 3\kappa)^2 k_1 \sum_{\alpha=1}^{N} H(\overline{E}_\alpha)\mathbf{A}_\alpha\mathbf{A}_\alpha\mathbf{A}_\alpha\mathbf{A}_\alpha, \qquad K_0 = \frac{2}{D},$$

where $\mathfrak{I}$ is the fourth-order unit tensor and $H(x)$ is the Heaviside unit step function.

User-defined form: strain-based

Abaqus also allows other forms of strain-based energy potentials to be defined via user subroutines **UANISOHYPER_STRAIN** in Abaqus/Standard and **VUANISOHYPER_STRAIN** in Abaqus/Explicit by programming the first and second derivatives of the strain energy potential with respect to the components of the modified Green strain and the elastic volume ratio, J^{el}.

User-defined form: invariant-based

Abaqus also allows other forms of invariant-based energy potentials to be defined via user subroutines **UANISOHYPER_INV** in Abaqus/Standard and **VUANISOHYPER_INV** in Abaqus/Explicit by programming the first and second derivatives of the strain energy potential with respect to each invariant.

Reference

- "Anisotropic hyperelastic behavior," Section 22.5.3 of the Abaqus Analysis User's Guide

4.7 Mullins effect and permanent set

- “Mullins effect,” Section 4.7.1
- “Permanent set,” Section 4.7.2

4.7.1 MULLINS EFFECT

Products: Abaqus/Standard Abaqus/Explicit

The Mullins effect material model provided in Abaqus is intended for modeling the phenomenon of stress softening, commonly observed in filled rubber elastomers as a result of damage associated with straining. When an elastomeric test specimen is subjected to simple tension from its virgin state, unloaded, and then reloaded, the stress required on reloading is less than that on the initial loading for stretches up to the maximum stretch achieved during the initial loading. This stress-softening phenomenon is known as the Mullins effect (Mullins, 1947).

Material behavior

The following sections describe the Mullins effect model in Abaqus in detail.

Idealized material behavior

The Mullins effect is depicted qualitatively in Figure 4.7.1–1.

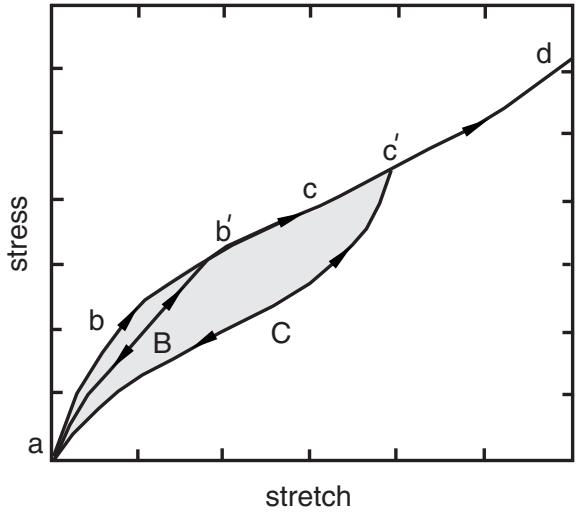


Figure 4.7.1–1 Idealized response with Mullins effect.

This figure and the accompanying description are based on work by Ogden and Roxburgh (Ogden and Roxburgh, 1999), which forms the basis of the model implemented in Abaqus. Consider the primary loading path abb' of a previously unstressed material, with loading until an arbitrary point b'. On unloading from b', the path $b'Ba$ is followed. When the material is loaded again, the softened path is retraced as aBb'. If further loading is then applied, the path $b'cc'$ is followed, where $b'cc'$ is a

continuation of the primary loading path $abb'cc'd$ (which is the path that would be followed if there was no unloading). If loading is now stopped at c', the path $c'Ca$ is followed on unloading and then retraced back to c' on reloading. If no further loading beyond c' is applied, the curve aCc' represents the subsequent material response, which is then elastic. For loading beyond c', the primary path is again followed and the pattern described is repeated. This is an ideal representation of Mullins effect. More details regarding the actual behavior and how test data that display such behavior can be used to calibrate the Abaqus model for Mullins effect are discussed in "Mullins effect," Section 22.6.1 of the Abaqus Analysis User's Guide, and in "Analysis of a solid disc with Mullins effect and permanent set," Section 3.1.7 of the Abaqus Example Problems Guide.

The loading path $abb'cc'd$ will henceforth be referred to as the "primary material response," and the constitutive behavior of the primary response can be defined using the standard energy potentials of the hyperelasticity models in Abaqus.

Stress softening is interpreted as being due to damage at the microscopic level. As the material is loaded, the damage occurs by the severing of bonds between filler particles and the rubber molecular chains. Different chain links break at different deformation levels, thereby leading to continuous damage with macroscopic deformation. An equivalent interpretation is that the energy required to cause the damage is not recoverable.

Basic framework for the model

Hyperelastic materials are described in terms of a strain energy potential function, $U(\mathbf{F})$, which defines the strain energy stored in the material per unit reference volume (volume in the initial configuration). The quantity $\mathbf{F}$ may be the deformation gradient tensor or some other appropriate strain measure. To account for Mullins effect, Ogden and Roxburgh proposed a material description that is based on an energy function of the form $U(\mathbf{F}, \eta)$, where η is a scalar variable. This scalar variable controls the material properties in the sense that it enables the material response to be governed by an energy function on unloading and subsequent submaximal reloading different from that on the primary (initial) loading path from a virgin state. Because of the interpretation and influence of η, it is no longer appropriate to think of U as the stored elastic energy potential. In fact, part of the energy is stored as strain energy, while the rest is dissipated due to damage. The shaded area in Figure 4.7.1–1 represents the energy dissipated by damage, while the unshaded portion represents the recoverable part of the strain energy. As pointed out by Ogden and Roxburgh, the inclusion of η results in the following additional equation, due to equilibrium:

$$\frac{\partial U}{\partial \eta}(\mathbf{F}, \eta) = \mathbf{0}.$$

The above equation determines the evolution of η during the course of the deformation. During a deformation process, the variable η may be either active or inactive and may switch from inactive to active or vice versa such that it always varies continously. When it is not active, the material behaves as an elastic material with strain energy $U(\mathbf{F}, \eta)$, with η held constant. However, when η is active, it is determined implicitly in terms of $\mathbf{F}$ by the preceding equation. The material again behaves as an elastic material but with a different strain energy function $U(\mathbf{F}, \eta(\mathbf{F}))$. When the parameter η switches from

being active to inactive and vice versa, the energy as well as the associated stresses remain continuous. When η is inactive, it is assumed to have a value of unity without any loss of generality.

Primary hyperelastic behavior

In preparation for writing the constitutive equations for the Mullins effect, it is useful to separate the deviatoric and the volumetric parts of the total strain energy density of the primary material response as

$$U = U_{dev} + U_{vol}.$$

In the above equation U, U_{dev}, and U_{vol} are the total, deviatoric, and volumetric parts of the strain energy density, respectively. All the hyperelasticity models in Abaqus use strain energy potential functions that are already separated into deviatoric and volumetric parts. For example, the polynomial models use a strain energy potential of the form

$$U = \sum_{i+j=1}^{N} C_{ij}(\overline{I}_1 - 3)^i(\overline{I}_2 - 3)^j + \sum_{i=1}^{N} \frac{1}{D_i}(J^{e\ell} - 1)^{2i},$$

where the symbols have the usual interpretations. The first term on the right represents the deviatoric part, and the second term represents the volumetric part of the elastic strain energy density function.

Augmented strain energy density function

The Mullins effect is accounted for by using an augmented energy function of the form

$$U(\overline{\lambda}_i, \eta) = \eta \tilde{U}_{dev}(\overline{\lambda}_i) + \phi(\eta) + \tilde{U}_{vol}(J^{e\ell}),$$

where $\tilde{U}_{dev}(\overline{\lambda}_i)$ is the deviatoric part of the strain energy density of the primary material response, defined, for example, by the first term on the right-hand-side of the polynomial strain energy function given above; $\tilde{U}_{vol}(J^{e\ell})$ is the volumetric part of the strain energy density, defined, for example, by the second term on the right-hand-side of the polynomial strain energy function given above; $\overline{\lambda}_i$ $(i = 1, 2)$ represent the deviatoric principal stretches; and $J^{e\ell}$ represents the elastic volume ratio. The function $\phi(\eta)$ is a continuous function of the damage variable η and is referred to as the "damage function." As the above expression suggests, the deviatoric part of the augmented energy function is related to the deviatoric part of the energy corresponding to the primary response by the scaling factor η. The volumetric part of the augmented energy function is the same as that for the primary response. A consequence of the above form of the augmented energy function is that the Mullins effect is associated only with the deviatoric part of the deformation. When $\eta = 1$, it is required that $\phi(\eta) = 0$, such that $U(\overline{\lambda}_i, 1) = \tilde{U}_{dev}(\overline{\lambda}_i) + \tilde{U}_{vol}(J^{e\ell})$, and the augmented energy function reduces to the strain energy density function of the primary material response. As pointed out earlier, this situation corresponds to η being inactive and physically represents the energy of a material point that is on the primary curve. The value of η varies continuously as the deformation proceeds. The above form of the energy function is an extension of the form proposed by Ogden and Roxburgh to account for material compressibility.

Stress computation

With the above modification to the energy function, the stresses, which are derived from the energy function in the usual manner, are given by

$$\boldsymbol{\sigma}(\overline{\lambda}_i, \eta, J^{e\ell}) = \eta\tilde{\mathbf{S}}(\overline{\lambda}_i) - \tilde{p}(J^{e\ell})\mathbf{I},$$

where $\tilde{\mathbf{S}}$ is the deviatoric stress corresponding to the primary material response at the current deviatoric deformation level, $\overline{\lambda}_i$, and $\tilde{p}$ is the hydrostatic pressure of the primary material response at the current volumetric deformation level, $J^{e\ell}$. Thus, the deviatoric stress as a result of Mullins effect is obtained by simply scaling the deviatoric stress of the primary material response with the damage variable η. For the above equation to predict stress softening, it is necessary that $\eta \leq 1$ on the unloading (and submaximal reloading) path. It is also assumed that $\eta > 0$, such that the deviatoric stresses remain positive until $\overline{\lambda}_1 = \overline{\lambda}_2 = 1$ is reached. The pressure stress of the augmented response is the same as that of the primary response. The model predicts unloading-reloading along a single curve (that is different, in general, from the primary loading curve) from any given strain level, that passes through the origin of the stress-strain plot. It cannot capture permanent strains upon removal of load. The model also predicts that a purely volumetric deformation will not have any damage or Mullins effect associated with it.

Damage function and damage variable

As a result of the assumed form of U and the evolution equation for η, the function $\phi(\eta)$ satisfies

$$-\phi^{'}(\eta) = \tilde{U}_{dev}(\overline{\lambda}_1, \overline{\lambda}_2),$$

which implicitly defines the damage parameter η in terms of the deformation as long as the function $\phi^{'}(\eta)$ is a monotonic function of its argument. The value of η derived from the above equation will depend on the maximum values of the principal stretches $\overline{\lambda}_1^m$ and $\overline{\lambda}_2^m$ attained on a primary loading path and also on the specific forms of the functions $\tilde{U}_{dev}(\overline{\lambda}_1, \overline{\lambda}_2)$ and $\phi(\eta)$ used. On the primary curve at any point where the unloading is initiated, $\eta = 1$, and the above equation specializes to

$$-\phi^{'}(1) = \tilde{U}_{dev}(\overline{\lambda}_1^m, \overline{\lambda}_2^m) \equiv U_{dev}^m,$$

which defines the notation U_{dev}^m. On the other hand, when the material is fully unloaded with $\overline{\lambda}_1 = \overline{\lambda}_2 = 1$ and η attains its minimum value, η_m:

$$-\phi^{'}(\eta_m) = \tilde{U}_{dev}(1, 1) = 0.$$

From the preceding two equations it follows that both the function $\phi(\eta)$ and the quantity η_m depend on the point $(\overline{\lambda}_1^m, \overline{\lambda}_2^m)$ at which unloading is initiated.

When the material is in a fully unloaded state ($\tilde{U}_{dev}(1, 1) = \tilde{U}_{vol}(1) = 0$), the augmented energy function has the residual value

$$U(1, \eta_m) = \phi(\eta_m),$$

which may be interpreted as the amount of energy dissipated by damage in the material. In a uniaxial test such as simple tension, this quantity denotes the area between the primary loading curve and the relevant unloading-reloading curve (the shaded area in Figure 4.7.1–1 for deformation until $c^{'}$). Recalling that the function ϕ depends, through U^m_{dev}, on the point at which unloading starts, the function ϕ may be constructed, following Ogden and Roxburgh, using

$$\phi^{'}(\eta) + U^m_{dev} = \phi^{'}(\eta) - \phi^{'}(1) = f(\eta, U^m_{dev}),$$

where the function $f(\eta, U^m_{dev})$ must satisfy $f(1, U^m_{dev}) = 0$ and $f(\eta_m, U^m_{dev}) = U^m_{dev}$. For a given U^m_{dev}, the above equation can be integrated to obtain $\phi(\eta)$ as

$$\phi(\eta) = \int_1^{\eta} f(\overline{\eta}, U^m_{dev}) d\overline{\eta} + (1-\eta)U^m_{dev}.$$

On the primary path $\eta = 1$, and the above equation satisfies the requirement $\phi(1) = 0$. Abaqus uses the following specific choice for $\phi^{'}(\eta)$, which has the required properties

$$\phi^{'}(\eta) = (m + \beta U^m_{dev})\mathrm{erf}^{-1}(r(1-\eta)) - U^m_{dev}.$$

The quantities r, m, and β are material constants, and erf^{-1} is the inverse of the error function, where the error function is defined as

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-w^2) dw.$$

The above choice of $\phi^{'}(\eta)$ reduces to the form proposed by Ogden and Roxburgh when $\beta = 0$.

On substituting the above form of $\phi^{'}(\eta)$ in the equation $\phi^{'}(\eta) = -\tilde{U}_{dev}(\overline{\lambda}_1, \overline{\lambda}_2)$, it can be shown that the damage variable η varies with the deformation according to

$$\eta = 1 - \frac{1}{r}\mathrm{erf}\left(\frac{U^m_{dev} - \tilde{U}_{dev}}{m + \beta U^m_{dev}}\right).$$

The above equation defines the evolution of the damage variable η with deformation $\tilde{U}_{dev}(\overline{\lambda}_1, \overline{\lambda}_2)$, the maximum value of the deviatoric strain energy density experienced by the material point during its deformation history, U^m_{dev}, and material constants r, m, and β. While the parameters r and β are dimensionless, the parameter m has the dimensions of energy.

When $\tilde{U}_{dev} = U^m_{dev}$, corresponding to a point on the primary curve, $\eta = 1.0$. On the other hand, η attains its minimum value, η_m upon removal of deformation, when $\tilde{U}_{dev} = 0$. For all intermediate values of $\tilde{U}_{dev}$, η varies monotonically between η_m and 1.0.

Material tangent stiffness

The nonlinear solution procedure in Abaqus/Standard requires the material tangent stiffness or the material Jacobian. The derivation for the material tangent stiffness follows the procedure outlined in "Hyperelastic material behavior," Section 4.6.1. As a result of the Mullins effect, the expression for $\mathbf{C}^{\mathbf{S}}$

is modified. The remaining terms are identical. To determine the form of $\mathbf{C}^{\mathbf{S}}$, it may be noted that the deviatoric stress with Mullins effect can be expressed in terms of the deviatoric stress of the primary material response and the damage variable η as

$$\mathbf{S} = \eta \tilde{\mathbf{S}}.$$

From the above equation it follows that

$$\partial \mathbf{S} = \eta \frac{\partial \tilde{\mathbf{S}}}{\partial \mathbf{e}} : \partial \mathbf{e} + \tilde{\mathbf{S}} \frac{\partial \eta}{\partial \tilde{U}_{dev}} \frac{\partial \tilde{U}_{dev}}{\partial \mathbf{e}} : \partial \mathbf{e}.$$

After some manipulations, it can be shown that

$$\frac{\partial \mathbf{S}}{\partial \mathbf{e}} = (\eta \frac{\partial \tilde{\mathbf{S}}}{\partial \mathbf{e}} + J \frac{\partial \eta}{\partial \tilde{U}_{dev}} \tilde{\mathbf{S}} \tilde{\mathbf{S}}) : \partial \mathbf{e},$$

which determines $\mathbf{C}^{\mathbf{S}}$.

Damage dissipation

As pointed out in the earlier discussion, the quantity $\phi(\eta_m)$ determines the energy dissipated due to damage. The equation for $\phi^{'}(\eta)$ can be integrated analytically through a change of variable of integration from η to $\tilde{U}_{dev}$ to yield the following expression for the function $\phi(\eta)$:

$$\phi(\eta) = \frac{(m + \beta U^m_{dev})}{r\sqrt{\pi}} [\exp(-[\frac{U^m_{dev} - \tilde{U}_{dev}}{m + \beta U^m_{dev}}])^2 - 1] + (1 - \eta) U^m_{dev}.$$

The damage dissipation can be obtained from the above expression by substituting $\eta = \eta_m$ (with η_m obtained by setting $\tilde{U}_{dev} = 0$ in the expression for η) and is given by

$$\phi(\eta_m) = [\mathrm{erf}(\frac{U^m_{dev}}{m + \beta U^m_{dev}}) - \frac{(m + \beta U^m_{dev})}{\sqrt{\pi} U^m_{dev}} (1 - \exp(-[\frac{U^m_{dev}}{m + \beta U^m_{dev}}]^2)] \frac{U^m_{dev}}{r}.$$

The recoverable part of the strain energy, U_{re}, can then be obtained by subtracting the damage dissipation from the total augmented energy

$$U_{re}(\overline{\lambda}_i, \eta) = \eta \tilde{U}_{dev}(\overline{\lambda}_i) + \phi(\eta) + \tilde{U}_{vol}(J^{e\ell}) - \phi(\eta_m).$$

When the material is fully or almost incompressible, we may assume that $\tilde{U}^m_{vol} << \tilde{U}^m_{dev}$, and the total energy corresponding to the maximum deformation along the primary curve is given by $\tilde{U}^m_{dev}$. The discussion below pertains to the above limit. From the expression for the damage dissipation, $\phi(\eta_m)$, it may be noted that when $m = 0$, the ratio of dissipation to U^m_{dev} is a constant that depends only on the material parameters r and β. In particular, this ratio is independent of $\tilde{U}^m_{dev}$, suggesting that the ratio of the dissipation to the total energy is a constant that is independent of the level of deformation. For nonzero m the above is true only when $\tilde{U}^m_{dev} >> m$, which correponds to relatively large levels of deformation. In

that case when $\tilde{U}^m_{dev} << m$, the energy dissipation is zero. Thus, the value of m (both zero and nonzero) may be thought of as providing a scale of the strain energy level for the total deformation beyond which dissipation takes place. In particular, when $m = 0$, dissipation occurs at low strain levels.

The behavior described above is depicted in Figure 4.7.1–2, which shows a plot of the normalized dissipation, $\phi(\eta_m)/U^m_{dev}$, versus the ratio U^m_{dev}/m, for nonzero m and for fixed values of the parameters $r(= 2.0)$ and $\beta(= 0.1)$. For small values of the ratio U^m_{dev}/m the dissipation tends to zero, while for larger values of the ratio, the normalized dissipation approaches a constant asymptotic value.

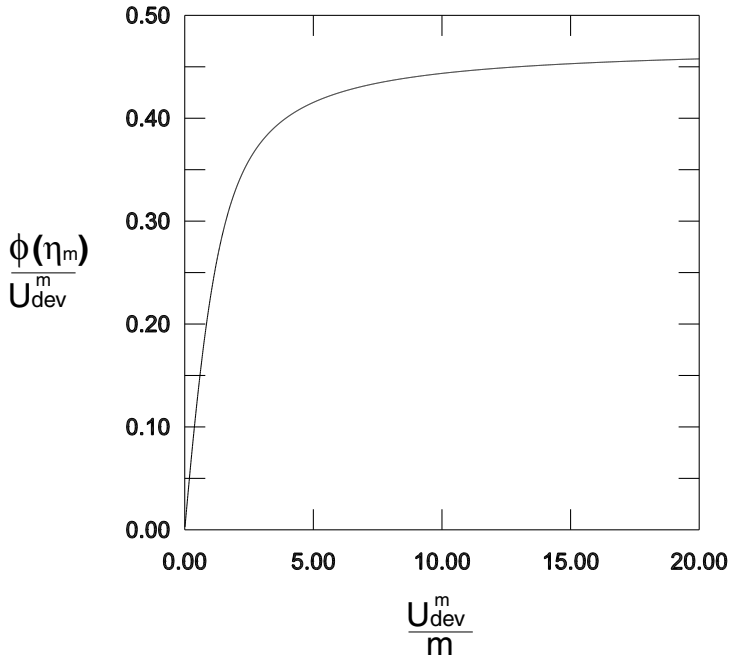


Figure 4.7.1–2 Normalized dissipation versus deformation.

Reference

- "Mullins effect," Section 22.6.1 of the Abaqus Analysis User's Guide

4.7.2 PERMANENT SET

Products: Abaqus/Standard Abaqus/Explicit

Abaqus can be used to model permanent set commonly observed in filled elastomers upon unloading after initial loading.

Approach used

Abaqus uses isotropic hardening Mises plasticity with an associated flow rule to capture permanent set. Since the underlying material is hyperelastic in nature, plasticity calculations are based on a multiplicative split of the deformation gradient into elastic and plastic components:

$$\mathbf{F} = \mathbf{F}^e \cdot \mathbf{F}^p,$$

where $\mathbf{F}^e$ is the elastic part of the deformation gradient, representing the hyperelastic behavior, and $\mathbf{F}^p$ is the plastic part of the deformation gradient, representing the stress-free intermediate configuration.

The plastic part of the deformation rate tensor based on the associated flow rule of the Mises yield condition is given by

$$\mathbf{D}^p = \frac{3}{2}\frac{\dot{\bar{\varepsilon}}^p}{q}\bar{\tau}.$$

In the above equation $\bar{\tau}$ is the deviatoric part of the Kirchoff stress tensor τ, q is the effective Kirchoff stress, and $\bar{\varepsilon}^p$ is the equivalent plastic strain. The resulting system of equations is solved using standard techniques outlined in Weber and Anand (1990) and Simo (1992).

An application that uses the above mentioned approach can be found in Govindarajan et al. (2007).

Reference

- "Permanent set in rubberlike materials," Section 23.7.1 of the Abaqus Analysis User's Guide

4.8 Viscoelasticity

- “Viscoelasticity,” Section 4.8.1
- “Finite-strain viscoelasticity,” Section 4.8.2
- “Frequency domain viscoelasticity,” Section 4.8.3

4.8.1 VISCOELASTICITY

Products: Abaqus/Standard Abaqus/Explicit

The basic hereditary integral formulation for linear isotropic viscoelasticity is

$$\boldsymbol{\sigma}(t) = \int_0^t 2G(\tau - \tau')\dot{\mathbf{e}}\, dt' + \mathbf{I}\int_0^t K(\tau - \tau')\dot{\phi}\, dt' .$$

Here $\mathbf{e}$ and ϕ are the mechanical deviatoric and volumetric strains; K is the bulk modulus and G is the shear modulus, which are functions of the reduced time τ; and $\dot{}$ denotes differentiation with respect to t'.

The reduced time is related to the actual time through the integral differential equation

$$\tau = \int_0^t \frac{dt'}{A_\theta(\theta(t'))}, \qquad \frac{d\tau}{dt} = \frac{1}{A_\theta(\theta(t))},$$

where θ is the temperature and A_θ is the shift function. (Hence, if $A_\theta = 1$, $\tau = t$.) A commonly used shift function is the Williams-Landel-Ferry (WLF) equation, which has the following form:

$$-\log A_\theta = h(\theta) = \frac{C_1^g(\theta - \theta_g)}{C_2^g + (\theta - \theta_g)},$$

where C_1^g and C_2^g are constants and θ_g is the "glass" transition temperature. This is the temperature at which, in principle, the behavior of the material changes from glassy to rubbery. If $\theta \le \theta_g - C_2^g$, deformation changes will be elastic. C_1^g and C_2^g were once thought to be "universal" constants whose values were obtained at θ_g, but these constants have been shown to vary slightly from polymer to polymer.

Abaqus allows the WLF equation to be used with any convenient temperature, other than the glass transition temperature, as the reference temperature. The form of the equation remains the same, but the constants are different. Namely,

$$-\log A_\theta = h(\theta) = \frac{C_1(\theta - \theta_0)}{C_2 + (\theta - \theta_0)},$$

where θ_0 is the reference temperature at which the relaxation data are given, and C_1 and C_2 are the calibration constants at the reference temperature. The "universal" constants C_1^g and C_2^g are related to C_1 and C_2 as follows:

$$C_1 = \frac{C_1^g}{1 + (\theta_0 - \theta_g)/C_2^g},$$
$$C_2 = C_2^g + \theta_0 - \theta_g .$$

Other forms of $h(\theta)$ are also used, such as a power series in $\theta - \theta_0$. Abaqus allows a general definition of the shift function with user subroutine **UTRS**.

The relaxation functions $K(t)$ and $G(t)$ can be defined individually in terms of a series of exponentials known as the Prony series:

$$K(\tau) = K_\infty + \sum_{i=1}^{n_K} K_i e^{-\tau/\tau_i^K} \qquad G(\tau) = G_\infty + \sum_{i=1}^{n_G} G_i e^{-\tau/\tau_i^G},$$

where K_∞ and G_∞ represent the long-term bulk and shear moduli. In general, the relaxation times τ_i^K and τ_i^G need not equal each other; however, Abaqus assumes that $\tau_i = \tau_i^K = \tau_i^G$. On the other hand, the number of terms in bulk and shear, n_K and n_G, need not equal each other. In fact, in many practical cases it can be assumed that $n_K = 0$. Hence, we now concentrate on the deviatoric behavior. The equations for the volumetric terms can be derived in an analogous way.

The deviatoric integral equation is

$$\begin{aligned}\mathbf{S} &= \int_0^t 2\left(G_\infty + \sum_{i=1}^{n_G} G_i e^{(\tau'-\tau)/\tau_i}\right)\dot{\mathbf{e}}\, dt' \\ &= \int_0^\tau 2\left(G_\infty + \sum_{i=1}^{n_G} G_i e^{(\tau'-\tau)/\tau_i}\right)\frac{d\mathbf{e}}{d\tau'}\, d\tau'.\end{aligned}$$

We rewrite this equation in the form

$$\mathbf{S} = 2G_0\left(\mathbf{e} - \sum_{i=1}^{n} \alpha_i \mathbf{e}_i\right), \tag{4.8.1–1}$$

where $G_0 = G_\infty + \sum_{i=1}^n G_i$ is the instantaneous shear modulus, $\alpha_i = G_i/G_0$ is the relative modulus of term i, and

$$\mathbf{e}_i = \int_0^\tau \left(1 - e^{(\tau'-\tau)/\tau_i}\right)\frac{d\mathbf{e}}{d\tau'}\, d\tau' \tag{4.8.1–2}$$

is the viscous (creep) strain in each term of the series. For finite element analysis this equation must be integrated over a finite increment of time. To perform this integration, we will assume that during the increment $\mathbf{e}$ varies linearly with τ; hence, $d\mathbf{e}/d\tau' = \Delta\mathbf{e}/\Delta\tau$. To use this relation, we break up Equation 4.8.1–2 into two parts:

$$\begin{aligned}\mathbf{e}_i^{n+1} = &\int_0^{\tau^n} \left(1 - e^{(\tau'-\tau^{n+1})/\tau_i}\right)\frac{d\mathbf{e}}{d\tau'}\, d\tau' \\ &+ \int_{\tau^n}^{\tau^{n+1}} \left(1 - e^{(\tau'-\tau^{n+1})/\tau_i}\right)\frac{d\mathbf{e}}{d\tau'}\, d\tau'.\end{aligned}$$

Now observe that

$$1 - e^{(\tau'-\tau^{n+1})/\tau_i} = 1 - e^{-\Delta\tau/\tau_i} + e^{-\Delta\tau/\tau_i}\left(1 - e^{(\tau'-\tau^n)/\tau_i}\right).$$

Use of this expression and the approximation for $d\mathbf{e}/d\tau'$ during the increment yields

$$\begin{aligned}\mathbf{e}_i^{n+1} = \left(1 - e^{-\Delta\tau/\tau_i}\right) \int_0^{\tau^n} \frac{d\mathbf{e}}{d\tau'}\, d\tau' \\ + e^{-\Delta\tau/\tau_i} \int_0^{\tau^n} \left(1 - e^{(\tau' - \tau^n)/\tau_i}\right) \frac{d\mathbf{e}}{d\tau'}\, d\tau' \\ + \frac{\Delta\mathbf{e}}{\Delta\tau} \int_{\tau^n}^{\tau^{n+1}} \left(1 - e^{(\tau' - \tau^{n+1})/\tau_i}\right)\, d\tau' \,.\end{aligned}$$

The first and last integrals in this expression are readily evaluated, whereas from Equation 4.8.1–2 follows that the second integral represents the viscous strain in the ith term at the beginning of the increment. Hence, the change in the ith viscous strain is

$$\begin{aligned}\Delta\mathbf{e}_i &= \left(1 - e^{-\Delta\tau/\tau_i}\right)\mathbf{e}^n + (e^{-\Delta\tau/\tau_i} - 1)\mathbf{e}_i^n + \left(\Delta\tau - \tau_i\left(1 - e^{-\Delta\tau/\tau_i}\right)\right)\frac{\Delta\mathbf{e}}{\Delta\tau} \\ &= \frac{\tau_i}{\Delta\tau}\left(\frac{\Delta\tau}{\tau_i} + e^{-\Delta\tau/\tau_i} - 1\right)\Delta\mathbf{e} + \left(1 - e^{-\Delta\tau/\tau_i}\right)(\mathbf{e}^n - \mathbf{e}_i^n)\,.\end{aligned} \tag{4.8.1–3}$$

If $\Delta\tau/\tau_i$ approaches zero, this expression can be approximated by

$$\Delta\mathbf{e}_i = \frac{\Delta\tau}{\tau_i}\left(\frac{1}{2}\Delta\mathbf{e} + \mathbf{e}^n - \mathbf{e}_i^n\right)\,. \tag{4.8.1–4}$$

The last form is used in the computations if $\Delta\tau/\tau_i < 10^{-7}$.

Hence, in an increment, Equation 4.8.1–3 or Equation 4.8.1–4 is used to calculate the new value of the viscous strains. Equation 4.8.1–1 is then used subsequently to obtain the new value of the stresses.

The tangent modulus is readily derived from these equations by differentiating the deviatoric stress increment, which is

$$\Delta\mathbf{S} = 2G_0\left(\Delta\mathbf{e} - \sum_{i=1}^{n_G}\alpha_i(\mathbf{e}_i^{n+1} - \mathbf{e}_i^n)\right)$$

with respect to the deviatoric strain increment $\Delta\mathbf{e}$. Since the equations are linear, the modulus depends only on the reduced time step:

$$G^T = \begin{cases} G_0\left[1 - \sum_{i=1}^n \alpha_i \frac{\tau_i}{\Delta\tau}\left(\frac{\Delta\tau}{\tau_i} + e^{-\Delta\tau/\tau_i} - 1\right)\right] & \text{if } \Delta\tau/\tau_i > 10^{-7} \\ G_0\left[1 - \sum_{i=1}^n \frac{1}{2}\alpha_i \frac{\Delta\tau}{\tau_i}\right] & \text{if } \Delta\tau/\tau_i < 10^{-7} \end{cases}$$

Finally, one needs a relation between the reduced time increment, $\Delta\tau$, and the actual time increment, Δt. To do this, we observe that A_θ varies very nonlinearly with temperature; hence, any direct approximation of A_θ is likely to lead to large errors. On the other hand, $h(\theta)$ will generally be a smoothly varying function of temperature that is well approximated by a linear function of temperature over an increment. If we further assume that incrementally the temperature θ is a linear function of time t, one finds the relation

$$h(\theta) = -\ln A_\theta(\theta(t)) = a + bt$$

or

$$A_\theta^{-1}(\theta(t)) = e^{a+bt}$$

with

$$a = \frac{1}{\Delta t}\left[t^{n+1}h(\theta^n) - t^n h(\theta^{n+1})\right]$$
$$b = \frac{1}{\Delta t}\left[h(\theta^{n+1}) - h(\theta^n)\right].$$

This yields the relation

$$\Delta\tau = \int_{t^n}^{t^{n+1}} e^{a+bt}\,dt$$
$$= \frac{1}{b}\left(e^{a+bt^{n+1}} - e^{a+bt^n}\right).$$

This expression can also be written as

$$\Delta\tau = \frac{A_\theta^{-1}(\theta^{n+1}) - A_\theta^{-1}(\theta^n)}{h(\theta^{n+1}) - h(\theta^n)}\Delta t\,.$$

Reduced states of stress

So far, we have discussed full triaxial stress states. If the stress state is reduced (i.e., plane stress or uniaxial stress), the equations derived here cannot be used directly because only the total stress state is reduced, not the individual terms in the series. Therefore, we use the following procedure.

For plane stress let the third component be the zero stress component. At the beginning of the increment we presumably know the volumetric elastic strain ϕ_e^n, the volumetric viscous strain ϕ_c^n, and the volumetric viscous strains ϕ_i^n associated with the Prony series. The total volumetric strain can be obtained by adding together the elastic volumetric strain and the volumetric viscous strain

$$\phi^n = \phi_e^n + \phi_c^n\,. \tag{4.8.1–5}$$

The deviatoric strain in the 3-direction follows from the relation $\phi = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$, which yields:

$$e_3^n = \varepsilon_3^n - \frac{1}{3}\phi^n = \frac{2}{3}\phi^n - \varepsilon_1^n - \varepsilon_2^n\,.$$

The out-of-plane deviatoric stress at the end of the increment is

$$s_3^{n+1} = 2G_0 \left(e_3^{n+1} - \sum_{i=1}^{n_G} \alpha_i^G e_{3i}^{n+1} \right) .$$

Substituting Equation 4.8.1–3 for e_{3i}^{n+1}, letting $e_3^{n+1} = e_3^n + \Delta e_3$, and collecting terms gives

$$\begin{aligned} s_3^{n+1} = 2G^T \Delta e_3 + 2G_0 e_3^n \left[1 - \sum_{i=1}^{n_G} \alpha_i^G (1 - e^{-\Delta\tau/\tau_i}) \right] \\ -2G_0 \sum_{i=1}^{n_G} \alpha_i^G e^{-\Delta\tau/\tau_i} e_{3i}^n . \end{aligned} \tag{4.8.1–6}$$

The hydrostatic stress is derived similarly as

$$\begin{aligned} -p^{n+1} = K^T \Delta\phi + K_0 \phi^n \left[1 - \sum_{i=1}^{n_K} \alpha_i^K (1 - e^{-\Delta\tau/\tau_i}) \right] \\ -K_0 \sum_{i=1}^{n_K} \alpha_i^K e^{-\Delta\tau/\tau_i} \phi_i^n . \end{aligned} \tag{4.8.1–7}$$

We can write these equations in the form

$$\begin{aligned} s_3^{n+1} &= 2G^T \Delta e_3 + \bar{s}_3 \\ -p^{n+1} &= K^T \Delta\phi - \bar{p} . \end{aligned}$$

In the third direction the deviatoric stress minus the hydrostatic pressure is zero; hence,

$$2G^T \Delta e_3 + K^T \Delta\phi + \bar{s}_3 - \bar{p} = 0 . \tag{4.8.1–8}$$

Since $\Delta e_3 = \frac{2}{3}\Delta\phi - \Delta\varepsilon_1 - \Delta\varepsilon_2$, it follows that

$$(K^T + \frac{4}{3} G^T)\Delta\phi = 2G^T(\Delta\varepsilon_1 + \Delta\varepsilon_2) - \bar{s}_3 + \bar{p},$$

from which $\Delta\phi$ can be solved. One can then also calculate Δe_1 and Δe_2, and with Equation 4.8.1–3 or Equation 4.8.1–4 one can update the deviatoric viscous strains $\mathbf{e}_i^{n+1}$. The volumetric strains ϕ_i^{n+1} are obtained with a relation similar to Equation 4.8.1–3.

For uniaxial stress states a similar procedure is used. As before, ϕ^n follows from Equation 4.8.1–5 and e_3^n and e_2^n follow from $\varepsilon_1 + 2\varepsilon_3 = \phi$:

$$e_3^n = e_2^n = \varepsilon_3^n - \frac{1}{3}\phi^n = \frac{1}{6}\phi^n - \frac{1}{2}\varepsilon_1^n . \tag{4.8.1–9}$$

Equation 4.8.1–6 and Equation 4.8.1–7 can be used to calculate $\bar{s}_3$ and $\bar{p}$, which again leads to Equation 4.8.1–8. Applying Equation 4.8.1–9 for Δe_3,

$$(K^T + \frac{1}{3}G^T)\Delta\phi = G^T\Delta\varepsilon_1 - \bar{s}_3 + \bar{p}.$$

After this, one can follow the same procedure as for plane stress.

Automatic time stepping procedure

To create an automatic time stepping procedure in Abaqus/Standard, we want to compare viscous strain rates at the beginning and the end of the increment. The strain rates in the individual terms at the beginning of the increment can be obtained directly by taking the limit of the incremental strain:

$$\begin{aligned}\dot{\mathbf{e}}_i^n &= \lim_{\Delta t\to 0}\frac{\mathbf{e}_i^{n+1}-\mathbf{e}_i^n}{\Delta t}\\ &= \lim_{\Delta t\to 0}\frac{\Delta\tau}{\Delta t\tau_i}\left(\frac{1}{2}\Delta\mathbf{e}+\mathbf{e}^n-\mathbf{e}_i^n\right)\\ &= \frac{\mathbf{e}^n-\mathbf{e}_i^n}{A_\theta(\theta^n)\tau_i}\,.\end{aligned}$$

A similar expression can be derived for the strain rate at the end of the increment:

$$\dot{\mathbf{e}}_i^{n+1} = \frac{\mathbf{e}^{n+1}-\mathbf{e}_i^{n+1}}{A_\theta(\theta^{n+1})\tau_i}\,.$$

If we use these expressions to calculate a difference in estimated total viscous strain increment, one finds

$$\begin{aligned}\Delta\bar{\mathbf{e}}_V &= \Delta t\sum_{i=1}^{n_G}\alpha_i^G(\dot{\mathbf{e}}_i^{n+1}-\dot{\mathbf{e}}_i^n)\\ &= \Delta t\sum_{i=1}^{n_G}\frac{\alpha_i^G}{\tau_i}\left(\frac{\mathbf{e}^{n+1}-\mathbf{e}_i^{n+1}}{A_\theta(\theta^n)}-\frac{\mathbf{e}^n-\mathbf{e}_i^n}{A_\theta(\theta^n)}\right)\,.\end{aligned}$$

This expression is readily evaluated. A similar expression can be calculated for volumetric strain $\Delta\bar{\phi}_V$, and from these two quantities a suitable scalar measure can be constructed; for example,

$$\Delta\varepsilon^{\text{est}} = \sqrt{\frac{2}{3}\Delta\bar{\mathbf{e}}_V:\Delta\bar{\mathbf{e}}_V+\frac{1}{3}\Delta\bar{\phi}_V^2}\,.$$

Comparison with the user-specified strain increment tolerance allows construction of an automatic time stepping scheme.

Reference

- "Time domain viscoelasticity," Section 22.7.1 of the Abaqus Analysis User's Guide

4.8.2 FINITE-STRAIN VISCOELASTICITY

Products: Abaqus/Standard Abaqus/Explicit

Integral formulation

The finite-strain viscoelasticity theory implemented in Abaqus is a time domain generalization of either the hyperelastic or hyperfoam constitutive models. It is assumed that the instantaneous response of the material follows from the hyperelastic constitutive equations:

$$\boldsymbol{\tau}_0(t) = \boldsymbol{\tau}_0^D(\overline{\mathbf{F}}(t)) + \boldsymbol{\tau}_0^H(J(t))$$

for a compressible material and

$$\boldsymbol{\tau}_0(t) = \boldsymbol{\tau}_0^D(\overline{\mathbf{F}}(t)) + \boldsymbol{\tau}_0^H(t)$$

for an incompressible material. In the above, $\boldsymbol{\tau}_0^D$ and $\boldsymbol{\tau}_0^H$ are, respectively, the deviatoric and the hydrostatic parts of the instantaneous Kirchhoff stress $\boldsymbol{\tau}_0$. $\overline{\mathbf{F}}$ is the "distortion gradient" related to the deformation gradient $\mathbf{F}$ by

$$\overline{\mathbf{F}} = \frac{\mathbf{F}}{J^{\frac{1}{3}}},$$

where

$$J = \det(\mathbf{F})$$

is the volume change.

Using integration by parts and a variable transformation, the basic hereditary integral formulation for linear isotropic viscoelasticity can be written in the form

$$\boldsymbol{\sigma}(t) = 2G_0\mathbf{e}(t) + \int_0^\tau 2\dot{G}(\tau')\mathbf{e}(t-t')d\tau' + \mathbf{I}\left(K_0\phi(t) + \int_0^\tau \dot{K}(\tau')\phi(t-t')d\tau'\right)$$

or entirely in terms of stress quantities,

$$\boldsymbol{\sigma}(t) = \mathbf{S}_0(t) + \int_0^\tau \frac{\dot{G}(\tau')}{G_0}\mathbf{S}_0(t-t')d\tau' + \mathbf{I}\left(p_0(t) + \int_0^\tau \frac{\dot{K}(\tau')}{K_0}p(t-t')d\tau'\right),$$

where τ is the reduced time, $\dot{G}(\tau') = dG(\tau')/d\tau'$, and $\dot{K}(\tau') = dK(\tau')/d\tau'$. G_0 and K_0 are the instantaneous small-strain shear and bulk moduli, and $G(t)$ and $K(t)$ are the time-dependent small-strain shear and bulk relaxation moduli. Recall that the reduced time represents a shift in time with temperature and is related to the actual time through the differential equation

$$d\tau' = \frac{dt'}{A_\theta(\theta(t'))},$$

where θ is the temperature and A_θ is the shift function.

Using the volumetric/deviatoric-split hereditary integral in the reference configuration for large strain (hyperelastic) materials, and then using a standard push-forward operator (see Simo, 1987), one obtains the following set of equations in the current configuration:

$$\begin{aligned}
\boldsymbol{\tau}^D(t) &= \boldsymbol{\tau}_0^D(t) + \operatorname{dev}\left[\int_0^\tau \frac{\dot{G}(\tau')}{G_0}\bar{\mathbf{F}}_t^{-1}(t-t')\cdot\boldsymbol{\tau}_0^D(t-t')\cdot\bar{\mathbf{F}}_t^{-T}(t-t')d\tau'\right], \\
\boldsymbol{\tau}^H(t) &= \boldsymbol{\tau}_0^H(t) + \int_0^\tau \frac{\dot{K}(\tau')}{K_0}\boldsymbol{\tau}_0^H(t-t')d\tau'.
\end{aligned} \tag{4.8.2–1}$$

where $\operatorname{dev}(\bullet) = (\bullet) - \frac{1}{3}(\bullet) : \mathbf{I}\,\mathbf{I}$ and $\bar{\mathbf{F}}_t(t-t')$ is the distortional deformation gradient of the state at $t-t'$ relative to the state at $\boldsymbol{t}$. A transformation is performed on the stress relating the state at time $t-t'$ to the state at time $\boldsymbol{t}$.

Implementation

As in small-strain viscoelasticity, we represent the relaxation moduli in terms of the Prony series

$$G(\tau) = G_0\left(g_\infty + \sum_{i=1}^{N_G} g_i e^{-\tau/\tau_i^G}\right), \tag{4.8.2–2}$$

$$K(\tau) = K_0\left(k_\infty + \sum_{i=1}^{N_K} k_i e^{-\tau/\tau_i^K}\right), \tag{4.8.2–3}$$

where g_i and k_i are the relative moduli of terms $\boldsymbol{i}$. Note that $g_\infty + \sum_{i=1}^{N_G} g_i = k_\infty + \sum_{i=1}^{N_K} k_i = 1$. Abaqus assumes that the relaxation times $\tau_i = \tau_i^K = \tau_i^G$ are the same so that from here on, we will sum on $\boldsymbol{N}$ terms for both bulk and shear behavior. In reality, the number of nonzero terms in bulk and shear, N_K and N_G, need not be equal, unless the instantaneous behavior is based on the hyperfoam model. In the latter case, the two deformation modes are closely related and are then assumed to relax equally and simultaneously.

Substituting Equation 4.8.2–2 and Equation 4.8.2–3 in Equation 4.8.2–1, we obtain

$$\begin{aligned}
\boldsymbol{\tau}^D(t) &= \boldsymbol{\tau}_0^D(t) - \operatorname{dev}\left[\sum_{i=1}^{N}\frac{g_i}{\tau_i}\int_0^\tau \bar{\mathbf{F}}_t^{-1}(t-t')\cdot\boldsymbol{\tau}_0^D(t-t')\cdot\bar{\mathbf{F}}_t^{-T}(t-t')e^{-\frac{\tau'}{\tau_i}}d\tau'\right], \\
\boldsymbol{\tau}^H(t) &= \boldsymbol{\tau}_0^H(t) - \sum_{i=1}^{N}\frac{k_i}{\tau_i}\int_0^\tau \boldsymbol{\tau}_0^H(t-t')e^{-\frac{\tau'}{\tau_i}}d\tau'.
\end{aligned} \tag{4.8.2–4}$$

Next, we introduce the internal stresses, associated with each term of the series

$$\boldsymbol{\tau}_i^D(t) \equiv \frac{g_i}{\tau_i}\int_0^\tau \bar{\mathbf{F}}_t^{-1}(t-t')\cdot\boldsymbol{\tau}_0^D(t-t')\cdot\bar{\mathbf{F}}_t^{-T}(t-t')e^{-\frac{\tau'}{\tau_i}}d\tau', \tag{4.8.2–5}$$

$$\boldsymbol{\tau}_i^H(t) \equiv \frac{k_i}{\tau_i}\int_0^\tau \boldsymbol{\tau}_0^H(t-t')e^{-\frac{\tau'}{\tau_i}}d\tau'. \tag{4.8.2–6}$$

These stresses are stored at each material point and are integrated forward in time. We will assume that the solution is known at time t, and we need to construct the solution at time $t+\Delta t$.

Integration of the hydrostatic stress

The internal hydrostatic stresses $\boldsymbol{\tau}_i^H$ at time $t+\Delta t$ follow from

$$\boldsymbol{\tau}_i^H(t+\Delta t) = \frac{k_i}{\tau_i}\int_0^{\tau+\Delta\tau}\boldsymbol{\tau}_0^H(t+\Delta t-t')e^{-\frac{\tau'}{\tau_i}}d\tau'.$$

With $\overline{\tau}=\tau'-\Delta\tau$ and $\overline{t}=t'-\Delta t$, it follows that

$$\begin{aligned}\boldsymbol{\tau}_i^H(t+\Delta t) =& \frac{k_i}{\tau_i}e^{-\frac{\Delta\tau}{\tau_i}}\int_{-\Delta\tau}^{\tau}\boldsymbol{\tau}_0^H(t-\overline{t})e^{-\frac{\overline{\tau}}{\tau_i}}d\overline{\tau}\\ =& \frac{k_i}{\tau_i}e^{-\frac{\Delta\tau}{\tau_i}}\int_{-\Delta\tau}^{0}\boldsymbol{\tau}_0^H(t-\overline{t})e^{-\frac{\overline{\tau}}{\tau_i}}d\overline{\tau}+\frac{k_i}{\tau_i}e^{-\frac{\Delta\tau}{\tau_i}}\int_0^{\tau}\boldsymbol{\tau}_0^H(t-\overline{t})e^{-\frac{\overline{\tau}}{\tau_i}}d\overline{\tau},\end{aligned}$$

which yields with Equation 4.8.2–6

$$\boldsymbol{\tau}_i^H(t+\Delta t) = \frac{k_i}{\tau_i}e^{-\frac{\Delta\tau}{\tau_i}}\int_{-\Delta\tau}^{0}\boldsymbol{\tau}_0^H(t-\overline{t})e^{-\frac{\overline{\tau}}{\tau_i}}d\overline{\tau}+e^{-\frac{\Delta\tau}{\tau_i}}\boldsymbol{\tau}_i^H(t). \tag{4.8.2–7}$$

To integrate the first integral in Equation 4.8.2–7, we assume that $\boldsymbol{\tau}_0^H(t-\overline{t})$ varies linearly with the reduced time $\overline{\tau}$ over the increment

$$\boldsymbol{\tau}_0^H(t-\overline{t}) = (1+\frac{\overline{\tau}}{\Delta\tau})\boldsymbol{\tau}_0^H(t)-\frac{\overline{\tau}}{\Delta\tau}\boldsymbol{\tau}_0^H(t+\Delta t)\qquad -\Delta\tau\le\overline{\tau}\le 0. \tag{4.8.2–8}$$

Substituting into Equation 4.8.2–7 yields

$$\begin{aligned}\boldsymbol{\tau}_i^H(t+\Delta t) =& \left[\frac{k_i}{\tau_i}e^{-\frac{\Delta\tau}{\tau_i}}\int_{-\Delta\tau}^{0}-\frac{\overline{\tau}}{\Delta\tau}e^{-\frac{\overline{\tau}}{\tau_i}}d\overline{\tau}\right]\boldsymbol{\tau}_0^H(t+\Delta t)\\ &+\left[\frac{k_i}{\tau_i}e^{-\frac{\Delta\tau}{\tau_i}}\int_{-\Delta\tau}^{0}(1+\frac{\overline{\tau}}{\Delta\tau})e^{-\frac{\overline{\tau}}{\tau_i}}d\overline{\tau}\right]\boldsymbol{\tau}_0^H(t)+e^{-\frac{\Delta\tau}{\tau_i}}\boldsymbol{\tau}_i^H(t).\end{aligned}$$

The integrals are readily evaluated, providing the solution at the end of the increment

$$\boldsymbol{\tau}_i^H(t+\Delta t) = \left[1 - \frac{\tau_i}{\Delta\tau}(1 - e^{-\frac{\Delta\tau}{\tau_i}})\right] k_i \boldsymbol{\tau}_0^H(t+\Delta t) + \left[\frac{\tau_i}{\Delta\tau}(1 - e^{-\frac{\Delta\tau}{\tau_i}}) - e^{-\frac{\Delta\tau}{\tau_i}}\right] k_i \boldsymbol{\tau}_0^H(t) + e^{-\frac{\Delta\tau}{\tau_i}} \boldsymbol{\tau}_i^H(t)$$

or, in a slightly different form

$$\boldsymbol{\tau}_i^H(t+\Delta t) = \alpha_i k_i \boldsymbol{\tau}_0^H(t+\Delta t) + \beta_i k_i \boldsymbol{\tau}_0^H(t) + \gamma_i \boldsymbol{\tau}_i^H(t), \tag{4.8.2–9}$$

with

$$\gamma_i = e^{-\frac{\Delta\tau}{\tau_i}}, \quad \alpha_i = 1 - \frac{\tau_i}{\Delta\tau}(1-\gamma_i), \quad \beta_i = \frac{\tau_i}{\Delta\tau}(1-\gamma_i) - \gamma_i.$$

Observe that for $\Delta t = \Delta\tau = 0$, $\gamma_i = 1$ and $\alpha_i = \beta_i = 0$. For $\Delta t = \Delta\tau = \infty$, $\alpha_i = 1$ and $\gamma_i = \beta_i = 0$.

Integration of the deviatoric stress

The internal deviatoric stresses $\boldsymbol{\tau}_i^D$ at time $t + \Delta t$ follow from

$$\boldsymbol{\tau}_i^D(t+\Delta t) = \frac{g_i}{\tau_i} \int_0^{\tau+\Delta\tau} \bar{\mathbf{F}}_{t+\Delta t}^{-1}(t+\Delta t - t') \cdot \boldsymbol{\tau}_0^D(t+\Delta t - t') \cdot \bar{\mathbf{F}}_{t+\Delta t}^{-T}(t+\Delta t - t') e^{-\frac{\tau'}{\tau_i}} d\tau'. \tag{4.8.2–10}$$

Observe that

$$\bar{\mathbf{F}}_{t+\Delta t}(t-t') = \bar{\mathbf{F}}_t(t-t') \cdot \bar{\mathbf{F}}_{t+\Delta t}(\tau),$$

and the inverse of this is

$$\bar{\mathbf{F}}_{t+\Delta t}^{-1}(t-t') = \bar{\mathbf{F}}_{t+\Delta t}^{-1}(t) \cdot \bar{\mathbf{F}}_t^{-1}(t-t') = \bar{\mathbf{F}}_t(t+\Delta t) \cdot \bar{\mathbf{F}}_t^{-1}(t-t'),$$

which—when substituted into Equation 4.8.2–10, with $\Delta\bar{\mathbf{F}} \equiv \bar{\mathbf{F}}_t(t+\Delta t)$ and $\Delta\bar{\mathbf{F}}^{-1} \equiv \bar{\mathbf{F}}_{t+\Delta t}(t)$—gives

$$\boldsymbol{\tau}_i^D(t+\Delta t) = \frac{g_i}{\tau_i} \Delta\bar{\mathbf{F}} \cdot \int_0^{\tau+\Delta\tau} \bar{\mathbf{F}}_t^{-1}(t+\Delta t - t') \cdot \boldsymbol{\tau}_0^D(t+\Delta t - t') \cdot \bar{\mathbf{F}}_t^{-T}(t+\Delta t - t') e^{-\frac{\tau'}{\tau_i}} d\tau' \cdot \Delta\bar{\mathbf{F}}^T.$$

With $\overline{\tau} = \tau' - \Delta\tau$ and $\overline{t} = t' - \Delta t$, it follows:

$$\begin{aligned}\boldsymbol{\tau}_i^D(t+\Delta t) =& \frac{g_i}{\tau_i} e^{-\frac{\Delta\tau}{\tau_i}} \Delta\bar{\mathbf{F}} \cdot \int_{-\Delta\tau}^{\tau} \bar{\mathbf{F}}_t^{-1}(t-\bar{t}) \cdot \boldsymbol{\tau}_0^D(t-\bar{t}) \cdot \bar{\mathbf{F}}_t^{-T}(t-\bar{t})\, e^{-\frac{\bar{\tau}}{\tau_i}} d\bar{\tau} \cdot \Delta\bar{\mathbf{F}}^T \\ =& \frac{g_i}{\tau_i} e^{-\frac{\Delta\tau}{\tau_i}} \int_{-\Delta\tau}^{0} \Delta\bar{\mathbf{F}} \cdot \bar{\mathbf{F}}_t^{-1}(t-\bar{t}) \cdot \boldsymbol{\tau}_0^D(t-\bar{t}) \cdot \bar{\mathbf{F}}_t^{-T}(t-\bar{t}) \cdot \Delta\bar{\mathbf{F}}^T\, e^{-\frac{\bar{\tau}}{\tau_i}} d\bar{\tau} \\ &+ \frac{g_i}{\tau_i} e^{-\frac{\Delta\tau}{\tau_i}} \Delta\bar{\mathbf{F}} \cdot \int_{0}^{\tau} \bar{\mathbf{F}}_t^{-1}(t-\bar{t}) \cdot \boldsymbol{\tau}_0^D(t-\bar{t}) \cdot \bar{\mathbf{F}}_t^{-T}(t-\bar{t})\, e^{-\frac{\bar{\tau}}{\tau_i}} d\bar{\tau} \cdot \Delta\bar{\mathbf{F}}^T. \end{aligned} \tag{4.8.2–11}$$

Now introduce the variable

$$\hat{\boldsymbol{\tau}}_0^D(t-\bar{t}) \equiv \Delta\bar{\mathbf{F}} \cdot \bar{\mathbf{F}}_t^{-1}(t-\bar{t}) \cdot \boldsymbol{\tau}_0^D(t-\bar{t}) \cdot \bar{\mathbf{F}}_t^{-T}(t-\bar{t}) \cdot \Delta\bar{\mathbf{F}}^T. \tag{4.8.2–12}$$

Note that

$$\hat{\boldsymbol{\tau}}_0^D(t) = \Delta\bar{\mathbf{F}} \cdot \bar{\mathbf{F}}_t^{-1}(t) \cdot \boldsymbol{\tau}_0^D(t) \cdot \bar{\mathbf{F}}_t^{-T}(t) \cdot \Delta\bar{\mathbf{F}}^T = \Delta\bar{\mathbf{F}} \cdot \boldsymbol{\tau}_0^D(t) \cdot \Delta\bar{\mathbf{F}}^T \tag{4.8.2–13}$$

and

$$\hat{\boldsymbol{\tau}}_0^D(t+\Delta t) = \Delta\bar{\mathbf{F}} \cdot \bar{\mathbf{F}}_t^{-1}(t+\Delta t) \cdot \boldsymbol{\tau}_0^D(t+\Delta t) \cdot \bar{\mathbf{F}}_t^{-T}(t+\Delta t) \cdot \Delta\bar{\mathbf{F}}^T = \boldsymbol{\tau}_0^D(t+\Delta t). \tag{4.8.2–14}$$

Then we can also introduce

$$\hat{\boldsymbol{\tau}}_i^D(t) = \Delta\bar{\mathbf{F}} \cdot \boldsymbol{\tau}_i^D(t) \cdot \Delta\bar{\mathbf{F}}^T. \tag{4.8.2–15}$$

Substitution of Equation 4.8.2–5, Equation 4.8.2–12, and Equation 4.8.2–15 into Equation 4.8.2–11 yields

$$\boldsymbol{\tau}_i^D(t+\Delta t) = \frac{g_i}{\tau_i} e^{-\frac{\Delta\tau}{\tau_i}} \int_{-\Delta\tau}^{0} \hat{\boldsymbol{\tau}}_0^D(t-\bar{t}) e^{-\frac{\bar{\tau}}{\tau_i}} d\bar{\tau} + e^{-\frac{\Delta\tau}{\tau_i}} \hat{\boldsymbol{\tau}}_i^D(t). \tag{4.8.2–16}$$

To integrate the first integral in Equation 4.8.2–16, we assume that $\hat{\boldsymbol{\tau}}_0^D(t-\bar{t})$ varies linearly with the reduced time $\bar{\tau}$ over the increment:

$$\hat{\boldsymbol{\tau}}_0^D(t-\bar{t}) = (1+\frac{\bar{\tau}}{\Delta\tau})\hat{\boldsymbol{\tau}}_0^D(t) - \frac{\bar{\tau}}{\Delta\tau}\hat{\boldsymbol{\tau}}_0^D(t+\Delta t) \qquad -\Delta\tau \le \bar{\tau} \le 0,$$

which with Equation 4.8.2–14 becomes

$$\hat{\boldsymbol{\tau}}_0^D(t-\bar{t}) = (1+\frac{\bar{\tau}}{\Delta\tau})\hat{\boldsymbol{\tau}}_0^D(t) - \frac{\bar{\tau}}{\Delta\tau}\boldsymbol{\tau}_0^D(t+\Delta t) \qquad -\Delta\tau \le \bar{\tau} \le 0. \tag{4.8.2–17}$$

Equation 4.8.2–16 and Equation 4.8.2–17 for the deviatoric stress have exactly the same form as Equation 4.8.2–7 and Equation 4.8.2–8 for the hydrostatic stress. Hence, after integration we obtain

$$\boldsymbol{\tau}_i^D(t+\Delta t) = \alpha_i g_i \boldsymbol{\tau}_0^D(t+\Delta t) + \beta_i g_i \hat{\boldsymbol{\tau}}_0^D(t) + \gamma_i \hat{\boldsymbol{\tau}}_i^D(t) \tag{4.8.2–18}$$

with

$$\gamma_i = e^{-\frac{\Delta\tau}{\tau_i}}, \quad \alpha_i = 1 - \frac{\tau_i}{\Delta\tau}(1-\gamma_i), \quad \beta_i = \frac{\tau_i}{\Delta\tau}(1-\gamma_i) - \gamma_i.$$

Equation 4.8.2–13, Equation 4.8.2–15, and Equation 4.8.2–18, thus, provide a straightforward integration scheme.

The total stress at the end of the increment becomes

$$\boldsymbol{\tau}(t+\Delta t) = \boldsymbol{\tau}_0(t+\Delta t) - \sum_{i=1}^{N} \boldsymbol{\tau}_i^D(t+\Delta t) - \sum_{i=1}^{N} \boldsymbol{\tau}_i^H(t+\Delta t), \tag{4.8.2–19}$$

which with Equation 4.8.2–9 and Equation 4.8.2–18 can also be written as

$$\begin{aligned}\boldsymbol{\tau}(t+\Delta t) = &\left(1 - \sum_{i=1}^{N} \alpha_i g_i\right) \boldsymbol{\tau}_0^D(t+\Delta t) + \left(1 - \sum_{i=1}^{N} \alpha_i k_i\right) \boldsymbol{\tau}_0^H(t+\Delta t) \\ &- \sum_{i=1}^{N} \beta_i g_i \mathrm{dev}\left(\hat{\boldsymbol{\tau}}_0^D(t)\right) - \sum_{i=1}^{N} \beta_i k_i \boldsymbol{\tau}_0^H(t) - \sum_{i=1}^{N} \gamma_i \mathrm{dev}\left(\hat{\boldsymbol{\tau}}_i^D(t)\right) - \sum_{i=1}^{N} \gamma_i \boldsymbol{\tau}_i^H(t).\end{aligned} \tag{4.8.2–20}$$

Rate equation

To solve the system of nonlinear equations generated by the constitutive equations, we need to generate the corotational constitutive rate equations. From Equation 4.8.2–20 it follows

$$\begin{aligned}\check{\boldsymbol{\tau}}(t+\Delta t) = &\left(1 - \sum_{i=1}^{N} \alpha_i g_i\right) \check{\boldsymbol{\tau}}_0^D(t+\Delta t) + \left(1 - \sum_{i=1}^{N} \alpha_i k_i\right) \check{\boldsymbol{\tau}}_0^H(t+\Delta t) \\ &- \sum_{i=1}^{N} \beta_i g_i \mathrm{dev}\left(\check{\hat{\boldsymbol{\tau}}}_0^D(t)\right) - \sum_{i=1}^{N} \gamma_i \mathrm{dev}\left(\check{\hat{\boldsymbol{\tau}}}_i^D(t)\right),\end{aligned} \tag{4.8.2–21}$$

where $\check{\boldsymbol{\tau}}(t)$ is the corotational (Jaumann) stress rate. The rate form of the above equation is used to compute the Jacobian.

Cauchy versus Kirchhoff stress

All equations have been worked out in terms of the Kirchhoff stress. However, the implementation in Abaqus uses the Cauchy stress. To transform to Cauchy stress, we use the relations

$$\mathbf{S}(t) = \boldsymbol{\tau}^D(t)/J(t),$$
$$p(t) = -\frac{1}{3}\mathbf{I} : \boldsymbol{\tau}^H(t)/J(t).$$

With $\Delta J \equiv J(t+\Delta t)/J(t)$, this allows us to write Equation 4.8.2–9, Equation 4.8.2–13, Equation 4.8.2–15, Equation 4.8.2–18, and Equation 4.8.2–19 in the following form:

$$p_i(t+\Delta t) = \alpha_i k_i p_0(t+\Delta t) + \frac{\beta_i k_i p_0(t) + \gamma_i p_i(t)}{\Delta J},$$
$$\hat{\mathbf{S}}_0(t) = \Delta\bar{\mathbf{F}} \cdot \mathbf{S}_0(t) \cdot \Delta\bar{\mathbf{F}}^T,$$
$$\hat{\mathbf{S}}_i(t) = \Delta\bar{\mathbf{F}} \cdot \mathbf{S}_i(t) \cdot \Delta\bar{\mathbf{F}}^T,$$
$$\mathbf{S}_i(t+\Delta t) = \alpha_i g_i \mathbf{S}_0(t+\Delta t) + \frac{\beta_i g_i \hat{\mathbf{S}}_0(t) + \gamma_i \hat{\mathbf{S}}_i(t)}{\Delta J},$$

$$\boldsymbol{\sigma}(t+\Delta t) = \boldsymbol{\sigma}_0(t+\Delta t) - \sum_{i=1}^{N} \operatorname{dev}\left(\mathbf{S}_i(t+\Delta t)\right) + \sum_{i=1}^{N} p_i(t+\Delta t)\mathbf{I}.$$

The virtual work and rate of virtual work equations are written with respect to the current volume. Therefore, the corotational stress rates are rates of Kirchhoff stress mapped into the current configuration and transformed in the same way as the stresses themselves.

This set of equations—combined with the expressions for α_i, β_i, and γ_i—describe the full implementation of the hyper-viscoelasticity model in a displacement formulation.

The rate equations can be written in a form similar to "Hyperelastic material behavior," Section 4.6.1. Introduce

$$\mathbf{C}_v^S = \left(1 - \sum_{i=1}^{N} \alpha_i g_i\right) \mathbf{C}_0^S$$

and

$$K_v = \left(1 - \sum_{i=1}^{N} \alpha_i k_i\right) K_0,$$

where $\mathbf{C}_0^S$ and K_0 are the instantaneous moduli, corresponding to $\mathbf{C}^S$ and K of "Hyperelastic material behavior," Section 4.6.1. Thus, all rate equations can be obtained by substitution of $\mathbf{C}_0^S$ by $\mathbf{C}_v^S$ and K_0 by K_v.

Reference

- "Time domain viscoelasticity," Section 22.7.1 of the Abaqus Analysis User's Guide

4.8.3 FREQUENCY DOMAIN VISCOELASTICITY

Product: Abaqus/Standard

Many applications of elastomers involve dynamic loading in the form of steady-state vibration, and often in such cases the dissipative losses in the material (the "viscous" part of the material's viscoelastic behavior) must be modeled to obtain useful results. In most problems of this class the structure is first preloaded statically, and this preloading generally involves large deformation of the elastomers. The response to that preloading is computed on the basis of purely elastic behavior in the elastomeric parts of the model—that is, we assume that the preloading is applied for a sufficiently long time so that any viscous response in the material has time to decay away.

The dynamic analysis problem in this case is, therefore, to investigate the dynamic, viscoelastic response about a predeformed elastic state. In some such cases we can reasonably assume that the vibration amplitude is sufficiently small that both the kinematic and material response in the dynamic phase of the problem can be treated as linear perturbations about the predeformed state. The small amplitude viscoelastic vibration capability provided in Abaqus/Standard, which is described in Morman and Nagtegaal (1983) and uses the procedure described in "Direct steady-state dynamic analysis," Section 2.6.1, is based on such a linearization. Its appropriateness to a particular application will depend on the magnitude of the vibration with respect to possible kinematic nonlinearities (the additional strains and rotations that occur during the dynamic loading must be small enough so that the linearization of the kinematics is reasonable) and with respect to possible nonlinearities in the material response, and on the particular constitutive assumptions incorporated in the viscoelastic model described in this section—in particular, the assumption of separation of prestrain and time effects described below.

In "Hyperelastic material behavior," Section 4.6.1, it is shown that the rate of change of the true (Cauchy) stress in an elastomeric material with a strain energy potential is given by

$$d(J\mathbf{S}) = J(\mathbf{C}^S : d\mathbf{e} + \mathbf{Q}\, d\varepsilon^{\mathrm{vol}} + d\boldsymbol{\Omega}\cdot\mathbf{S} - \mathbf{S}\cdot d\boldsymbol{\Omega}) \tag{4.8.3–1}$$

for the deviatoric part of the stress and

$$dp = -\mathbf{Q} : d\mathbf{e} - K\, d\varepsilon^{\mathrm{vol}} \tag{4.8.3–2}$$

for the equivalent pressure stress in a compressible material. The various quantities in these equations are defined in "Hyperelastic material behavior," Section 4.6.1. For a fully incompressible material all components of $\mathbf{Q}$ are zero and the equivalent pressure stress is defined only by the loading of the structure, so that the second equation is not applicable.

For small viscoelastic vibrations about a predeformed state we linearize the additional motions that occur during the vibration so that the differential of a quantity in Equation 4.8.3–1 and Equation 4.8.3–2 can be interpreted as the additional incremental value,

$$d(f) \to \Delta(f) \stackrel{\mathrm{def}}{=} f|_t - f|_0,$$

for any quantity f, where $f|_t$ is the current value of f at some time during the vibration and $f|_0$ is the reference value of f, that is, $f|_0$ is the value of f at the end of the static (long term) preloading, about which f is fluctuating during the vibration.

The incremental elastic constitutive behavior for small added motions defined by this interpretation of Equation 4.8.3–1 and Equation 4.8.3–2 is now generalized to include viscous dissipation as well as elastic response in the material, following Lianis (1965), to give

$$\Delta(J\mathbf{S}) = J\left(\mathbf{C}^S|_0 : \Delta\mathbf{e} + \mathbf{Q}|_0\Delta\varepsilon^{\text{vol}} + \Delta\boldsymbol{\Omega}\cdot\mathbf{S}|_0 - \mathbf{S}|_0\cdot\Delta\boldsymbol{\Omega} + \int_0^t \boldsymbol{\Phi}(|_0, t-\tau) : \dot{\mathbf{e}}(\tau)\,d\tau\right),$$

and, for a compressible material,

$$\Delta p = -\mathbf{Q}|_0 : \Delta\mathbf{e} - K|_0\Delta\varepsilon^{\text{vol}} - \int_0^t \kappa(|_0, t-\tau)\dot{\varepsilon}^{\text{vol}}(\tau)\,d\tau.$$

In these expressions $f(|_0, t-\tau)$ is meant to indicate that f depends on the elastic predeformation that has occurred prior to the small dynamic vibrations (the state at $t = 0$) and is evaluated at time $t-\tau$, between the start of the vibrations and the current time, t. $\boldsymbol{\Phi}$ and κ are the functions that define the viscous part of the material's response: the notation is intended to imply that these are functions of the elastic predeformation and time. $\dot{f} \overset{\text{def}}{=} df/dt$ is the time rate of change of a quantity.

The definitions of the viscous parts of the behavior, $\boldsymbol{\Phi}$ and κ, provided in Abaqus are simplified by assuming that this viscous behavior exhibits separation of time and prestrain effects; that is, that

$$\boldsymbol{\Phi}(|_0, t-\tau) = g(t-\tau)\mathbf{C}^S|_0$$

and

$$\kappa(|_0, t-\tau) = k(t-\tau)K|_0,$$

where $\mathbf{C}^S|_0$ and $K|_0$ are the "effective elasticity" of the material in its predeformed state, prior to the vibration. This assumption simply means that measurements of the viscous behavior during small motions of the material about a predeformed state depend only on the predeformation to the extent that the effective elasticity of the material also depends on that predeformation. There is experimental evidence that this simplification is appropriate for some practical materials (see Morman's (1979) discussion). With this assumption the definition of the viscous part of the material's behavior is reduced to finding the scalar functions of time, g and k (only g for fully incompressible materials), and the constitutive response to small perturbations is simplified to

$$\Delta(J\mathbf{S}) = J\left(\mathbf{C}^S|_0 : \left\{\Delta\mathbf{e} + \int_0^t g(t-\tau)\dot{\mathbf{e}}(\tau)\,d\tau\right\} + \mathbf{Q}|_0\Delta\varepsilon^{\text{vol}} + \Delta\boldsymbol{\Omega}\cdot\mathbf{S}|_0 - \mathbf{S}|_0\cdot\Delta\boldsymbol{\Omega}\right),$$

and, for compressible materials,

$$\Delta p = -\mathbf{Q}|_0 : \Delta\mathbf{e} - K|_0 \left(\Delta\varepsilon^{\mathrm{vol}} + \int_0^t k(t-\tau)\dot{\varepsilon}^{\mathrm{vol}}(\tau)\, d\tau \right).$$

In Abaqus this model is provided only for the direct-solution and subspace-based steady-state dynamic analysis procedures, in which we assume that the dynamic response is steady-state harmonic vibration, so that we can write

$$f|_t - f|_0 = \big(\Re(\Delta f) + i\Im(\Delta f)\big)\exp(i\omega t),$$

where $\Re(\Delta f) + i\Im(\Delta f)$ is the complex amplitude of a variable $\boldsymbol{f}$.

Defining the Fourier transforms of the viscous relaxation functions $g(t)$ and $k(t)$ as

$$\Re(g) + i\Im(g) \stackrel{\mathrm{def}}{=} \int_{-\infty}^{\infty} g(t)\exp(-i\omega t)\, dt,$$

and

$$\Re(k) + i\Im(k) \stackrel{\mathrm{def}}{=} \int_{-\infty}^{\infty} k(t)\exp(-i\omega t)\, dt,$$

allows the constitutive model to be written for such harmonic motions in the linear form

$$\begin{aligned}\Re\big(\Delta(J\mathbf{S})\big) = J\Big(&\mathbf{C}^S|_0 : \big\{\big(1-\omega\Im(g)\big)\Re(\Delta\mathbf{e}) - \omega\Re(g)\Im(\Delta\mathbf{e})\big\} \\ &+ \mathbf{Q}|_0\Re(\Delta\varepsilon^{\mathrm{vol}}) + \Re(\Delta\boldsymbol{\Omega})\cdot\mathbf{S}|_0 - \mathbf{S}|_0\cdot\Re(\Delta\boldsymbol{\Omega})\Big),\end{aligned}$$

and

$$\begin{aligned}\Im\big(\Delta(J\mathbf{S})\big) = J\Big(&\mathbf{C}^S|_0 : \big\{\omega\Re(g)\Re(\Delta\mathbf{e}) + \big(1-\omega\Im(g)\big)\Im(\Delta\mathbf{e})\big\} \\ &+ \mathbf{Q}|_0\Im(\Delta\varepsilon^{\mathrm{vol}}) + \Im(\Delta\boldsymbol{\Omega})\cdot\mathbf{S}|_0 - \mathbf{S}|_0\cdot\Im(\Delta\boldsymbol{\Omega})\Big);\end{aligned}$$

and, for compressible materials,

$$\Re(\Delta p) = -\mathbf{Q}|_0 : \Re(\Delta\mathbf{e}) - K|_0 : \big\{\big(1-\omega\Im(k)\big)\Re(\Delta\varepsilon^{\mathrm{vol}}) - \omega\Re(k)\Im(\Delta\varepsilon^{\mathrm{vol}})\big\},$$

and

$$\Im(\Delta p) = -\mathbf{Q}|_0 : \Im(\Delta\mathbf{e}) - K|_0 : \big\{\omega\Re(k)\Re(\Delta\varepsilon^{\mathrm{vol}}) + \big(1-\omega\Im(k)\big)\Im(\Delta\varepsilon^{\mathrm{vol}})\big\}.$$

The viscous behavior of the material is, thus, reduced to defining $\Re(g)$, $\Im(g)$, $\Re(k)$, and $\Im(k)$ as functions of frequency.

When the pure displacement formulation is used for a compressible material, the virtual work equation for dynamic response is

$$\delta W_I - \delta W_D - \delta W_E = 0, \tag{4.8.3–3}$$

where

$$\delta W_I = \int_V \boldsymbol{\sigma} : \delta \mathbf{D}\, dV$$

is the internal virtual work,

$$\delta W_D = -\int_{V^0} \rho^0 \delta \mathbf{u} \cdot \ddot{\mathbf{u}}\, dV^0$$

is the virtual work of the d'Alembert forces (ρ^0 is the mass density of the material in the original configuration), and

$$\delta W_E = \int_S \delta \mathbf{u} \cdot \mathbf{p}\, dS + \int_V \delta \mathbf{u} \cdot \mathbf{f}\, dV$$

is the virtual work of externally prescribed surface tractions $\mathbf{p}$ per current surface area and body forces $\mathbf{f}$ per current volume.

For the linearized perturbations considered here we recast Equation 4.8.3–3 in incremental form, giving

$$\Delta\delta W_I - \Delta\delta W_D - \Delta\delta W_E = 0, \tag{4.8.3–4}$$

where $\Delta\delta W_I$ is obtained from Equation 4.6.1–12 with the interpretation $d(f) \rightarrow \Delta(f)$;

$$\Delta\delta W_D \stackrel{\text{def}}{=} \delta W_D$$

and

$$\begin{aligned}\Delta\delta W_E = &\int_S \delta \mathbf{u} \cdot (\mathbf{P}_u \cdot \Delta\mathbf{u} + \mathbf{p}_p \Delta p)\ dS \\ &+ \int_V \delta \mathbf{u} \cdot (\mathbf{F}_u \cdot \Delta\mathbf{u} + \mathbf{f}_f \Delta f)\ dV,\end{aligned}$$

where

$$\mathbf{P}_u = \frac{1}{J_A} \frac{\partial (J_A \mathbf{p})}{\partial \mathbf{u}},$$

in which

$$J_A = \left| \frac{dA}{dA_0} \right|$$

is the ratio of current to reference surface area;

$$\mathbf{p}_p = \frac{\partial \mathbf{p}}{\partial p};$$

$$\mathbf{F}_u = \frac{1}{J} \frac{\partial (J\mathbf{f})}{\partial \mathbf{u}};$$

$$\mathbf{f}_f = \frac{\partial \mathbf{f}}{\partial f};$$

and $\boldsymbol{p}$ and $\boldsymbol{f}$ are the externally prescribed tractions, so that Δp and Δf are externally prescribed traction increments. Note that $\Delta\delta W_E$ includes terms dependent on $\Delta\mathbf{u}$: these give rise to the "load stiffness matrix" when the finite element interpolations are introduced.

When the motion is harmonic we can recast these quantities as

$$\begin{aligned}
\Delta\delta W_I = \int_V \Bigg(& \lfloor\, \Re(\delta\mathbf{e}) \quad \Im(\delta\mathbf{e}) \quad \Re(\delta\varepsilon^{\mathrm{vol}}) \quad \Im(\delta\varepsilon^{\mathrm{vol}}) \,\rfloor \\
& : \begin{bmatrix} \Re\left(\widetilde{\mathbf{C}}\right) & \Im\left(\widetilde{\mathbf{C}}\right) & \mathbf{Q}|_0 & 0 \\ \Im\left(\widetilde{\mathbf{C}}\right) & -\Re\left(\widetilde{\mathbf{C}}\right) & 0 & -\mathbf{Q}|_0 \\ \mathbf{Q}|_0 & 0 & \Re\left(\widetilde{K}\right) & \Im\left(\widetilde{K}\right) \\ 0 & -\mathbf{Q}|_0 & \Im\left(\widetilde{K}\right) & -\Re\left(\widetilde{K}\right) \end{bmatrix} : \begin{Bmatrix} \Re(\Delta\mathbf{e}) \\ \Im(\Delta\mathbf{e}) \\ \Re(\Delta\varepsilon^{\mathrm{vol}}) \\ \Im(\Delta\varepsilon^{\mathrm{vol}}) \end{Bmatrix} \\
& - \mathbf{S}_0 : \left[2\lfloor\, \Re(\delta\mathbf{e}) \quad \Im(\delta\mathbf{e}) \,\rfloor : \begin{Bmatrix} \Re(\Delta\mathbf{e}) \\ \Im(\Delta\mathbf{e}) \end{Bmatrix} - \lfloor\, \Re(\delta\mathbf{L}) \quad \Im(\delta\mathbf{L}) \,\rfloor : \begin{Bmatrix} \Re(\Delta\mathbf{L}) \\ \Im(\Delta\mathbf{L}) \end{Bmatrix} \right] \Bigg)\, dV,
\end{aligned}$$

where

$$\begin{aligned}
\Re\left(\widetilde{\mathbf{C}}\right) &= \big(1 - \omega\Im(g)\big)\mathbf{C}^S|_0, \\
\Im\left(\widetilde{\mathbf{C}}\right) &= -\omega\Re(g)\mathbf{C}^S|_0, \\
\Re\left(\widetilde{K}\right) &= \big(1 - \omega\Im(k)\big)K|_0, \\
\Im\left(\widetilde{K}\right) &= -\omega\Re(k)K|_0,
\end{aligned}$$

$$\Delta\delta W_D = -\omega^2 \int_{V^0} \rho^0 \lfloor\, \Re(\delta\mathbf{u}) \quad \Im(\delta\mathbf{u}) \,\rfloor \cdot \begin{Bmatrix} \Re(\Delta\mathbf{u}) \\ \Im(\Delta\mathbf{u}) \end{Bmatrix} dV^0,$$

and

$$\Delta\delta W_E = \int_S \lfloor\, \Re(\delta\mathbf{u}) \quad \Im(\delta\mathbf{u}) \,\rfloor \cdot \begin{bmatrix} \mathbf{P}_u & 0 \\ 0 & \mathbf{P}_u \end{bmatrix} \cdot \begin{Bmatrix} \Re(\Delta\mathbf{u}) \\ \Im(\Delta\mathbf{u}) \end{Bmatrix} dS$$
$$+ \int_V \lfloor\, \Re(\delta\mathbf{u}) \quad \Im(\delta\mathbf{u}) \,\rfloor \cdot \begin{bmatrix} \mathbf{F}_u & 0 \\ 0 & \mathbf{F}_u \end{bmatrix} \cdot \begin{Bmatrix} \Re(\Delta\mathbf{u}) \\ \Im(\Delta\mathbf{u}) \end{Bmatrix} dV$$
$$+ \int_S \lfloor\, \Re(\delta\mathbf{u}) \quad \Im(\delta\mathbf{u}) \,\rfloor \cdot \mathbf{p}_p \begin{Bmatrix} \Re(\Delta p) \\ \Im(\Delta p) \end{Bmatrix} dS + \int_V \lfloor\, \Re(\delta\mathbf{u}) \quad \Im(\delta\mathbf{u}) \,\rfloor \cdot \mathbf{f}_f \begin{Bmatrix} \Re(\Delta f) \\ \Im(\Delta f) \end{Bmatrix} dV.$$

In these expressions $\Re(\delta\mathbf{u})$ and $\Im(\delta\mathbf{u})$ are understood to be independent variations. Thus, when the finite element displacement interpolations are introduced into Equation 4.8.3–4, we obtain a linear, frequency-dependent system that can be solved at each frequency for the real and imaginary parts of the nodal degrees of freedom of the model. In like fashion, the augmented variational principles for almost incompressible behavior and for fully incompressible behavior described in "Hyperelastic material behavior," Section 4.6.1, can be used to obtain linear, frequency-dependent systems for harmonic viscoelastic vibration problems. The steady-state dynamic analysis procedure based on modal superposition cannot be used because the viscous behavior assumed does not correspond to a small amount of Rayleigh damping, which is a requirement for steady-state harmonic response based on modal superposition.

Reference

- "Frequency domain viscoelasticity," Section 22.7.2 of the Abaqus Analysis User's Guide

4.9 Hysteresis

- “Hysteresis,” Section 4.9.1

4.9.1 HYSTERESIS

Product: Abaqus/Standard

The hysteretic, elastomeric material model provided in Abaqus/Standard (Bergström and Boyce, 1997) is intended for modeling large-strain, rate-dependent behavior of elastomers. A primary aspect of such behavior is a pronounced hysteresis in stress-strain curves under cyclic loading. The material model is intended to model response under many load cycles. It does not model the "Mullins effect," which refers to the significant softening that is observed in the material response of elastomers in the first few load cycles.

The model decomposes the mechanical response of elastomers into an equilibrium network, A, corresponding to the state that is approached in long-time stress relaxation tests, and a time-dependent network, B, that captures the nonlinear rate-dependent deviation from the equilibrium state. The time dependence of the second network is assumed to be governed by the reptational motion of molecules having the ability to change conformation significantly, thereby relaxing the overall stress state.

The total stress is assumed to be the sum of the stresses in the two networks. The deformation gradient, $\mathbf{F}$, is assumed to act on both networks and is decomposed into elastic and inelastic parts in network B according to the multiplicative decomposition

$$\mathbf{F} = \mathbf{F}_B^e \cdot \mathbf{F}_B^{cr}.$$

The material model is defined completely by specifying an isotropic, hyperelastic material model that characterizes the response of network A; a stress-scaling factor, S, that defines the ratio of the stress carried by network B to that carried by network A under identical elastic stretching; a positive exponent, m, generally greater than 1, characterizing the effective stress dependence of the effective creep strain rate in network B; an exponent C, restricted to lie in $[-1, 0]$, characterizing the creep strain dependence of the effective creep strain rate in network B; a scaling constant A in the expression for the effective creep strain rate to maintain dimensional consistency in the equation; and a parameter E that helps regularize the creep strain rate in the vicinity of the undeformed state.

The mechanical response of network A under the imposed deformation gradient $\mathbf{F}$ is governed by standard isotropic hyperelasticity. The stress response of network B is dependent only on $\mathbf{F}_B^e$ and is governed by the same hyperelastic potential as network A, up to the stress-scaling factor, S. Given $\mathbf{F}$, the determination of $\mathbf{F}_B^e$ requires a constitutive specification for $\mathbf{F}_B^{cr}$, which is provided through an evolution equation for $\mathbf{F}_B^{cr}$ given by

$$\mathbf{F}_B^e \cdot \dot{\mathbf{F}}_B^{cr} \cdot \mathbf{F}_B^{cr^{-1}} \cdot \mathbf{F}_B^{e^{-1}} = \dot{\epsilon}_B^{cr} \frac{\mathbf{S}_B}{\sigma_B}, \tag{4.9.1–1}$$

where $\dot{\epsilon}_B^{cr}$ is the effective creep strain rate in network B, $\mathbf{S}_B$ is the Cauchy stress deviator tensor in network B, and $\sigma_B = \sqrt{\frac{3}{2}\mathbf{S}_B : \mathbf{S}_B}$ is the effective stress in network B. $\dot{\epsilon}_B^{cr}$ is governed by the expression

$$\dot{\epsilon}_B^{cr} = A[\lambda_B^{cr} - 1 + E]^C (\sigma_B)^m,$$

where $\lambda_B^{cr} - 1$ is the nominal creep strain in network B. λ_B^{cr}, the chain stretch in network B, is defined as

$$\lambda_B^{cr} = \sqrt{\frac{1}{3}\mathbf{I} : \mathbf{C}_B^{cr}},$$

where $\mathbf{C}_B^{cr} = \mathbf{F}_B^{cr\,T} \cdot \mathbf{F}_B^{cr}$. The correlation of the microscopic chain stretch to the principal macroscopic stretch state through the above formula comes from Arruda and Boyce (1993).

The numerical implementation of the material response of network A is identical to that for hyperelastic models in Abaqus/Standard, documented in "Hyperelastic material behavior," Section 4.6.1.

The implementation of the mechanics of network B is based on an implicit exponential approximation to the flow rule, Equation 4.9.1–1, which preserves the property of inelastic incompressibility of the flow rule exactly (Weber and Anand, 1990). The choice of the exponential integrator along with the isotropy of the stress-response function result in the principal directions of the right elastic stretch tensor being equal to those of the right stretch tensor for a "predictor" to the elastic deformation gradient. This predictor is a known quantity in the discrete integration that is dependent only on the deformation gradient at the end of the increment and the plastic deformation gradient at the end of the previous increment. The complete state at the end of the increment can be obtained from the calculation of the elastic stretches and the effective creep strain increment. All other quantities required for the state update are known. This material update formula is linearized exactly in closed form and leads to an unsymmetric tangent stiffness matrix caused by the nominal creep strain dependence of the constitutive equation for the effective creep strain increment.

Reference

- "Hysteresis in elastomers," Section 22.8.1 of the Abaqus Analysis User's Guide

5. Interface Modeling

Contact modeling 5.1
Surface interactions 5.2

5.1 Contact modeling

- “Small-sliding interaction between bodies,” Section 5.1.1
- “Finite-sliding interaction between deformable bodies,” Section 5.1.2
- “Finite-sliding interaction between a deformable and a rigid body,” Section 5.1.3

5.1.1 SMALL-SLIDING INTERACTION BETWEEN BODIES

Product: Abaqus/Standard

In Abaqus/Standard a capability is included to model small-sliding contact of two bodies with respect to each other. With this formulation the contacting surfaces can undergo only relatively small sliding relative to each other, but arbitrary rotation of the bodies is permitted. Small-sliding contact is computationally less expensive than finite-sliding contact, which is described in "Finite-sliding interaction between deformable bodies," Section 5.1.2.

The small-sliding capability can be used to model the interaction between two deformable bodies or between a deformable body and a rigid body in two and three dimensions. With this approach one surface definition provides the "master" surface and the other surface definition provides the "slave" surface. A kinematic constraint that the slave surface nodes do not penetrate the master surface is then enforced. The contacting surfaces need not have matching meshes; however, the best accuracy is obtained when the meshes are initially matching. For initially nonmatching meshes, accuracy can be improved by judiciously specifying initial adjustments to ensure that all slave nodes that should initially be in contact are located on the master surface.

The small-sliding contact capability is implemented by means of four internal contact elements designed to handle the following kinematic constraints:

1. two-dimensional contact between a slave node and a deformable master surface,
2. two-dimensional contact between a slave node and a rigid master surface,
3. three-dimensional contact between a slave node and a deformable master surface, and
4. three-dimensional contact between a slave node and a rigid master surface.

These elements are not accessible to the user, and Abaqus will automatically cover a slave surface with the appropriate element type, based on the nature of the corresponding master surface.

Identification of tangent plane based on nearest neighbor interaction

Although four internal element types are used to model the various types of small-sliding contact interactions supported by Abaqus/Standard, all four formulations are based on the notion that a given slave node always interacts with the same subset of master surface nodes. This nodal subset is initially determined by the Abaqus analysis input file processor from the undeformed model definition, thus avoiding the need to "track" the slave node during the course of the analysis. This set of nearest-neighbor nodes to the point on the master surface closest to the slave node is used to parametrize a contact plane with which the slave node will interact during the analysis. This concept is illustrated next for the case of a two-dimensional slave node interacting with a first-order master surface. This formulation can be generalized to second order as well as three-dimensional situations, but this generalization will not be discussed here.

Consider the contact interaction of three nodes—102, 103, and 104—on the slave surface with a master surface made up of first-order element faces described by nodes 1, 2, and 3, as shown in Figure 5.1.1–1.

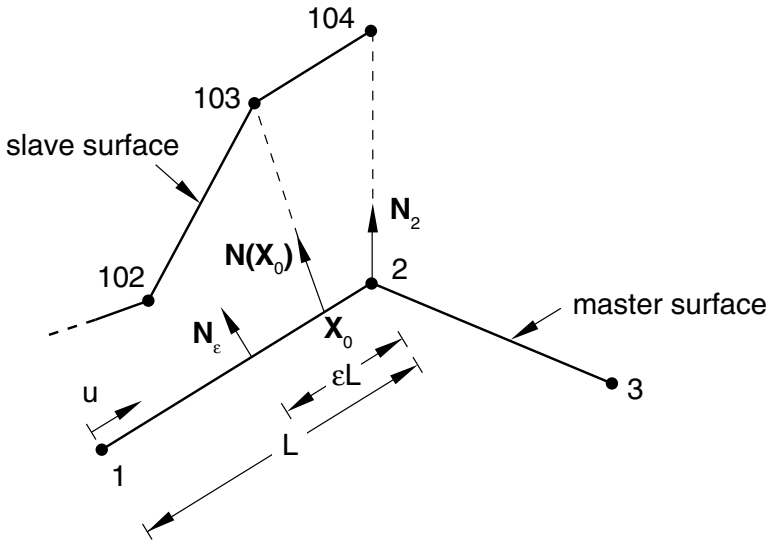


Figure 5.1.1–1 Slave nodes interacting with a two-dimensional master surface.

Before initiating the search for the nodal subset of the master surface nodes that will interact with each node on the slave surface, unit normal vectors are computed for all the nodes on the master surface. For example, the unit normal vector $\mathbf{N}_2$ is computed by averaging the unit normal vectors of segments 1–2 and 2–3. The user can also specify the normal vector for each node on the master surface. Additional unit normal vectors are computed for each segment a distance εL from each end of the segment, where ε is a fraction and L is the length of the segment; e.g., $\mathbf{N}_\varepsilon$. Currently the value of ε is set to 0.5, and the user cannot change this value. The unit normals computed are then used to define a smooth varying normal vector, $\mathbf{N}(\mathbf{X})$, at any point, $\mathbf{X}$, on the master surface.

An "anchor" point on the master surface, $\mathbf{X}_0$, is computed for each slave node so that the vector formed by the slave node and $\mathbf{X}_0$ coincides with the normal vector $\mathbf{N}(\mathbf{X}_0)$. Suppose that a search for the anchor point, $\mathbf{X}_0$, of slave node 103 reveals that $\mathbf{X}_0$ is on segment 1–2. Then, we find that

$$\mathbf{X}_0 = \mathbf{X}(u_0) = (1 - u_0)\mathbf{X}_1 + u_0\mathbf{X}_2 \,,$$

where $\mathbf{X}_1$ and $\mathbf{X}_2$ are the coordinates of nodes 1 and 2, respectively, and u_0 is calculated so that $\mathbf{X}_0 - \mathbf{X}_{103}$ coincides with $\mathbf{N}(\mathbf{X}_0)$. Moreover, the contact plane tangent direction, $\mathbf{v}_0$, at $\mathbf{X}_0$ is chosen so that it is perpendicular to $\mathbf{N}(\mathbf{X}_0)$; i.e.,

$$\mathbf{v}_0 = \mathbf{N}(\mathbf{X}_0) \times \mathbf{e}_z = \mathbf{T} \cdot \frac{\partial \mathbf{X}(u_0)}{\partial u} = \mathbf{T} \cdot (\mathbf{X}_2 - \mathbf{X}_1) \,,$$

where $\mathbf{T}$ is a (constant) rotation matrix.

The small-sliding contact constraint is achieved by requiring that slave node 103 interact with the tangent plane whose current anchor point coordinates are, at any time, given by

$$\mathbf{x}_0 = N_1(u_0)\mathbf{x}_1 + N_2(u_0)\mathbf{x}_2 \,,$$

where $N_1(u_0) = 1 - u_0$ and $N_2(u_0) = u_0$, and whose current tangent direction is given by

$$\mathbf{v} = \mathbf{T} \cdot (N_1^u(u_0)\mathbf{x}_1 + N_2^u(u_0)\mathbf{x}_2) \,,$$

where $N_1^u(u_0) = -1$ and $N_2^u(u_0) = 1$. Since the above expressions for the point $\mathbf{x}_0$ and the vector $\mathbf{v}$ resulted from barycentric (affine) combinations of the points $\mathbf{x}_1$ and $\mathbf{x}_2$—that is,

$$\begin{aligned} N_1(u_0) + N_2(u_0) &= 1 \,, \\ N_1^u(u_0) + N_2^u(u_0) &= 0 \,, \end{aligned}$$

the contact plane will be mapped properly under affine transformations such as translation, scaling (stretching), and rotation.

Next, suppose that a search for the anchor point of slave node 104 reveals that the anchor point is coincident to the master node 2. In this case the anchor point is chosen to be $\mathbf{X}_2$, or in terms of the coordinates of the three master nodes 1, 2, and 3,

$$\mathbf{X}_0 = N_1(u_0)\mathbf{X}_1 + N_2(u_0)\mathbf{X}_2 + N_3(u_0)\mathbf{X}_3 \,,$$

where $N_1(u_0) = 0$, $N_2(u_0) = 1$, and $N_3(u_0) = 0$. The contact tangent direction at $\mathbf{X}_2$ is simply

$$\mathbf{v}_0 = \mathbf{N}_2 \times \mathbf{e}_z \,.$$

However, we want to express $\mathbf{v}_0$ in terms of the coordinates of the three nodes 1, 2, and 3 to be able to track the evolution of the tangent plane. To this end, we solve for the N_i^u from the equation

$$N_1^u(u_0)\mathbf{X}_1 + N_2^u(u_0)\mathbf{X}_2 + N_3^u(u_0)\mathbf{X}_3 = \mathbf{v}_0 \,,$$

subject to the barycentric constraint

$$N_1^u(u_0) + N_2^u(u_0) + N_3^u(u_0) = 0 \,.$$

The barycentric constraint ensures that the resulting expression for the contact plane tangent direction behaves properly under affine transformations such as translations and rotations.

Contact formulation as a constrained variational principle

At each slave node that can come into contact with a master surface we construct a measure of overclosure h (penetration of the node into the master surface) and measures of relative slip s_i. These kinematic measures are then used, together with appropriate Lagrange multiplier techniques, to introduce surface

interaction theories for contact and friction, as described in "Contact pressure definition," Section 5.2.1, and "Coulomb friction," Section 5.2.3.

In two dimensions the overclosure h along the unit contact normal $\mathbf{n}$ between a slave point $\mathbf{x}_{N+1}$ and a master line $\mathbf{p}(\xi)$, where ξ parametrizes the line, is determined by finding the vector $(\mathbf{p} - \mathbf{x}_{N+1})$ from the slave node to the line that is perpendicular to the tangent vector $\mathbf{v}$ at $\mathbf{p}$. Mathematically, we express the required condition as

$$h\mathbf{n} = \mathbf{p}(\xi) - \mathbf{x}_{N+1}\,, \tag{5.1.1–1}$$

when

$$\mathbf{v} \cdot (\mathbf{p}(\xi) - \mathbf{x}_{N+1}) = 0\,.$$

Similarly, in three dimensions the overclosure h along the unit contact normal $\mathbf{n}$ between a slave point $\mathbf{x}_{N+1}$ and a master plane $\mathbf{p}(\xi_1, \xi_2)$, where ξ_i parametrize the plane, is determined by finding the vector $(\mathbf{p} - \mathbf{x}_{N+1})$ from the slave node to the plane that is perpendicular to the tangent vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ at $\mathbf{p}$. Mathematically, we express the required condition as

$$h\mathbf{n} = \mathbf{p}(\xi_1, \xi_2) - \mathbf{x}_{N+1}\,, \tag{5.1.1–2}$$

when

$$\mathbf{v}_1 \cdot (\mathbf{p}(\xi_1, \xi_2) - \mathbf{x}_{N+1}) = 0\,,$$
$$\mathbf{v}_2 \cdot (\mathbf{p}(\xi_1, \xi_2) - \mathbf{x}_{N+1}) = 0\,.$$

If at a given slave node $h < 0$, there is no contact between the surfaces at that node, and no further surface interaction calculations are needed. If $h \geq 0$, the surfaces are in contact. The contact constraint $h = 0$ is enforced by introducing a Lagrange multiplier, $\tilde{p}$, whose value provides the contact pressure at the point. To enforce the contact constraint, we need the first variation δh; and for the Newton iterations, we need the second variation, $d\delta h$. Likewise, if frictional forces are to be transmitted across the contacting surfaces, the first variations of relative slip, δs_i, and the second variations, $d\delta s_i$, are needed in the formulation. The derivation of some of these quantities is described next for all four small-sliding contact formulations.

Formulation for two-dimensional small-sliding deformable contact

For the case of two-dimensional small-sliding deformable contact, a point on the contact line associated with a slave node n_{N+1} is represented by the vector

$$\mathbf{p}(\xi) = \mathbf{x}_0 + \xi\mathbf{v}\,, \tag{5.1.1–3}$$

where, as described previously, the line's anchor point—$\mathbf{x}_0$—and its tangent vector—$\mathbf{v}$—are functions of the current master node coordinates, $\mathbf{x}_1 \ldots \mathbf{x}_N$. Hence, the vector $\mathbf{v}$ is, in general, a nonunit vector. Linearization of Equation 5.1.1–1 yields

$$\delta h\mathbf{n} + h\delta\mathbf{n} = \delta\mathbf{x}_0 + \delta\xi\mathbf{v} + \xi\delta\mathbf{v} - \delta\mathbf{u}_{N+1} \tag{5.1.1–4}$$

where $\delta\mathbf{u}_{N+1} = \delta(\mathbf{X}_{N+1} + \mathbf{u}_{N+1}) = \delta\mathbf{x}_{N+1}$, $\delta\mathbf{x}_0 = N_i(u_0)\delta\mathbf{x}_i$, and $\delta\mathbf{v} = \mathbf{R}\cdot N_i^u(u_0)\delta\mathbf{x}_i$.

Taking the dot product of Equation 5.1.1–4 with $\mathbf{n}$ results in the following expression for δh:

$$\delta h = -\mathbf{n}\cdot(\delta\mathbf{u}_{N+1} - \delta\mathbf{x}_0 - \xi\delta\mathbf{v})\,. \tag{5.1.1–5}$$

Likewise, taking the dot product of Equation 5.1.1–4 with the normalized tangent vector $\mathbf{t} = \mathbf{v}/\|\mathbf{v}\|$ and setting $h = 0$ results in the following expression for the variation in slip:

$$\delta s \overset{\text{def}}{=} \delta\xi\,\mathbf{v}\cdot\mathbf{t} = \mathbf{t}\cdot(\delta\mathbf{u}_{N+1} - \delta\mathbf{x}_0 - \xi\delta\mathbf{v})\,. \tag{5.1.1–6}$$

Suitable expressions for $d\delta h$ and $d\delta s$ can be derived by linearizing Equation 5.1.1–4 and applying the techniques highlighted above. Since the resulting expressions do not provide additional insight into the understanding of this capability, they will not be presented here.

Formulation for three-dimensional small-sliding deformable contact

The three-dimensional small-sliding deformable contact formulation is a straightforward generalization of the previous two-dimensional formulation. A point on the contact plane associated with a slave node n_{N+1} is represented by the vector

$$\mathbf{p}(\xi_1, \xi_2) = \mathbf{x}_0 + \xi_1\mathbf{v}_1 + \xi_2\mathbf{v}_2\,,$$

where the plane's anchor point—$\mathbf{x}_0$—and its two tangent vectors—$\mathbf{v}_1$ and $\mathbf{v}_2$—are functions of the current master node coordinates $\mathbf{x}_1, \ldots, \mathbf{x}_N$. Linearization of Equation 5.1.1–2 yields

$$\delta h\mathbf{n} + h\delta\mathbf{n} = \delta\mathbf{x}_0 + \delta\xi_1\mathbf{v}_1 + \xi_1\delta\mathbf{v}_1 + \delta\xi_2\mathbf{v}_2 + \xi_2\delta\mathbf{v}_2 - \delta\mathbf{u}_{N+1}\,, \tag{5.1.1–7}$$

where $\delta\mathbf{x}_0 = \mathbf{f}(\delta\mathbf{x}_1, \ldots, \delta\mathbf{x}_N)$ and $\delta\mathbf{v}_i = \mathbf{g}_i(\delta\mathbf{x}_1, \ldots, \delta\mathbf{x}_N)$.

Taking the dot product of Equation 5.1.1–7 with $\mathbf{n}$ results in the following expression for δh:

$$\delta h = -\mathbf{n}\cdot(\delta\mathbf{u}_{N+1} - \delta\mathbf{x}_0 - \xi_1\delta\mathbf{v}_1 - \xi_2\delta\mathbf{v}_2)\,. \tag{5.1.1–8}$$

Likewise, taking the dot product of Equation 5.1.1–7 with $\mathbf{t}_1 = \mathbf{v}_1/\|\mathbf{v}_1\|$ and setting $h = 0$ results in the following expression for the variation of the first slip component:

$$\delta s_1 \overset{\text{def}}{=} \delta\xi_1\,\mathbf{v}_1\cdot\mathbf{t}_1 = \mathbf{t}_1\cdot(\delta\mathbf{u}_{N+1} - \delta\mathbf{x}_0 - \xi_1\delta\mathbf{v}_1 - \xi_2\delta\mathbf{v}_2)\,. \tag{5.1.1–9}$$

Similarly, taking the dot product of Equation 5.1.1–7 with $\mathbf{t}_2 = \mathbf{v}_2/\|\mathbf{v}_2\|$ and setting $h = 0$ results in the following expression for the variation of the second slip component:

$$\delta s_2 \stackrel{\text{def}}{=} \delta\xi_2\, \mathbf{v}_2 \cdot \mathbf{t}_2 = \mathbf{t}_2 \cdot (\delta\mathbf{u}_{N+1} - \delta\mathbf{x}_0 - \xi_1\delta\mathbf{v}_1 - \xi_2\delta\mathbf{v}_2)\,. \tag{5.1.1–10}$$

Formulation for two-dimensional small-sliding rigid contact

The formulation for two-dimensional small-sliding rigid contact follows from its deformable counterpart by exploiting the fact that the evolution of the contact plane is fully determined by the motion of the rigid body's reference node. Figure 5.1.1–2 shows how the undeformed coordinates $\mathbf{X}_0$ of the contact plane's anchor point are related vectorially to the undeformed coordinates of the rigid reference node, $\mathbf{X}_{rs}$, and the relative position vector $\mathbf{R}$.We can express this relationship as

$$\mathbf{X}_0 = \mathbf{X}_{rs} + \mathbf{R}\,.$$

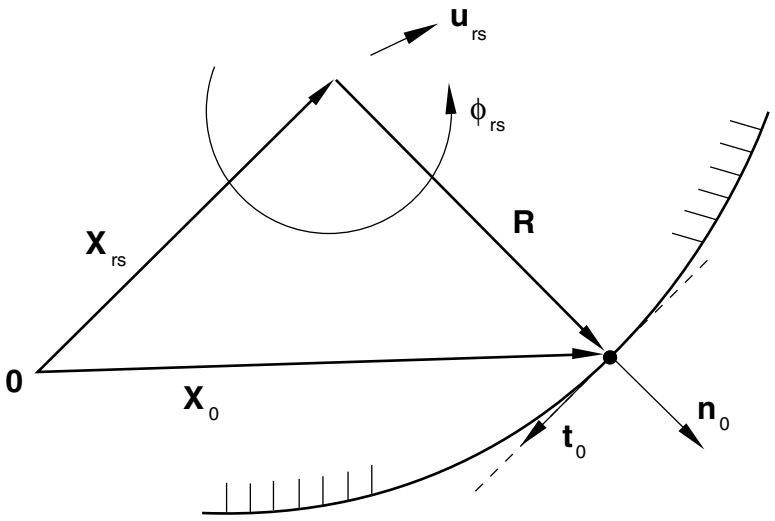


Figure 5.1.1–2 Rigid body reference geometry.

Suppose the rigid reference node undergoes a motion described by the displacement vector $\mathbf{u}_{rs}$ and the rotation vector $\phi_{rs}\mathbf{e}_z$, then the current coordinates of the contact plane's anchor point are given by

$$\begin{aligned}\mathbf{x}_0 &= \mathbf{X}_{rs} + \mathbf{u}_{rs} + \mathbf{C}(\phi_{rs}\mathbf{e}_z)\cdot\mathbf{R}\,,\\ &= \mathbf{x}_{rs} + \mathbf{r}\,,\end{aligned} \tag{5.1.1–11}$$

where $\mathbf{C}$ is the orthogonal matrix that produces the rotation $\phi_{rs}\mathbf{e}_z$ (see "Rotation variables," Section 1.3.1) and $\mathbf{r}$ is the current position vector from the rigid reference node to the anchor point. The rotation matrix is also used to obtain the contact plane's current tangent and current normal as follows:

$$\begin{aligned} \mathbf{t} &= \mathbf{C}(\phi_{rs}\mathbf{e}_z)\cdot\mathbf{t}_0\,, \\ \mathbf{n} &= \mathbf{C}(\phi_{rs}\mathbf{e}_z)\cdot\mathbf{n}_0\,, \end{aligned} \tag{5.1.1–12}$$

where $\mathbf{t}_0$ and $\mathbf{n}_0$ are the initial contact tangent and normal at $\mathbf{X}_0$, respectively, as shown in Figure 5.1.1–2. By definition, a rigid surface cannot stretch; therefore, a point on the rigid contact line associated with a slave node n_{N+1} is represented by the vector

$$\mathbf{p}(\xi) = \mathbf{x}_0 + \xi\mathbf{t}\,,$$

where—unlike the vector $\mathbf{v}$ in Equation 5.1.1–3—the tangent $\mathbf{t}$ is always a unit vector.

Linearization of Equation 5.1.1–11 and Equation 5.1.1–12 results in the following expressions for the first variations of the anchor coordinates and the contact plane's tangent:

$$\begin{aligned} \delta\mathbf{x}_0 &= \delta\mathbf{u}_{rs} + (-r_y\mathbf{e}_x + r_x\mathbf{e}_y)\delta\phi_{rs}\,, \\ \delta\mathbf{t} &= (-t_y\mathbf{e}_x + t_x\mathbf{e}_y)\delta\phi_{rs}\,. \end{aligned} \tag{5.1.1–13}$$

Replacing $\mathbf{v}$ by $\mathbf{t}$ in Equation 5.1.1–5, noting that $\mathbf{t}\cdot\delta\mathbf{t} = 0$, and substituting for $\delta\mathbf{x}_0$ and $\delta\mathbf{t}$ from Equation 5.1.1–13 results in the following expression for δh:

$$\delta h = -\mathbf{n}\cdot\left[\delta\mathbf{u}_{N+1} - \delta\mathbf{u}_{rs} + \big((r_y + \xi t_y)\mathbf{e}_x - (r_x + \xi t_x)\mathbf{e}_y\big)\delta\phi_{rs}\right]. \tag{5.1.1–14}$$

Similar manipulation of Equation 5.1.1–6 yields the following expression for δs:

$$\delta s \overset{\text{def}}{=} \delta\xi = \mathbf{t}\cdot\left[\delta\mathbf{u}_{N+1} - \delta\mathbf{u}_{rs} + (r_y\mathbf{e}_x - r_x\mathbf{e}_y)\delta\phi_{rs}\right]. \tag{5.1.1–15}$$

Formulation for three-dimensional small-sliding rigid contact

The three-dimensional small-sliding rigid contact formulation can also be derived from its deformable counterpart by generalizing some of the expressions that were introduced in the previous section. In particular, in this formulation the rigid reference node can undergo an arbitrary finite rotation described by the rotation vector $\boldsymbol{\phi}_{rs}$. Consequently, Equation 5.1.1–13 for the variations of the anchor point coordinates and the contact plane's tangent generalize to

$$\begin{aligned} \delta\mathbf{x}_0 &= \delta\mathbf{u}_{rs} + \widehat{\delta\boldsymbol{\theta}}_{rs}\cdot\mathbf{r}\,, \\ \delta\mathbf{t}_i &= \widehat{\delta\boldsymbol{\theta}}_{rs}\cdot\mathbf{t}_i\,, \qquad i = 1, 2, \end{aligned}$$

where $\widehat{\delta\boldsymbol{\theta}}_{rs}$ is the skew-symmetric matrix associated with the linearized rotation $\delta\boldsymbol{\theta}_{rs}$, as explained in "Rotation variables," Section 1.3.1.

Replacing $\mathbf{v}_i$ by $\mathbf{t}_i$ in Equation 5.1.1–8 through Equation 5.1.1–10 and substituting for $\delta\mathbf{x}_0$ and $\delta\mathbf{t}_i$ from above results in the following expressions for δh, δs_1, and δs_2:

$$\delta h = -\mathbf{n} \cdot \left[\delta\mathbf{u}_{N+1} - \delta\mathbf{u}_{rs} - \widehat{\delta\boldsymbol{\theta}}_{rs} \cdot (\mathbf{r} + \xi_1\mathbf{t}_1 + \xi_2\mathbf{t}_2)\right], \tag{5.1.1–16}$$

$$\delta s_1 \overset{\text{def}}{=} \delta\xi_1 = \mathbf{t}_1 \cdot \left[\delta\mathbf{u}_{N+1} - \delta\mathbf{u}_{rs} - \widehat{\delta\boldsymbol{\theta}}_{rs} \cdot (\mathbf{r} + \xi_2\mathbf{t}_2)\right], \tag{5.1.1–17}$$

$$\delta s_2 \overset{\text{def}}{=} \delta\xi_2 = \mathbf{t}_2 \cdot \left[\delta\mathbf{u}_{N+1} - \delta\mathbf{u}_{rs} - \widehat{\delta\boldsymbol{\theta}}_{rs} \cdot (\mathbf{r} + \xi_1\mathbf{t}_1)\right]. \tag{5.1.1–18}$$

Reference

- "Contact formulations in Abaqus/Standard," Section 38.1.1 of the Abaqus Analysis User's Guide

5.1.2 FINITE-SLIDING INTERACTION BETWEEN DEFORMABLE BODIES

Product: Abaqus/Standard

Abaqus/Standard provides two formulations for modeling the interaction between two deformable bodies. The first is a small-sliding formulation in which the contacting surfaces can undergo only relatively small sliding relative to each other but arbitrary rotation of the surfaces is permitted. This formulation is discussed in "Small-sliding interaction between bodies," Section 5.1.1. The second is a finite-sliding formulation where separation and sliding of finite amplitude and arbitrary rotation of the surfaces may arise. The formulation for two-dimensional and axisymmetric analysis, as well as for tube-in-tube analysis, is discussed in this section.

Depending on the type of contact problem, two approaches are available to the user for specifying the finite-sliding capability: (1) defining possible contact conditions by identifying and pairing potential contact surfaces and (2) using contact elements. With the first approach Abaqus automatically generates the appropriate contact elements.

In axisymmetric problems with asymmetric deformations, ISL21A and ISL22A elements can be used to model contact with CAXA or SAXA elements. Sliding of tubes inside each other can be modeled with ITT21 and ITT31 elements.

To define a sliding interface between two surfaces, one of the surfaces (the "slave" surface) is covered with ISL or ITT elements. The other surface (the "master" surface) is defined as a slide line surface composed of a series of nodes ordered in sequence. The slide line itself can consist of linear or quadratic segments. If smoothing is used, these segments are connected with quadratic or cubic segments such that full slope continuity is achieved. The smoothing procedure is described later in this section.

Two-dimensional and axisymmetric slide line elements

Consider contact of a node on the slave surface n_1 with a segment of the master surface described by nodes $n_2, n_3, \ldots$, where the number of nodes depends on the order of the segment. For a linear segment the number of nodes is 2, whereas for a quadratic segment the number of nodes is 3. For a smoothed section of a linear slide line, the number of nodes is also 3; and for a smoothed section of a quadratic slide line, the number of nodes is 5. If the contact occurs at the (convex) vertex of two segments, only a single node will enter the equations. A typical linear segment is shown in Figure 5.1.2–1, and a quadratic segment is displayed in Figure 5.1.2–2. Smoothed segments are shown later in this section.

To derive the equations governing these elements, we consider the coordinates in the plane of the slide line. For the axisymmetric elements, this plane coincides with the two-dimensional space. First, we determine the point $\mathbf{x}$ on the segment closest to the point $\mathbf{x}_1$ on the slave surface. We also determine the normal $\mathbf{n}$ and tangent $\mathbf{t}$ to the segment at that point. The point $\mathbf{x}$ and the normal $\mathbf{n}$ can be related to the overclosure h with the relation

$$\mathbf{n}h = \mathbf{x} - \mathbf{x}_1. \tag{5.1.2–1}$$

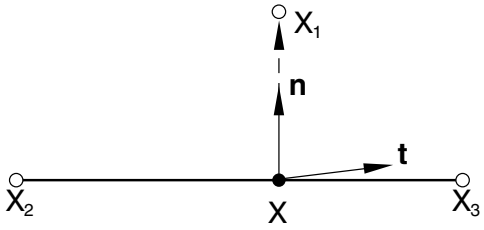


Figure 5.1.2–1 Linear slide line segment.

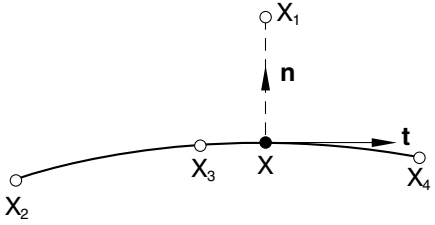


Figure 5.1.2–2 Quadratic slide line segment.

Since $\mathbf{x}$ is on the segment, its position is defined completely by the interpolation function N_i for the segment, the position g, and the position $\mathbf{x}_i$ of the nodes n_i that are part of the segment. That allows us to write for Equation 5.1.2–1

$$\mathbf{n}h = N_i(g)\mathbf{x}_i, \tag{5.1.2–2}$$

where $N_1 = -1$ and N_2, N_3, ... are functions of g. For instance, for a linear segment you obtain $N_2 = \frac{1}{2}(1-g)$, $N_3 = \frac{1}{2}(1+g)$. For a quadratic segment you use $N_2 = \frac{1}{2}g(g-1)$, $N_3 = 1-g^2$, $N_4 = \frac{1}{2}g(g+1)$. Similar expressions are obtained for smoothed segments of the slide line. The tangent $\mathbf{t}$ to the slide line at point $\mathbf{x}$ follows with

$$\mathbf{t} \overset{\text{def}}{=} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}s} = \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}g} \Big/ \left|\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}g}\right|, \tag{5.1.2–3}$$

where

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}g} = \frac{\mathrm{d}N_i}{\mathrm{d}g}\mathbf{x}_i.$$

The position of point $\mathbf{x}$ is determined from the condition that the normal and tangent must be orthogonal, which leads to the following equation for g:

$$\mathbf{n} \cdot \mathbf{t} = N_i(g) \frac{\mathrm{d}N_j(g)}{\mathrm{d}g} \, \mathbf{x}_i \cdot \mathbf{x}_j = 0.$$

For linear segments this yields a linear equation, which can be solved directly. For quadratic and cubic segments it leads to third- or fifth-order equations, which must be solved iteratively. The equation is solved using Newton's method preceded by a number of bisections to find the true minimum distance solution.

To obtain the contact/slip equation, the position equation (Equation 5.1.2–2) is linearized. This linearization yields

$$\delta\mathbf{n}h + \mathbf{n}\delta h = \frac{\mathrm{d}N_i}{\mathrm{d}g}\mathbf{x}_i\delta g + N_i\delta\mathbf{x}_i = \mathbf{t}\delta s + N_i\delta\mathbf{x}_i,$$

where δs is the slip. In the direction of contact, $\mathbf{n}$, this yields

$$\delta h = N_i \, \mathbf{n} \cdot \delta\mathbf{x}_i; \tag{5.1.2–4}$$

and in the direction of slip, $\mathbf{t}$, one finds

$$\delta s = -N_i \, \mathbf{t} \cdot \delta\mathbf{x}_i - h \, \mathbf{t} \cdot \delta\mathbf{n}.$$

It is assumed that slip is relevant only if the node $\mathbf{x}_1$ is on the slide line; hence, it will be assumed that $h = 0$. From this follows

$$\delta s = -N_i \, \mathbf{t} \cdot \delta\mathbf{x}_i. \tag{5.1.2–5}$$

To obtain the initial stress stiffness terms, the second variations of h and s must be calculated. From Equation 5.1.2–4 follows

$$\mathrm{d}\delta h = \delta\mathbf{x}_i \cdot \mathbf{n} \, \frac{\mathrm{d}N_i}{\mathrm{d}g}\mathrm{d}g + N_i \, \delta\mathbf{x}_i \cdot \mathrm{d}\mathbf{n}. \tag{5.1.2–6}$$

The first term is expressed readily in terms of $\mathrm{d}\mathbf{x}_i$ with the help of Equation 5.1.2–5:

$$\frac{\mathrm{d}N_i}{\mathrm{d}g}\mathrm{d}g = \frac{\mathrm{d}N_i}{\mathrm{d}g}\frac{\mathrm{d}g}{\mathrm{d}s}\frac{\mathrm{d}s}{\mathrm{d}g}\mathrm{d}g = \frac{\mathrm{d}N_i}{\mathrm{d}s}\mathrm{d}s = -\frac{\mathrm{d}N_i}{\mathrm{d}s}N_j \, \mathbf{t} \cdot \mathrm{d}\mathbf{x}_j. \tag{5.1.2–7}$$

The rate of change of the normal can be re-expressed as

$$\mathrm{d}\mathbf{n} = (\mathbf{n}\,\mathbf{n} + \mathbf{t}\,\mathbf{t}) \cdot \mathrm{d}\mathbf{n} = \mathbf{t}\,\mathbf{t} \cdot \mathrm{d}\mathbf{n} = -\mathbf{t}\,\mathbf{n} \cdot \mathrm{d}\mathbf{t}. \tag{5.1.2–8}$$

In this equation $\mathbf{n} \cdot \mathrm{d}\mathbf{t}$ can be obtained from Equation 5.1.2–3:

$$\begin{aligned}\mathbf{n}\cdot\mathrm{d}\mathbf{t} =&\mathbf{n}\cdot\mathrm{d}\left(\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}g}\Big/\left|\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}g}\right|\right)\Big| = \mathbf{n}\cdot\left(\mathbf{x}_i\frac{\mathrm{d}^2N_i}{\mathrm{d}g^2}\mathrm{d}g+\frac{\mathrm{d}N_i}{\mathrm{d}g}\mathrm{d}\mathbf{x}_i\right)\Big/\left|\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}g}\right| \\ =&-\rho_n\mathrm{d}s+\frac{\mathrm{d}N_i}{\mathrm{d}s}\,\mathbf{n}\cdot\mathrm{d}\mathbf{x}_i=\rho_nN_i\,\mathbf{t}\cdot\mathrm{d}\mathbf{x}_i+\frac{\mathrm{d}N_i}{\mathrm{d}s}\mathbf{n}\cdot\mathrm{d}\mathbf{x}_i,\end{aligned} \tag{5.1.2–9}$$

where use was made of Equation 5.1.2–5 and ρ_n (the segment curvature) is defined as

$$\rho_n \overset{\text{def}}{=} -\mathbf{n}\cdot\frac{\mathrm{d}^2\mathbf{x}}{\mathrm{d}g^2}\Big/\left|\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}g}\right|^2. \tag{5.1.2–10}$$

For straight segments ρ_n obviously vanishes. Substitution of Equation 5.1.2–7 to Equation 5.1.2–9 in Equation 5.1.2–6 yields the final result:

$$d\delta h=-\delta\mathbf{x}_i\cdot\left(\mathbf{n}\frac{\mathrm{d}N_i}{\mathrm{d}s}N_j\mathbf{t}+\mathbf{t}N_i\frac{\mathrm{d}N_j}{\mathrm{d}s}\mathbf{n}+\mathbf{t}N_i\rho_nN_j\mathbf{t}\right)\cdot\mathrm{d}\mathbf{x}_j. \tag{5.1.2–11}$$

This expression is symmetric, as should be expected. The second variation in s can be derived along similar lines:

$$\mathrm{d}\delta s=-\delta\mathbf{x}_i\cdot\mathbf{t}\frac{\mathrm{d}N_i}{\mathrm{d}g}\mathrm{d}g-N_i\,\delta\mathbf{x}_i\cdot\mathrm{d}\mathbf{t}. \tag{5.1.2–12}$$

$$d\mathbf{t}=(\mathbf{n}\,\mathbf{n}+\mathbf{t}\,\mathbf{t})\cdot\mathrm{d}\mathbf{t}=\mathbf{n}\,\mathbf{n}\cdot\mathrm{d}\mathbf{t}. \tag{5.1.2–13}$$

Substitution of Equation 5.1.2–7, Equation 5.1.2–9, and Equation 5.1.2–13 in Equation 5.1.2–12 yields

$$\mathrm{d}\delta s=\delta\mathbf{x}_i\cdot\left(\mathbf{t}\frac{\mathrm{d}N_i}{\mathrm{d}s}N_j\mathbf{t}-\mathbf{n}N_i\frac{\mathrm{d}N_j}{\mathrm{d}s}\mathbf{n}-\mathbf{n}N_i\rho_nN_j\mathbf{t}\right)\cdot\mathrm{d}\mathbf{x}_j. \tag{5.1.2–14}$$

This expression is nonsymmetric. The second variations of h and s vanish if no slip in the slide plane occurs ($\mathrm{d}s=\delta s=0$).

Tube-tube interface elements

In the case of tube-tube interface elements it is assumed that the inner tube can be considered the slave surface and the outer tube the master surface. The tube-tube interface elements differ from the axisymmetric slide line elements in two ways. In the first place there is assumed to be a finite clearance between the tubes, which has the effect that the separation distance h is finite even when contact occurs. In the second place for the three-dimensional element ITT31 there is a second possible local tangent direction in the plane of the cross-section of the tubes. The derivations for the ITT elements follow much along the same line as the derivations for the ISL elements. The contact equation can be written in the form

$$-\mathbf{n}(h + h_o) = \mathbf{x}_1 - \mathbf{x}. \tag{5.1.2–15}$$

Here $\mathbf{x}$ is a point on the outer tube that potentially contacts point $\mathbf{x}_1$ on the inner tube, as shown in Figure 5.1.2–3.

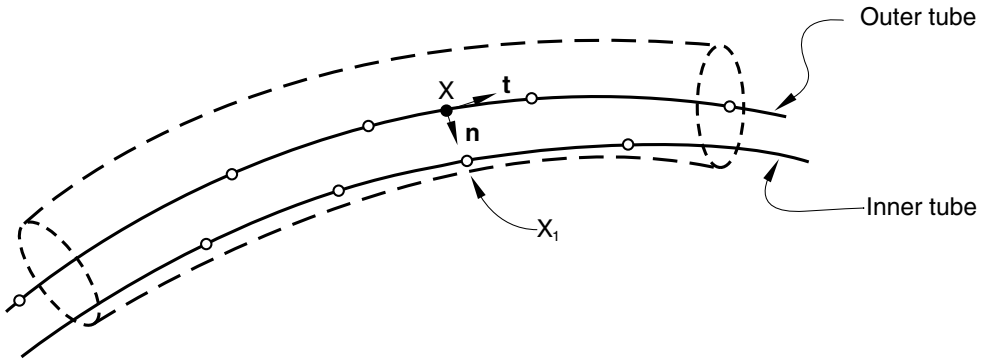


Figure 5.1.2–3 Tube-tube contact.

In this equation h_o is the (positive) radial clearance between the tubes. In a similar way as was done for the ISL elements, the contact equation can be written in the form

$$\mathbf{n}(h + h_o) = -N_i(g)\, \mathbf{x}_i. \tag{5.1.2–16}$$

The sign reversal as compared to Equation 5.1.2–2 is related to the internal nature of the contact as compared to the external nature for the regular slide line elements. The linearized form of Equation 5.1.2–16 is

$$\delta\mathbf{n}(h + h_o) + \mathbf{n}\delta h = -\mathbf{t}\delta s - N_i\delta\mathbf{x}_i.$$

As before, we assume that during contact $h = 0$. In the contact direction this yields

$$\delta h = -N_i\, \mathbf{n} \cdot \delta\mathbf{x}_i, \tag{5.1.2–17}$$

and in the direction along the pipe,

$$\delta s = -h_o\, \mathbf{t} \cdot \delta\mathbf{n} - N_i\, \mathbf{t} \cdot \delta\mathbf{x}_i. \tag{5.1.2–18}$$

With use of $\mathbf{n} \cdot \mathbf{t} = 0$, it readily follows

$$\mathbf{t}\cdot\delta\mathbf{n} = -\mathbf{n}\cdot\delta\mathbf{t} = \rho_n\delta s - \frac{\mathrm{d}N_i}{\mathrm{d}s}\mathbf{n}\cdot\delta\mathbf{x}_i, \tag{5.1.2–19}$$

which transforms Equation 5.1.2–18 into

$$\delta s = \frac{1}{1+\rho_n h_o}(-N_i\,\mathbf{t}\cdot\delta\mathbf{x}_i + h_o\frac{\mathrm{d}N_i}{\mathrm{d}s}\mathbf{n}\cdot\delta\mathbf{x}_i), \tag{5.1.2–20}$$

where ρ_n is the segment curvature as defined by Equation 5.1.2–10. For the three-dimensional tube-tube interface element, one can define the transverse local tangent direction $\mathbf{s} \stackrel{\mathrm{def}}{=} \mathbf{t}\times\mathbf{n}$, which yields

$$\delta s_2 = -N_i\,\mathbf{s}\cdot\delta\mathbf{x}_i \;\; (= h_o\,\mathbf{s}\cdot\delta\mathbf{n}). \tag{5.1.2–21}$$

The initial stress stiffness terms are again obtained by taking the second variations in h_o, s, and s_2. From Equation 5.1.2–17 follows

$$\mathrm{d}\delta h = \rho_n\delta s\mathrm{d}s - \delta s\frac{\mathrm{d}N_i}{\mathrm{d}s}\mathbf{n}\cdot\mathrm{d}\mathbf{x}_i - \delta\mathbf{x}_i\cdot\mathbf{n}\frac{\mathrm{d}N_i}{\mathrm{d}s}\mathrm{d}s + h_o\delta\mathbf{n}\cdot\mathrm{d}\mathbf{n}, \tag{5.1.2–22}$$

and with Equation 5.1.2–19 and Equation 5.1.2–21 follows

$$\mathrm{d}\mathbf{n} = \mathbf{t}\mathbf{t}\cdot\mathrm{d}\mathbf{n} + \mathbf{s}\,\mathbf{s}\cdot\mathrm{d}\mathbf{n} = \mathbf{t}\rho_n\mathrm{d}s - \frac{\mathrm{d}N_i}{\mathrm{d}s}\mathbf{n}\cdot\mathrm{d}\mathbf{x}_i. \tag{5.1.2–23}$$

Substitution of Equation 5.1.2–20 and Equation 5.1.2–23 in Equation 5.1.2–22 yields

$$\begin{aligned}\mathrm{d}\delta h = \rho_n(1+h_o\rho_n)\delta s\mathrm{d}s & \\ -\delta s\,(1+h_o\rho_n)\frac{\mathrm{d}N_i}{\mathrm{d}s}\mathbf{n}\cdot\mathrm{d}\mathbf{x}_i & \\ -\delta\mathbf{x}_i\cdot\mathbf{n}\frac{\mathrm{d}N_i}{\mathrm{d}s}(1+h_o\rho_n)\mathrm{d}s & \\ +h_o\delta\mathbf{x}_i\cdot\mathbf{n}\frac{\mathrm{d}N_i}{\mathrm{d}s}\frac{\mathrm{d}N_j}{\mathrm{d}s}\mathbf{n}\cdot\mathrm{d}\mathbf{x}_j & \\ +\frac{\delta s_2\mathrm{d}s_2}{h_o} &\end{aligned} \tag{5.1.2–24}$$

This expression is symmetric. In the $\mathbf{t}$-direction the virtual work term has the form

$$F_t\,(\delta s + h_o\,\mathbf{t}\cdot\delta\mathbf{n}) \stackrel{\mathrm{def}}{=} F_t\,\delta\overline{s},$$

which yields with Equation 5.1.2–18

$$\delta\overline{s} = -N_i\,\mathbf{t}\cdot\delta\mathbf{x}_i. \tag{5.1.2–25}$$

For the second variation follows

$$\mathrm{d}\delta\overline{s} = h_o\delta\mathbf{n}\cdot\mathrm{d}\mathbf{t} - \mathbf{t}\cdot\delta\mathbf{x}_i\frac{\mathrm{d}N_i}{\mathrm{d}s}\mathrm{d}s - \mathbf{t}\cdot\delta\mathbf{n}\mathrm{d}h - \delta h\mathbf{t}\cdot\mathrm{d}\mathbf{n}. \tag{5.1.2–26}$$

$$\begin{aligned}\mathrm{d}\mathbf{t} = \mathrm{d}\left(\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}s}\right) &= \frac{\mathrm{d}^2\mathbf{x}}{\mathrm{d}g^2}\mathrm{d}g\Big/\left|\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}g}\right| - \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}g}\,\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}g}\cdot\frac{\mathrm{d}^2\mathbf{x}}{\mathrm{d}g^2}\mathrm{d}g\Big/\left|\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}g}\right|^3 \\ &= \frac{\mathrm{d}N_i}{\mathrm{d}s}(\mathbf{I}-\mathbf{t}\,\mathbf{t})\cdot\mathrm{d}\mathbf{x}_i + \frac{\mathrm{d}^2\mathbf{x}}{\mathrm{d}g^2}\Big/\left|\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}g}\right|^2\cdot(\mathbf{I}-\mathbf{t}\,\mathbf{t})\mathrm{d}s \\ &= \frac{\mathrm{d}N_i}{\mathrm{d}s}(\mathbf{n}\,\mathbf{n}+\mathbf{s}\,\mathbf{s})\cdot\mathrm{d}\mathbf{x}_i - (\rho_n\mathbf{n}+\rho_s\mathbf{s})\mathrm{d}s,\end{aligned} \tag{5.1.2–27}$$

where the transverse segment curvature ρ_s is defined by

$$\rho_s \overset{\text{def}}{=} -\mathbf{s}\cdot\frac{\mathrm{d}^2\mathbf{x}}{\mathrm{d}g^2}\Big/\left|\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}g}\right|^2. \tag{5.1.2–28}$$

In this expression the terms involving $\mathbf{s}$ vanish for element type ITT21. Substitution of Equation 5.1.2–20 and Equation 5.1.2–27 in Equation 5.1.2–26 yields

$$\begin{aligned}\mathrm{d}\delta\overline{s} = \delta s\cdot\Big(&-\rho_s\mathrm{d}s + \frac{\mathrm{d}N_i}{\mathrm{d}s}\mathbf{s}\cdot\mathrm{d}\mathbf{x}_i\Big) \\ &- \delta\mathbf{x}_i\cdot\mathbf{t}\frac{\mathrm{d}N_i}{\mathrm{d}s}\mathrm{d}s \\ &- \rho_n\delta s\mathrm{d}h + \delta\mathbf{x}_i\cdot\mathbf{n}\frac{\mathrm{d}N_i}{\mathrm{d}s}\mathrm{d}h \\ &- \rho_n\delta h\mathrm{d}s + \delta h\frac{\mathrm{d}N_i}{\mathrm{d}s}\mathbf{n}\cdot\mathrm{d}\mathbf{x}_i\end{aligned} \tag{5.1.2–29}$$

In the $\mathbf{s}$-direction one simply looks at the first variation in δs_2:

$$\mathrm{d}\delta s_2 = \delta\mathbf{x}_i\cdot\left(-\mathbf{s}\frac{\mathrm{d}N_i}{\mathrm{d}s}\mathrm{d}s - N_i\mathrm{d}\mathbf{s}\right). \tag{5.1.2–30}$$

$$\mathrm{d}\mathbf{s} = \mathbf{n}\,\mathbf{n}\cdot\mathrm{d}\mathbf{s} + \mathbf{t}\,\mathbf{t}\cdot\mathrm{d}\mathbf{s} = -\mathbf{n}\,\mathbf{s}\cdot\mathrm{d}\mathbf{n} - \mathbf{t}\,\mathbf{s}\cdot\mathrm{d}\mathbf{t}.$$

With Equation 5.1.2–20, Equation 5.1.2–23, and Equation 5.1.2–27 follows

$$\mathrm{d}\mathbf{s} = \frac{-\mathbf{n}\mathrm{d}s_2}{h_o} + \mathbf{t}\left(\rho_s\mathrm{d}s - \frac{\mathrm{d}N_i}{\mathrm{d}s}\mathbf{s}\cdot\mathrm{d}\mathbf{x}_i\right).$$

Substituted in Equation 5.1.2–30 this yields

$$\begin{aligned} \mathrm{d}\delta s_2 = &- \delta\mathbf{x}_i \cdot \mathbf{s}\frac{\mathrm{d}N_i}{\mathrm{d}s}\mathrm{d}s + N_i\delta\mathbf{x}_i \cdot \mathbf{n}\frac{\mathrm{d}s_2}{h_0} \\ &+ N_i\delta\mathbf{x}_i \cdot \mathbf{t}\left(-\rho_s\mathrm{d}s + \frac{\mathrm{d}N_j}{\mathrm{d}s}\mathbf{s}\cdot\mathrm{d}\mathbf{x}_j\right). \end{aligned} \tag{5.1.2–31}$$

Slide line smoothing

At the junction of two segments along a slide line, a discontinuity in slope may occur. This discontinuity can cause convergence problems because during iteration a contact point might move back and forth between two segments. Hence, it is useful to smooth the transition between segments. Consider first the transition between two linear segments (Figure 5.1.2–4).

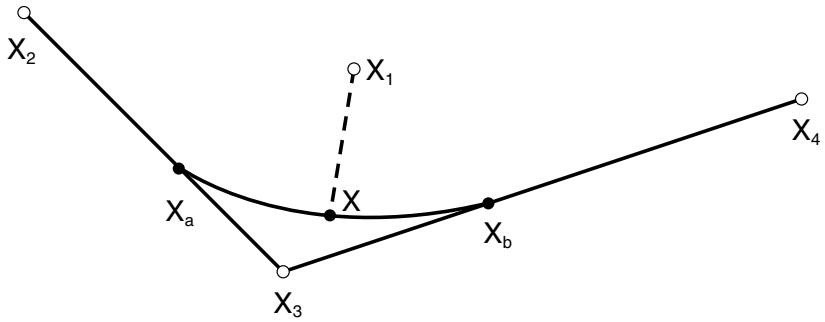


Figure 5.1.2–4 Transition between linear segments.

The junction of the two segments is connected by a Hermitian polynomial between the points $\mathbf{x}_a$ and $\mathbf{x}_b$, which are located on the segments:

$$\mathbf{x}_a = \alpha\mathbf{x}_2 + (1-\alpha)\mathbf{x}_3, \quad \mathbf{x}_b = \alpha\mathbf{x}_4 + (1-\alpha)\mathbf{x}_3.$$

The positions of $\mathbf{x}_a$ and $\mathbf{x}_b$ are, hence, determined by the smoothing factor α, which is specified directly by the user in the range $0 \le \alpha \le 0.5$.

We choose g as the parameter coordinate on the smoothed segment, with $-1 \le g \le 1$. At the extreme values for g the coordinates are $\mathbf{x}_a$ and $\mathbf{x}_b$, and for the coordinate derivatives we choose

$$\frac{\mathrm{d}\mathbf{x}_a}{\mathrm{d}g} = \mathbf{x}_3 - \mathbf{x}_a = \alpha(\mathbf{x}_3 - \mathbf{x}_2), \quad \frac{\mathrm{d}\mathbf{x}_b}{\mathrm{d}g} = \mathbf{x}_b - \mathbf{x}_3 = \alpha(\mathbf{x}_4 - \mathbf{x}_3).$$

With the Hermitian interpolation functions we can calculate the position $\mathbf{x}$ of a point on the segment as a function of g:

$$\mathbf{x} = \frac{1}{4}(g^3 - 3g + 2)\mathbf{x}_a + \frac{1}{4}(-g^3 + 3g + 2)\mathbf{x}_b + \frac{1}{4}(g^3 - g^2 - g + 1)\frac{\mathrm{d}\mathbf{x}_a}{\mathrm{d}g} + \frac{1}{4}(g^3 + g^2 - g - 1)\frac{\mathrm{d}\mathbf{x}_b}{\mathrm{d}g}.$$

Combining the above equations yields

$$\mathbf{x} = \frac{\alpha}{4}(g^2 - 2g + 1)\mathbf{x}_2 + \frac{1}{4}(4 - 2\alpha(g^2 + 1))\mathbf{x}_3 + \frac{\alpha}{4}(g^2 + 2g + 1)\mathbf{x}_4. \tag{5.1.2–32}$$

The smoothing has in fact been done with a second-order polynomial. In the formulation of the interface contact and friction equations, the treatment of a smoothing segment is identical to the treatment of a regular segment. A transition between quadratic segments is shown in Figure 5.1.2–5.

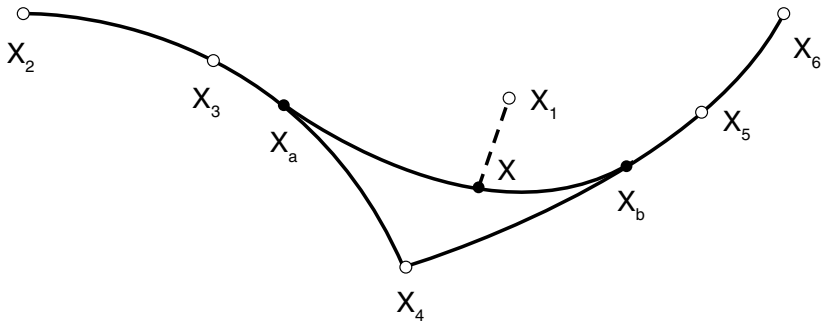


Figure 5.1.2–5 Transition between quadratic segments.

It is readily established that in this case

$$\mathbf{x}_a = (2\alpha^2 - \alpha)\mathbf{x}_2 + (-4\alpha^2 + 4\alpha)\mathbf{x}_3 + (1 - 3\alpha + 2\alpha^2)\mathbf{x}_4,$$
$$\mathbf{x}_b = (2\alpha^2 - \alpha)\mathbf{x}_6 + (-4\alpha^2 + 4\alpha)\mathbf{x}_5 + (1 - 3\alpha + 2\alpha^2)\mathbf{x}_4.$$

For the coordinate derivatives we choose

$$\frac{\mathrm{d}\mathbf{x}_a}{\mathrm{d}g} = -\alpha\frac{\mathrm{d}\mathbf{x}_a}{\mathrm{d}\alpha} = (\,\alpha - 4\alpha^2)\mathbf{x}_2 + (-4\alpha + 8\alpha^2)\mathbf{x}_3 + (\,3\alpha - 4\alpha^2)\mathbf{x}_4,$$
$$\frac{\mathrm{d}\mathbf{x}_b}{\mathrm{d}g} = \alpha\frac{\mathrm{d}\mathbf{x}_b}{\mathrm{d}\alpha} = (-\alpha + 4\alpha^2)\mathbf{x}_6 + (\,4\alpha - 8\alpha^2)\mathbf{x}_5 + (-3\alpha + 4\alpha^2)\mathbf{x}_4.$$

As before, α is specified directly by the user. Substitution of these equations in the Hermite shape functions yields

$$\begin{aligned}\mathbf{x} = \frac{1}{4}\big(&\alpha[-2\alpha g^3 + (\,4\alpha - 1)g^2 + (-2\alpha + 2)g - 1]\mathbf{x}_2 \\ &+ \alpha[\,4\alpha g^3 + (-8\alpha + 4)g^2 + (\,4\alpha - 8)g + 4]\mathbf{x}_3 \\ &+ [(8\alpha^2 - 6\alpha)g^2 + (-6\alpha + 4)]\mathbf{x}_4 \\ &+ \alpha[-4\alpha g^3 + (-8\alpha + 4)g^2 + (-4\alpha + 8)g + 4]\mathbf{x}_5 \\ &+ \alpha[\,2\alpha g^3 + (4\alpha - 1)g^2 + (\,2\alpha - 2)g - 1]\mathbf{x}_6\big).\end{aligned} \tag{5.1.2–33}$$

If $\mathbf{x}_3 = \frac{1}{2}(\mathbf{x}_2 + \mathbf{x}_4)$ and $\mathbf{x}_5 = \frac{1}{2}(\mathbf{x}_4 + \mathbf{x}_6)$, the equations reduce to the same equations as were used for smoothing between linear segments.

For the transition between a linear segment with nodes $\mathbf{x}_2$ and $\mathbf{x}_4$ and a quadratic segment with nodes $\mathbf{x}_4$, $\mathbf{x}_5$, and $\mathbf{x}_6$ the formulas become

$$\begin{aligned}\mathbf{x}_a &= \alpha\mathbf{x}_2 + (1 - \alpha)\mathbf{x}_4, \\ \mathbf{x}_b &= (2\alpha^2 - \alpha)\mathbf{x}_6 + (-4\alpha^2 + 4\alpha)\mathbf{x}_5 + (1 - 3\alpha + 2\alpha^2)\mathbf{x}_4.\end{aligned}$$

The coordinate derivatives are

$$\begin{aligned}\frac{\mathrm{d}\mathbf{x}_a}{\mathrm{d}g} &= \mathbf{x}_4 - \mathbf{x}_a = \alpha(\mathbf{x}_4 - \mathbf{x}_2), \\ \frac{\mathrm{d}\mathbf{x}_b}{\mathrm{d}g} &= \alpha\frac{\mathrm{d}\mathbf{x}_b}{\mathrm{d}\alpha} = (-\alpha + 4\alpha^2)\mathbf{x}_6 + (\,4\alpha - 8\alpha^2)\mathbf{x}_5 + (-3\alpha + 4\alpha^2)\mathbf{x}_4.\end{aligned}$$

As a result,

$$\begin{aligned}\mathbf{x} = \frac{1}{4}\big(&\alpha[g^2 - 2g + 1]\mathbf{x}_2 \\ &+ [2\alpha^2 g^3 + (4\alpha^2 - 4\alpha)g^2 + (2\alpha^2 - 4\alpha)g + (4 - 4\alpha)]\mathbf{x}_4 \\ &+ \alpha[-4\alpha g^3 + (-8\alpha + 4)g^2 + (-4\alpha + 8)g + 4]\mathbf{x}_5 \\ &+ \alpha[\,2\alpha g^3 + (4\alpha - 1)g^2 + (\,2\alpha - 2)g - 1]\mathbf{x}_6\big).\end{aligned} \tag{5.1.2–34}$$

For the transition between a quadratic segment with nodes $\mathbf{x}_2$, $\mathbf{x}_3$, and $\mathbf{x}_4$ and a linear segment with nodes $\mathbf{x}_4$ and $\mathbf{x}_6$, the formulas become

$$\begin{aligned}\mathbf{x}_a &= (2\alpha^2 - \alpha)\mathbf{x}_2 + (-4\alpha^2 + 4\alpha)\mathbf{x}_3 + (1 - 3\alpha + 2\alpha^2)\mathbf{x}_4, \\ \mathbf{x}_b &= \alpha\mathbf{x}_6 + (1 - \alpha)\mathbf{x}_4.\end{aligned}$$

The coordinate derivatives are

$$\begin{aligned}\frac{\mathrm{d}\mathbf{x}_a}{\mathrm{d}g} &= -\alpha\frac{\mathrm{d}\mathbf{x}_a}{\mathrm{d}\alpha} = (\,\alpha - 4\alpha^2)\mathbf{x}_2 + (-4\alpha + 8\alpha^2)\mathbf{x}_3 + (\,3\alpha - 4\alpha^2)\mathbf{x}_4, \\ \frac{\mathrm{d}\mathbf{x}_b}{\mathrm{d}g} &= \mathbf{x}_b - \mathbf{x}_4 = \alpha(\mathbf{x}_6 - \mathbf{x}_4).\end{aligned}$$

As a result,

$$\begin{aligned}\mathbf{x} = \frac{1}{4}\big(&\alpha[-2\alpha g^3 + (\,4\alpha - 1)g^2 + (-2\alpha + 2)g - 1]\mathbf{x}_2 \\ &+ \alpha[\,4\alpha g^3 + (-8\alpha + 4)g^2 + (\,4\alpha - 8)g + 4]\mathbf{x}_3 \\ &+ [-2\alpha^2 g^3 + (4\alpha^2 - 4\alpha)g^2 - (2\alpha^2 - 4\alpha)g + (4 - 4\alpha)]\mathbf{x}_4 \\ &+ \alpha[\,g^2 + 2g + 1]\mathbf{x}_6\big).\end{aligned} \tag{5.1.2–35}$$

Compared to "pure" linear-to-linear and quadratic-to-quadratic segment transitions, in both "mixed" linear-to-quadratic and quadratic-to-linear cases, only the junction node $\mathbf{x}_4$ terms change.

Self contact

Self contact is available in the context of a surface folding and touching itself. Appropriate contact elements are generated internally for each node on the surface. A node is allowed to contact all of the surface segments, with the exception of the segments that are adjacent to the node. Since a node is simultaneously a master and a slave (producing symmetric master-slave relationships), overconstraints would occur, for example, if the meshes on both sides of the interface matched exactly. If only a pair of nodes matches in two dimensions, no problem occurs due to the smoothing carried out on the master side of the interface. To avoid solver problems due to overconstraining, three-dimensional self-contact is activated only if penalty-type contact is used.

The mathematical treatment of contact elements is the same as described above, with a few modifications in two dimensions. When two adjacent segments of the surface fold forming a sharp crack, the contact algorithm becomes pure master-slave instead of symmetric to prevent redundant contact constraints. It has been arbitrarily chosen that of these two segments the shortest is the slave and the longest is the master.

References

- "Contact formulations in Abaqus/Standard," Section 38.1.1 of the Abaqus Analysis User's Guide
- "Tube-to-tube contact elements," Section 40.3.1 of the Abaqus Analysis User's Guide
- "Slide line contact elements," Section 40.4.1 of the Abaqus Analysis User's Guide

5.1.3 FINITE-SLIDING INTERACTION BETWEEN A DEFORMABLE AND A RIGID BODY

Product: Abaqus/Standard

Abaqus/Standard provides two formulations for modeling the interaction between a deformable body and an arbitrarily shaped rigid body that may move during the history being modeled. The first is a small-sliding formulation in which the contacting surfaces can only undergo relatively small sliding relative to each other, but arbitrary rotation of the surfaces is permitted. This formulation is discussed in "Small-sliding interaction between bodies," Section 5.1.1. The second is a finite-sliding formulation where separation and sliding of finite amplitude and arbitrary rotation of the surfaces may arise. This formulation is discussed in this section.

The finite-sliding rigid contact capability is implemented by means of a family of contact elements that Abaqus automatically generates based on the data associated with the user-specified contact pairs. At each integration point these elements construct a measure of overclosure (penetration of the point on the surface of the deforming body into the rigid surface) and measures of relative shear sliding. These kinematic measures are then used, together with appropriate Lagrange multiplier techniques, to introduce surface interaction theories (contact and friction). A library of interaction theories is provided in Abaqus—these may be thought of as a library of "surface constitutive models." In this section we discuss only the kinematics of the interacting surfaces. The surface constitutive models are described in Chapter 4, "Mechanical Constitutive Theories."

Let A be a point on the deforming mesh, with current coordinates $\mathbf{x}_A$. Let C be the "rigid body reference node"—the node that defines the position of the rigid body—with current coordinates $\mathbf{x}_C$. Let A' be the closest point on the surface of the rigid body to A at which the normal to the surface of the rigid body, $\mathbf{n}$, passes through A. Define $\mathbf{r}$ as the vector from C to A'. The geometry described by these quantities is shown in Figure 5.1.3–1.

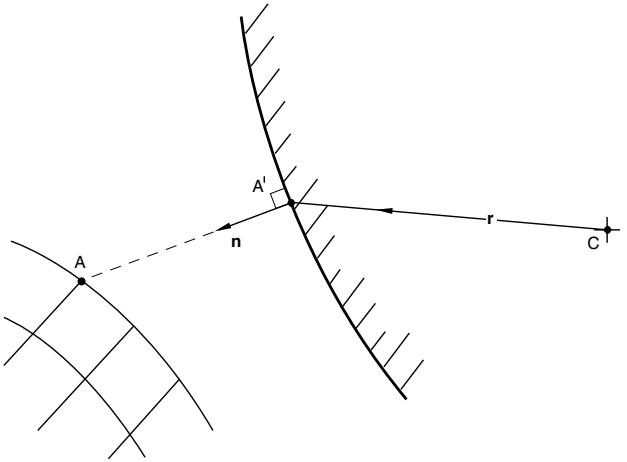


Figure 5.1.3–1 Rigid surface interface geometry.

Let h be the distance from A' to A along $-\mathbf{n}$: the "overclosure" of the surfaces. From the definitions introduced above,

$$\mathbf{n}h = -\mathbf{x}_A + \mathbf{x}_C + \mathbf{r}.$$

Then if $h < -c$ there is no contact between the surfaces at A, and no further surface interaction calculations need be done at this point. Here c is the clearance below which contact occurs. For a "hard" surface $c = 0$, but Abaqus/Standard also allows a "softened" surface to be introduced in which c may be nonzero (although c is usually very small compared to other dimensions). If $h \geq -c$ the surfaces are in contact. To enforce the contact constraint we will need the first variation of h, δh, and its second variation, $d\delta h$. These quantities are now derived.

Let S^α, $\alpha = 1, 2$ be locally orthogonal, distance measuring surface coordinates on the surface at A'. The S^α measure distance along the tangents $\mathbf{t}_\alpha$ to the surface at A': these tangents are constructed according to the standard Abaqus convention for such tangents to a surface in space. As the point A and the rigid body move, the projected point A' will move along. The movement consists of two parts: movement due to motion of the rigid body and motion relative to the body

$$\delta\mathbf{x}_{A'} = \delta\mathbf{x}_C + \delta\mathbf{r}|_{\gamma_\alpha} + \delta\mathbf{r}|_{\phi_C} = \delta\mathbf{x}_C + \delta\boldsymbol{\phi}_C \times \mathbf{r} + \mathbf{t}_\alpha \delta\gamma_\alpha,$$

where $\delta\gamma_\alpha$ is the "slip" of point A'. The normal $\mathbf{n}$ will also change due to rotation of the rigid surface and due to slip along the surface

$$\delta\mathbf{n} = \delta\mathbf{n}|_{\gamma_\alpha} + \delta\mathbf{n}|_{\phi_C} = \delta\boldsymbol{\phi}_C \times \mathbf{n} + \frac{\partial \mathbf{n}}{\partial S_\alpha}\delta\gamma_\alpha.$$

The linearized form of the contact equation, thus, becomes

$$\mathbf{n}\delta h + h(\delta\boldsymbol{\phi}_C \times \mathbf{n} + \frac{\partial \mathbf{n}}{\partial S_\alpha}\delta\gamma_\alpha) = -\delta\mathbf{x}_A + \delta\mathbf{x}_C + \delta\boldsymbol{\phi}_C \times \mathbf{r} + \mathbf{t}_\alpha\delta\gamma_\alpha.$$

For "hard" contact $h = 0$ exactly, and for soft contact we will assume $h = 0$ as well. The linearized kinematic equation, thus, becomes

$$\mathbf{n}\delta h = -\delta\mathbf{x}_A + \delta\mathbf{x}_C + \delta\boldsymbol{\phi}_C \times \mathbf{r} + \mathbf{t}_\alpha\delta\gamma_\alpha.$$

This equation can be split into normal and tangential components, which yields the contact equation,

$$\delta h = -\mathbf{n} \cdot (\delta\mathbf{x}_A - \delta\mathbf{x}_C) + (\mathbf{r} \times \mathbf{n}) \cdot \delta\boldsymbol{\phi}_C,$$

and the slip equations,

$$\delta\gamma_\alpha = \mathbf{t}_\alpha \cdot (\delta\mathbf{x}_A - \delta\mathbf{x}_C) - (\mathbf{r} \times \mathbf{t}_\alpha) \cdot \delta\boldsymbol{\phi}_C.$$

To obtain the second variation of h, it will again be assumed that $h = 0$. In addition, it will be assumed that $dh = \delta h = 0$, which is accurate for relatively "hard" contact. It then directly follows that

$$\mathbf{n}d\delta h = d\delta\mathbf{r},$$

and from the linearized kinematic equation follows

$$\mathbf{n}d\delta h = d(\delta\boldsymbol{\phi}_C \times \mathbf{r})|_{\gamma_\beta} + \delta\boldsymbol{\phi}_C \times d\mathbf{r}|_{\phi_C} + d\mathbf{t}|_{\gamma_\beta}\delta\gamma_\alpha + d\mathbf{t}|_{\phi_C}\delta\gamma_\alpha + \mathbf{t}_\alpha d\delta\gamma_\alpha,$$

where we have used $d\delta\mathbf{x}_A = d\delta\mathbf{x}_C = d\delta\boldsymbol{\phi}_C = 0$. The first term corresponds to a second-order variation on the vector $\mathbf{r}$ for rigid body rotations around point C and is given by (see "Rotation variables," Section 1.3.1):

$$d(\delta\boldsymbol{\phi}_C \times \mathbf{r})|_{\gamma_\beta} = \delta\boldsymbol{\phi}_C \cdot d\boldsymbol{\phi}_C\mathbf{r} - \frac{1}{2}\delta\boldsymbol{\phi}_C d\boldsymbol{\phi}_C \cdot \mathbf{r} - \frac{1}{2}\mathbf{r} \cdot \delta\boldsymbol{\phi}_C d\boldsymbol{\phi}_C.$$

The second term in the expression for the second variation is obtained with the previously used expression for the "slip" along the surface:

$$\delta\boldsymbol{\phi}_C \times d\mathbf{r}|_{\phi_C} = \delta\boldsymbol{\phi}_C \times \mathbf{t}_\alpha d\gamma_\alpha.$$

The third term follows from the expression for the rigid body rotation:

$$d\mathbf{t}|_{\gamma_\beta}\delta\gamma_\alpha = d\boldsymbol{\phi}_C \times \mathbf{t}_\alpha\delta\gamma_\alpha.$$

Finally, the fourth term is obtained by differentiation along the surface coordinates:

$$d\mathbf{t}|_{\phi_C}\delta\gamma_\alpha = \frac{\partial\mathbf{t}_\alpha}{\partial S_\beta}d\gamma_\beta\delta\gamma_\alpha = \delta\gamma_\alpha\boldsymbol{\kappa}_{\alpha\beta}d\gamma_\beta,$$

where

$$\boldsymbol{\kappa}_{\alpha\beta} = \frac{\partial\mathbf{t}_\alpha}{\partial S_\beta} = \frac{\partial\mathbf{t}_\beta}{\partial S_\alpha} = \frac{\partial^2\mathbf{r}}{\partial S_\alpha \partial S_\beta}$$

is the surface curvature matrix.

Substitution of the last four expressions in the expression for the second variation yields

$$\mathbf{n}d\delta h = \delta\boldsymbol{\phi}_C \cdot d\boldsymbol{\phi}_C\mathbf{r} - \frac{1}{2}\delta\boldsymbol{\phi}_C d\boldsymbol{\phi}_C \cdot \mathbf{r} - \frac{1}{2}\mathbf{r} \cdot \delta\boldsymbol{\phi}_C d\boldsymbol{\phi}_C$$

$$+\delta\boldsymbol{\phi}_C \times \mathbf{t}_\alpha d\gamma_\alpha + d\boldsymbol{\phi}_C \times \mathbf{t}_\alpha\delta\gamma_\alpha + \delta\gamma_\alpha\boldsymbol{\kappa}_{\alpha\beta}d\gamma_\beta + \mathbf{t}_\alpha d\delta\gamma_\alpha.$$

As in the first variation, one can split the second variation into a normal and tangential components. For the normal component one finds

$$d\delta h = (\mathbf{n} \cdot \mathbf{r})\delta\boldsymbol{\phi}_C \cdot d\boldsymbol{\phi}_C - \frac{1}{2}\delta\boldsymbol{\phi}_C \cdot (\mathbf{nr} + \mathbf{rn}) \cdot d\boldsymbol{\phi}_C$$

$$+\delta\boldsymbol{\phi}_C\cdot(\mathbf{t}_\alpha\times\mathbf{n})d\gamma_\alpha+d\boldsymbol{\phi}_C\cdot(\mathbf{t}_\alpha\times\mathbf{n})\delta\gamma_\alpha+\delta\gamma_\alpha\mathbf{n}\cdot\boldsymbol{\kappa}_{\alpha\beta}d\gamma_\beta$$

and for the tangential components,

$$d\delta\gamma_\alpha=-(\mathbf{t}_\gamma\cdot\mathbf{r})\delta\boldsymbol{\phi}_C\cdot d\boldsymbol{\phi}_C+\frac{1}{2}\delta\boldsymbol{\phi}_C\cdot(\mathbf{t}_\gamma\mathbf{r}+\mathbf{r}\mathbf{t}_\gamma)\cdot d\boldsymbol{\phi}_C+\delta\gamma_\alpha\mathbf{t}_\gamma\cdot\boldsymbol{\kappa}_{\alpha\beta}d\gamma_\beta.$$

The expression involving $\boldsymbol{\kappa}_{\alpha\beta}$ can be simplified somewhat. Observe that $\mathbf{n}\cdot\mathbf{t}_\alpha=0$; hence,

$$\mathbf{n}\cdot\boldsymbol{\kappa}_{\alpha\beta}=\mathbf{n}\cdot\frac{\partial\mathbf{t}_\alpha}{\partial S_\beta}=-\mathbf{t}_\alpha\cdot\frac{\partial\mathbf{n}}{\partial S_\beta}=\mathbf{t}_\beta\cdot\frac{\partial\mathbf{n}}{\partial S_\alpha}.$$

Similarly

$$\mathbf{t}_\gamma\cdot\boldsymbol{\kappa}_{\alpha\beta}=\mathbf{t}_\gamma\cdot\frac{\partial\mathbf{t}_\alpha}{\partial S_\beta}=-\mathbf{t}_\alpha\cdot\frac{\partial\mathbf{t}_\gamma}{\partial S_\beta}.$$

If the local surface coordinate system is created by projection of a tangential Cartesian X–Y system onto the surface, it is readily established that the last terms vanish. Hence, we will assume that the last term in the second variation is zero. The final result is obtained by substitution of the expressions for the first-order variation of the slip in the expressions for the second variation. After some reordering and with $\kappa_{\alpha\beta}\equiv\mathbf{n}\cdot\boldsymbol{\kappa}_{\alpha\beta}$ this furnishes

$$d\delta h=(\delta\mathbf{x}_A-\delta\mathbf{x}_C)\cdot\mathbf{t}_\alpha\kappa_{\alpha\beta}\mathbf{t}_\beta\cdot(d\mathbf{x}_A-d\mathbf{x}_C)$$

$$+(\delta\mathbf{x}_A-\delta\mathbf{x}_C)\cdot\big(\mathbf{t}_\alpha\kappa_{\alpha\beta}(\mathbf{t}_\beta\times\mathbf{r})+\mathbf{t}_\alpha(\mathbf{t}_\beta\times\mathbf{n})\big)\cdot d\boldsymbol{\phi}_C$$

$$+\delta\boldsymbol{\phi}_C\cdot\big((\mathbf{t}_\alpha\times\mathbf{r})\kappa_{\alpha\beta}\mathbf{t}_\beta+(\mathbf{t}_\alpha\times\mathbf{n})\mathbf{t}_\beta\big)\cdot(d\mathbf{x}_A-d\mathbf{x}_C)$$

$$+\delta\boldsymbol{\phi}_C\cdot(\mathbf{t}_\alpha\times\mathbf{r})\kappa_{\alpha\beta}(\mathbf{t}_\beta\times\mathbf{r})\cdot d\boldsymbol{\phi}_C$$

$$+(\mathbf{n}\cdot\mathbf{r})\delta\boldsymbol{\phi}_C\cdot d\boldsymbol{\phi}_C-\frac{1}{2}\delta\boldsymbol{\phi}_C\cdot(\mathbf{n}\mathbf{r}+\mathbf{r}\mathbf{n})\cdot d\boldsymbol{\phi}_C$$

$$d\delta\gamma_\alpha=-(\mathbf{t}_\alpha\cdot\mathbf{r})\delta\boldsymbol{\phi}_C\cdot d\boldsymbol{\phi}_C+\frac{1}{2}\delta\boldsymbol{\phi}_C\cdot(\mathbf{t}_\alpha\mathbf{r}+\mathbf{r}\mathbf{t}_\alpha)\cdot d\boldsymbol{\phi}_C.$$

The first two terms of the expression for $d\delta h$ will only need to be included if slip occurs, whereas the expression for $d\delta\gamma_\alpha$ only needs to be taken into account if frictional forces are transmitted.

For dynamic applications we need the velocity and acceleration terms $\dot{h}$ and $\ddot{h}$ to calculate impact forces and impulses correctly. These terms are

$$\dot{h} = -\mathbf{n} \cdot (\dot{\mathbf{x}}_A - \dot{\mathbf{x}}_C - \dot{\boldsymbol{\phi}}_C \times \mathbf{r})$$

(this is the same form as the first variation of h); and

$$\begin{aligned}\ddot{h} = &- \mathbf{n} \cdot (\ddot{\mathbf{x}}_A - \ddot{\mathbf{x}}_C - \ddot{\boldsymbol{\phi}}_C \times \mathbf{r}) + \mathbf{n} \cdot \dot{\boldsymbol{\phi}}_C \, \mathbf{r} \cdot \dot{\boldsymbol{\phi}}_C - \mathbf{n} \cdot \mathbf{r} \, \dot{\boldsymbol{\phi}}_C \cdot \dot{\boldsymbol{\phi}}_C \\ &- (\dot{\mathbf{x}}_A - \dot{\mathbf{x}}_C - \dot{\boldsymbol{\phi}}_C \times \mathbf{r}) \cdot \Big(\dot{\boldsymbol{\phi}}_C \times \mathbf{n} + \frac{\partial \mathbf{n}}{\partial S^\alpha} \, \mathbf{t}_\alpha \cdot (\dot{\mathbf{x}}_A - \dot{\mathbf{x}}_C - \dot{\boldsymbol{\phi}}_C \times \mathbf{r})\Big).\end{aligned}$$

Reference

- “Contact formulations in Abaqus/Standard,” Section 38.1.1 of the Abaqus Analysis User’s Guide

5.2 Surface interactions

- “Contact pressure definition,” Section 5.2.1
- “Pressure and fluid flow in pore pressure contact,” Section 5.2.2
- “Coulomb friction,” Section 5.2.3
- “Thermal interface definition,” Section 5.2.4
- “Heat generation caused by frictional sliding,” Section 5.2.5
- “Heat generation caused by electrical current,” Section 5.2.6
- “Surface-based acoustic-structural medium interaction,” Section 5.2.7

5.2.1 CONTACT PRESSURE DEFINITION

Products: Abaqus/Standard Abaqus/Explicit

The contact modeling capabilities in Abaqus allow access to a library of "surface constitutive models." Part of this library in Abaqus/Standard is the definition of the contact pressure between two surfaces at a point, p, as a function of the "overclosure," h, of the surfaces (the interpenetration of the surfaces). Two models for $p = p(h)$ are provided as described below.

Hard contact

In this case

$$\begin{aligned} p &= 0 \quad \text{for } h < 0 \quad (\text{open}), \text{and} \\ h &= 0 \quad \text{for } p > 0 \quad (\text{closed}). \end{aligned}$$

The contact constraint is enforced with a Lagrange multiplier representing the contact pressure in a mixed formulation. The virtual work contribution is

$$\delta\Pi = \delta p h + p\delta h,$$

and the linearized form of the contribution is

$$d\delta\Pi = \delta p dh + dp\delta h.$$

Softened contact defined with an exponential pressure-overclosure relationship

This model provides an exponential p–h relationship, as shown in Figure 5.2.1–1.

The user defines an initial contact distance, c, and a typical pressure value, p^o, which is the pressure value at zero overclosure ($h = 0$). Then, we define

$$\begin{aligned} p &= 0 && \text{for } h \le -c, \\ p &= \frac{p^o}{(\exp(1)-1)}\left[\left(\frac{h}{c}+1\right)\left(\exp\left(\frac{h}{c}+1\right)-1\right)\right] && \text{for } h > -c, \end{aligned}$$

and

$$\begin{aligned} \frac{dp}{dh} &= 0 && \text{for } h \le -c, \\ \frac{dp}{dh} &= \frac{p^o}{(\exp(1)-1)}\left[\frac{1}{c}\left(\frac{h}{c}+2\right)\exp\left(\frac{h}{c}+1\right)-\frac{1}{c}\right] && \text{for } h > -c. \end{aligned}$$

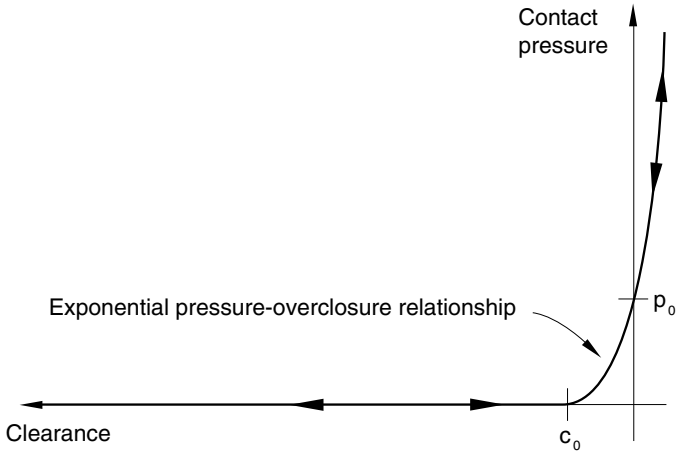


Figure 5.2.1–1 "Softened" pressure-overclosure relationship using an exponential law.

To avoid numerical difficulties at high penetration ($h > 6c$), a linearized pressure-overclosure relation with continuous slope is used.

Softened contact defined with a tabular pressure-overclosure relationship

The pressure-overclosure (p-h) relationship can be entered directly in tabular form as shown in Figure 5.2.1–2.

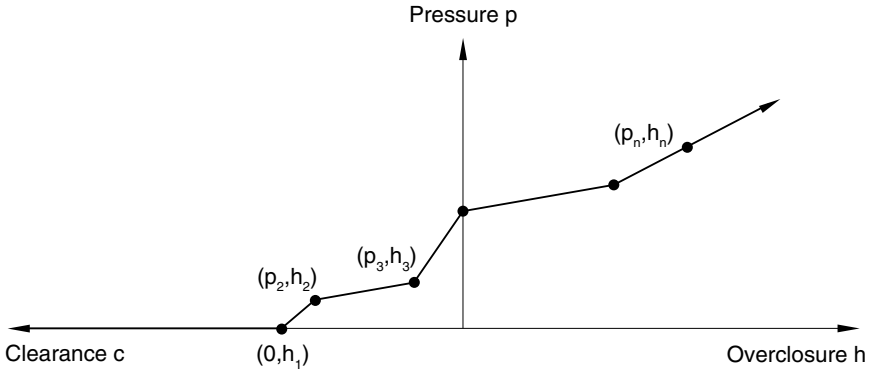


Figure 5.2.1–2 "Softened" pressure-overclosure relationship defined in tabular form.

Softened contact defined with a linear pressure-overclosure relationship

The linear pressure-overclosure relationship is similar to the tabular relationship except that the linear form requires only a single value to be input to define the slope and the curve always passes through the origin.

Softened contact implementation

A mixed formulation is used because the exponential stiffness associated with softened contact tends to slow down convergence or, due to the development of excessive contact stresses, may cause divergence. For the mixed formulation the virtual work contribution is

$$\delta\Pi = \delta p(h - \overline{h}) + p\,\delta h, \tag{5.2.1–1}$$

where p is the contact pressure, h is the actual overclosure, and $\overline{h}(p)$ is the overclosure associated with the contact pressure, p. A local Newton loop is used to calculate $\overline{h}$ for the current value of p. The linearized form of this contribution is

$$\mathrm{d}\delta\Pi = \delta p\,\mathrm{d}h + \mathrm{d}p\,\delta h - \frac{d\overline{h}}{dp}\delta p\,\mathrm{d}p, \tag{5.2.1–2}$$

where $d\overline{h}/dp = (dp/d\overline{h})^{-1}$ is evaluated for the overclosure $\overline{h}$. Since there is no term involving $\delta h dh$, there is a zero on the diagonal of the Jacobian. A zero on the diagonal is not desirable because it may lead to equation solver problems if a rigid body mode is constrained only by contact elements. Hence, a small reference stiffness k_0 is introduced by splitting the contact pressure as follows:

$$p = q + k_0\,(h + c) \qquad \text{and} \qquad \delta p = \delta q + k_0\,\delta h,$$

where q is a Lagrange multiplier and k_0 is a small reference stiffness (see Figure 5.2.1–1):

$$k_0 = \left.\frac{dp}{dh}\right|_{h=-0.9999c}.$$

Substituting for the pressure p in Equation 5.2.1–1 and Equation 5.2.1–2, we obtain

$$\delta\Pi = \delta q(h - \overline{h}) + \delta h\left[k_0(h - \overline{h}) + q + k_0(h + c)\right] \tag{5.2.1–3}$$

and

$$d\delta\Pi = \delta q(dh - d\overline{h}) + \delta h\left[k_0(dh - d\overline{h}) + dq + k_0 dh\right]. \tag{5.2.1–4}$$

Further,

$$d\overline{h} = \frac{d\overline{h}}{dp}dp = \frac{d\overline{h}}{dp}(dq + k_0 dh).$$

Therefore, substituting for $d\overline{h}$ in Equation 5.2.1–4,

$$d\delta\Pi = \delta q \left[(1 - \frac{d\overline{h}}{dp}k_0)dh - \frac{d\overline{h}}{dp}dq\right] + \delta h \left[(2 - \frac{d\overline{h}}{dp}k_0)k_0 dh + (1 - \frac{d\overline{h}}{dp}k_0)dq\right].$$

Thus, the corresponding system of equations in matrix form is

$$\begin{bmatrix} (2 - \frac{d\overline{h}}{dp}k_0)k_0 & 1 - \frac{d\overline{h}}{dp}k_0 \\ 1 - \frac{d\overline{h}}{dp}k_0 & -\frac{d\overline{h}}{dp} \end{bmatrix} \begin{Bmatrix} dh \\ dq \end{Bmatrix} = - \begin{Bmatrix} q + k_0(h + c) + k_0(h - \overline{h}) \\ h - \overline{h} \end{Bmatrix}.$$

In the mixed formulation the difference between the actual and the calculated overclosure $h - \overline{h}$ will go to zero as part of the iterative solution process. The difference must be sufficiently small to obtain an accurate solution. The admissible error in $h - \overline{h}$ is set to $0.005c$ for $p \geq p^0$. For $0 \leq p \leq p^0$ the admissible error is interpolated linearly between $0.005c$ and $0.1c$, where $0.1c$ represents the tolerance level at $p = 0.0$; alternatively, the tolerances can be specified by the user as part of the solution controls.

Viscous damping option

In addition to the surface constitutive models described above, where the contact pressure is a function of the surface overclosure, Abaqus/Standard allows for the definition of a "viscous" pressure that is proportional to the relative velocity, $\dot{h}$, at which the surfaces approach or separate from each other. This option is intended for the regularization of snap-through problems involving contact where convergence difficulties arise due to the sudden violation of contact constraints.

The damping pressure, f, is given by

$$f = f(h, \dot{h}) = \mu(h)\dot{h}\,,$$

where μ is the damping coefficient. This coefficient is specified as a function of the overclosure, h, as follows:

$$\mu(h) = \begin{cases} 0 & \text{for } h < -c_0 \\ \mu_0(h + c_0)/(1 - \eta)c_0 & \text{for } -c_0 \leq h < -\eta c_0, \\ \mu_0 & \text{for } h \geq -\eta c_0 \end{cases}$$

where μ_0 is the value of the damping coefficient at zero overclosure and η is the fraction of the overclosure interval $[-c_0, 0]$ over which the damping coefficient is equal to μ_0.

The virtual work contribution associated with the damping pressure is

$$\delta\Pi = f\delta h\,.$$

The contribution to the stiffness matrix for the Newton solution is given by the linearized form of the virtual work contribution:

$$\begin{aligned} \mathrm{d}\delta\Pi = \delta h \mathrm{d}f &= \delta h \left(\frac{\partial f}{\partial h} + \frac{\partial f}{\partial \dot{h}} \frac{\partial \dot{h}}{\partial h} \right) \mathrm{d}h \\ &= \delta h \left(\frac{d\mu}{dh} \dot{h} + \mu \frac{\partial \dot{h}}{\partial h} \right) \mathrm{d}h, \end{aligned}$$

where

$$\frac{d\mu}{dh} = \begin{cases} 0 & \text{for } h < -c_0 \\ \mu_0/(1-\eta)c_0 & \text{for } -c_0 \le h < -\eta c_0 \\ 0 & \text{for } h \ge -\eta c_0 \end{cases}.$$

In static analysis the velocity is defined as the displacement increment divided by the time increment. Therefore, $\dot{h} = \Delta h / \Delta t$, and the stiffness contribution reduces to

$$\mathrm{d}\delta\Pi = \delta h \left(\frac{d\mu}{dh} \frac{\Delta h}{\Delta t} + \frac{\mu}{\Delta t} \right) \mathrm{d}h \,.$$

The previous expression also applies to dynamics if the backward Euler time integration operator is used. In the case of dynamics with the Hilber-Hughes-Taylor time integration operator, the stiffness contribution can be written as

$$\mathrm{d}\delta\Pi = \delta h \left(\frac{d\mu}{dh} \dot{h} + \frac{\gamma \mu}{\beta \Delta t} \right) \mathrm{d}h \,,$$

where γ and β are the Hilber-Hughes-Taylor time integration operator parameters. The viscous damping option cannot be used in a Riks analysis since velocity is not defined.

Reference

- "Contact pressure-overclosure relationships," Section 37.1.2 of the Abaqus Analysis User's Guide

5.2.2 PRESSURE AND FLUID FLOW IN PORE PRESSURE CONTACT

Product: Abaqus/Standard

Abaqus/Standard provides a surface-based capability for modeling fully saturated porous media. The surface-based capability can be used for small or finite sliding. Only flow normal to the surface is considered; tangential flow cannot be modeled.

The pore fluid constraints on the contact interface for infinite interface permeability

Let p_1 and p_2 be the pore pressures at the two sides of the interface. If the interface permeability is considered infinite (i.e., no resistance to fluid flow), it is required at all times that the pore pressures on opposite sides be equal:

$$p_2 - p_1 = 0.$$

Similarly, let q_1 and q_2 be the volume flow rate densities normal to the interface at the two sides, and let $\dot{h} = v_2 - v_1 = \mathbf{n} \cdot (\mathbf{v_2} - \mathbf{v_1})$ be the relative velocity of the two sides in the direction of the interface normal.

It is assumed that the interface is filled with fluid up to a separation threshold. Hence, continuity requires that

$$\begin{aligned} q_1 + q_2 &= -\dot{h} \quad \text{for } -h < d_{threshold}, \\ q_1 + q_2 &= 0 \quad \text{for } -h \geq d_{threshold}, \end{aligned}$$

whereas the difference corresponds to twice the flow across the interface.

$$q_2 - q_1 = 2q_{across}$$

is undetermined and is to be treated as an independent variable in the case of infinite permeability. Inversion of these equations yields

$$\begin{aligned} q_1 &= -\frac{1}{2}\dot{h} - q_{across}, \\ q_2 &= -\frac{1}{2}\dot{h} + q_{across}. \end{aligned}$$

The pore fluid constraints on the contact interface for finite interface permeability

For finite gap permeability, k, the volume flow rate density directly across the interface is

$$q_{across} = k(p_2 - p_1).$$

It is assumed that the interface is filled with fluid up to a separation threshold. Hence, continuity requires that

$$q_1 + q_2 = -\dot{h} \text{ for } -h < d_{threshold},$$
$$q_1 + q_2 = 0 \quad \text{for } h \geq d_{threshold}.$$

Volume flow rate densities normal to the interface at the two sides of the interface are

$$q_1 = -\eta \frac{k_1}{k_1 + k_2} \dot{h} - q_{across},$$
$$q_2 = -\eta \frac{k_2}{k_1 + k_2} \dot{h} + q_{across},$$

where

$$\eta = 1 \quad \text{if } -h < d_{threshold},$$
$$\eta = 0 \quad \text{if } -h \geq d_{threshold}.$$

k_1 and k_2 are underlying material permeabilities.

The transient equations

The contribution of the interfacial virtual work equation and its linearized form are first obtained in the general form including finite sliding. The equations are then specialized to the various formulations implemented in Abaqus.

Since we want to achieve force and volume flow rate equilibrium at each side of the interface, as well as obtain continuity in the pore pressures, we add the following integral to the virtual work equation:

$$\delta\Pi^{ipp} = \int_A [q_1 \delta p_1 + q_2 \delta p_2 - p_2 \delta v_2 + p_1 \delta v_1 - (p_1 - p_2)\delta\lambda] dA,$$

where $\delta\lambda$ is an arbitrary Lagrange multiplier and A is the interface area. Eliminating q_1 and q_2 and using a suitable choice for $\delta\lambda$,

$$\delta\lambda = \frac{1}{2}(\delta\hat{q} + \delta v_2 + \delta v_1),$$

we obtain

$$\delta\Pi^{ipp} = -\int_A \left[\dot{h}\delta\bar{p} + \bar{p}\delta\dot{h} + \hat{q}\delta\hat{p} + \hat{p}\delta\hat{q} \right] dA,$$

where

$$\bar{p} = \frac{p_1 + p_2}{2};\ \hat{p} = \frac{p_1 - p_2}{2};\ \delta\bar{p} = \frac{\delta p_1 + \delta p_2}{2};\ \delta\hat{p} = \frac{\delta p_1 - \delta p_2}{2}.$$

Since Abaqus uses displacements (not velocities) and fluxes integrated over Δt, the equation can be multiplied by Δt to obtain

$$\delta\Pi^{ipp} = -\int_A \left[\Delta h \delta\bar{p} + \bar{p}\delta h + \Delta t(\hat{q}\delta\hat{p} + \hat{p}\delta\hat{q})\right] dA,$$

where Δh is the incremental change in h in the direction of the interface normal. Linearization yields

$$\mathrm{d}\delta\Pi^{ipp} = -\int_A \left[dh\delta\bar{p} + \Delta h d\delta\bar{p} + d\bar{p}\delta h + \bar{p}d\delta h + \Delta t(d\hat{q}\delta\hat{p} + \hat{q}d\delta\hat{p} + d\hat{p}\delta\hat{q} + \hat{p}d\delta\hat{q})\right] dA.$$

Closed contact

If the two sides are locally in contact, $\Delta h = 0$ and $d = 0$; therefore, the virtual work simplifies to

$$\delta\Pi^{ipp} = -\int_A \left[\bar{p}\delta h + \Delta t(\hat{q}\delta\hat{p} + \hat{p}\delta\hat{q})\right] dA.$$

Similarly, the linearized form simplifies to

$$\mathrm{d}\delta\Pi^{ipp} = -\int_A \left[d\bar{p}\delta h + \bar{p}d\delta h + \Delta t(d\hat{q}\delta\hat{p} + \hat{q}d\delta\hat{p} + d\hat{p}\delta\hat{q} + \hat{p}d\delta\hat{q})\right] dA.$$

The steady-state equations

For steady-state analysis the transient terms can be omitted, and the terms involving fluid flow are written in rate form. In this case we can assume that the interface displacements vanish, which leads to the simplified virtual work contribution

$$\delta\Pi^{ipp}_{ss} = -\int_A \left[(\bar{p}\delta h + \hat{q}\delta\hat{p} + \hat{p}\delta\hat{q})\right] dA$$

and the linearized form

$$\mathrm{d}\delta\Pi^{ipp}_{ss} = -\int_A \left[d\bar{p}\delta h + \bar{p}d\delta h + d\hat{q}\delta\hat{p} + \hat{q}d\delta\hat{p} + d\hat{p}\delta\hat{q} + \hat{p}d\delta\hat{q}\right] dA.$$

Small sliding

When the small-sliding contact formulation is used, the terms $d\delta\bar{p}$, $\bar{p}d\delta h$, $d\delta\hat{p}$, and $\hat{p}d\delta\hat{q}$ in the linearized virtual work equation will vanish.

Reference

- "Pore fluid contact properties," Section 37.4.1 of the Abaqus Analysis User's Guide

5.2.3 COULOMB FRICTION

Products: Abaqus/Standard Abaqus/Explicit

An extended version of the classical isotropic Coulomb friction model is provided in Abaqus for use with all contact analysis cababilities. The extensions include an additional limit on the allowable shear stress, anisotropy, and the definition of a "secant" friction coefficient.

The standard Coulomb friction model assumes that no relative motion occurs if the equivalent frictional stress

$$\tau_{eq} = \sqrt{\tau_1^2 + \tau_2^2}$$

is less than the critical stress, τ_{crit}, which is proportional to the contact pressure, p, in the form

$$\tau_{crit} = \mu p,$$

where μ is the friction coefficient that can be defined as a function of the contact pressure, p; the slip rate, $\dot{\gamma}_{eq}$; the average surface temperature at the contact point; and the average field variables at the contact point. Rate-dependent friction cannot be used in a static Riks analysis since velocity is not defined. In Abaqus it is possible to put a limit on the critical stress:

$$\tau_{crit} = min(\mu p, \tau_{max}),$$

where τ_{max} is user-specified. If the equivalent stress is at the critical stress ($\tau_{eq} = \tau_{crit}$), slip can occur. If the friction is isotropic, the direction of the slip and the frictional stress coincide, which is expressed in the form

$$\frac{\tau_i}{\tau_{eq}} = \frac{\dot{\gamma}_i}{\dot{\gamma}_{eq}},$$

where $\dot{\gamma}_i$ is the slip rate in direction i and $\dot{\gamma}_{eq}$ is the magnitude of the slip velocity,

$$\dot{\gamma}_{eq} = \sqrt{\dot{\gamma}_1^2 + \dot{\gamma}_2^2}.$$

As will be shown later, the same laws can be used for anisotropic friction after some simple transformations.

The above behavior can be modeled in Abaqus/Standard in two different ways. By default, the condition of no relative motion is approximated by stiff elastic behavior. The stiffness is chosen such that the relative motion from the position of zero shear stress is bounded by a value γ_{crit}. (In the Abaqus Analysis User's Guide γ_{crit} is referred to as the allowable maximum elastic slip.) The critical slip value, γ_{crit}, can be specified by the user. If it is not specified by the user, γ_{crit} is, by default, set to 0.5% of the average length of all contact elements in the model. It is worth noting that this approximate implementation method can also be considered an implementation of a nonlocal friction model; that is, a friction model for which the Coulomb condition is not applied pointwise but weighted over a small area with a so-called mollifying function (Zhong, 1989). See Oden and Pires (1983) for further discussion of nonlocal friction models.

Optionally, the relative motion in the absence of slip can be made exactly zero with the use of a Lagrange multiplier formulation. Although this procedure appears attractive because of the exact sticking constraint, it has two disadvantages:

1. The additional Lagrange multipliers increase the cost of the analysis.
2. The presence of rigid constraints tends to slow or sometimes prevent convergence of the Newton solution scheme used in Abaqus/Standard. This is likely to occur in areas where contact conditions change.

A special case of friction in Abaqus/Standard is so-called rough friction, where it is assumed that there is no bound on the shear stress; that is, no relative motion can occur as long as the surfaces are in contact. Rough friction is implemented with the Lagrange multiplier method.

In Abaqus/Explicit the relative motion in the absence of slip is always equal to zero if the kinematic contact algorithm is used with hard tangential surface behavior; at the end of each increment the positions of the nodes on the contact surfaces are adjusted so that the relative motion is zero. With the penalty contact algorithm in Abaqus/Explicit the relative motion in the absence of slip is equal to the friction force divided by the penalty stiffness.

Elastic stick formulation

In the elastic stick formulation in Abaqus/Standard, the "elastic" tangential slip γ_i^{el} is defined as the *reversible* relative tangential motion from the point of zero shear stress. The elastic slip is related to the interface shear stress with the relation

$$\tau_i = k_s \gamma_i^{el},$$

where k_s is the (current) "stiffness in stick," which follows from the relation

$$k_s = \frac{\tau_{crit}}{\gamma_{crit}}.$$

Since τ_{crit} may be dependent on contact pressure, slip rate, average surface temperature at the contact point, and field variables, k_s may change during the analysis. The behavior remains elastic as long as the equivalent stress does not exceed the critical stress; hence,

$$\gamma_i^{el}(t + \Delta t) = \gamma_i^{el}(t) + \Delta\gamma_i.$$

Consistent linearization of this expression yields

$$\begin{aligned} \mathrm{d}\tau_i &= k_s \, \mathrm{d}\gamma_i && \text{if} \quad \tau_{crit} = \tau_{max}, \\ \mathrm{d}\tau_i &= k_s \, \mathrm{d}\gamma_i + \frac{\tau_i}{\tau_{crit}} \left(\mu + p \frac{\partial \mu}{\partial p} \right) \mathrm{d}p && \text{if} \quad \tau_{crit} = \mu p. \end{aligned} \tag{5.2.3–1}$$

The contributions from the contact pressure are nonsymmetric for the second case. Since the slip rate is zero in the elastic stick formulation, derivatives with respect to the slip velocity are not needed.

The above expressions hold if the equivalent shear stress remains less than the critical stress. If the equivalent stress exceeds the critical stress, slip must be taken into consideration so that the condition

$\tau_{eq} = \tau_{crit}$ is satisfied. Let the starting situation be characterized by the elastic slip $\overline{\gamma}_i^{el}$. The critical stress at the end of the increment follows from the contact pressure, p, and the slip rate, $\dot{\gamma}_{eq}^{sl}$.

Let the (as yet unknown) elastic slip at the end of the increment be γ_i^{el} and the slip increment be $\Delta\gamma_i^{sl}$. Consistency requires that

$$\Delta\gamma_i = \gamma_i^{el} - \overline{\gamma}_i^{el} + \Delta\gamma_i^{sl},$$

and the shear stress at the end of the increment follows from the elasticity relation,

$$\tau_i = k_s\, \gamma_i^{el} = \frac{\tau_{crit}}{\gamma_{crit}} \gamma_i^{el}.$$

The slip increment is related to the stress at the end of the increment with the backward difference approach:

$$\Delta\gamma_i^{sl} = \frac{\tau_i}{\tau_{crit}} \Delta\gamma_{eq}^{sl}.$$

With these equations and the critical stress equality $\tau_{eq} = \tau_{crit}$, it is possible to solve for γ_i^{el}, $\Delta\gamma_i^{sl}$, and τ_i. Elimination of γ_i^{el} and $\Delta\gamma_i^{sl}$ from the above equations yields,

$$\Delta\gamma_i = \frac{\tau_i}{\tau_{crit}} \gamma_{crit} - \overline{\gamma}_i^{el} + \frac{\tau_i}{\tau_{crit}} \Delta\gamma_{eq}^{sl}$$

or

$$\tau_i = \frac{\overline{\gamma}_i^{el} + \Delta\gamma_i}{\gamma_{crit} + \Delta\gamma_{eq}^{sl}} \tau_{crit}.$$

It is convenient to define the "elastic predictor strain"

$$\gamma_i^{pr} \stackrel{\text{def}}{=} \overline{\gamma}_i^{el} + \Delta\gamma_i,$$

which simplifies the expression for the stress to

$$\tau_i = \frac{\gamma_i^{pr}}{\gamma_{crit} + \Delta\gamma_{eq}^{sl}} \tau_{crit}.$$

Substitution in the critical stress equality yields

$$\Delta\gamma_{eq}^{sl} = \gamma_{eq}^{pr} - \gamma_{crit},$$

where

$$\gamma_{eq}^{pr} \stackrel{\text{def}}{=} \sqrt{(\gamma_1^{pr})^2 + (\gamma_2^{pr})^2}.$$

Substitution in the expression for τ_i and introduction of the normalized slip direction $n_i = \gamma_i^{pr}/\gamma_{eq}^{pr}$ furnishes the final result

$$\tau_i = n_i \tau_{crit}. \tag{5.2.3–2}$$

Here τ_{crit} is a function of the slip rate, which is obtained with

$$\dot{\gamma}_{eq}^{sl} = \Delta\gamma_{eq}^{sl}/\Delta t,$$

where Δt is the time increment in a static analysis. In the case of dynamics with the Hilber-Hughes-Taylor time integration operator, Δt is scaled by the Hilber-Hughes-Taylor time integration operator parameters, γ and β.

For the iterative solution scheme this equation must be linearized. Some straightforward algebra yields

$$\begin{aligned} \mathrm{d}\tau_i =& \frac{\mathrm{d}\gamma_i^{pr}}{\gamma_{eq}^{pr}}\tau_{crit} - n_i n_j \frac{\tau_{crit}}{\gamma_{eq}^{pr}}\mathrm{d}\gamma_j^{pr} + n_i\,\mathrm{d}\tau_{crit} \\ =& \left(\delta_{ij} - n_i n_j\right)\frac{\tau_{crit}}{\gamma_{eq}^{pr}}\mathrm{d}\gamma_j^{pr} + n_i\left(\frac{\partial\tau_{crit}}{\partial p}\mathrm{d}p + \frac{\partial\tau_{crit}}{\partial\dot{\gamma}_{eq}^{sl}}\mathrm{d}\dot{\gamma}_{eq}^{sl}\right). \end{aligned}$$

With the expression for the equivalent slip the final result is

$$\begin{aligned} \mathrm{d}\tau_i &= \left(\delta_{ij} - n_i n_j\right)\frac{\tau_{crit}}{\gamma_{eq}^{pr}}\mathrm{d}\gamma_j && \text{if} \quad \tau_{\mathrm{crit}} = \tau_{\max}; \\ \mathrm{d}\tau_i &= \left(\delta_{ij} - n_i n_j\right)\frac{\tau_{crit}}{\gamma_{eq}^{pr}}\mathrm{d}\gamma_j + n_i\left(\mu + p\frac{\partial\mu}{\partial p}\right)\mathrm{d}p + n_i n_j \frac{p}{\Delta t}\frac{\partial\mu}{\partial\dot{\gamma}_{eq}}\mathrm{d}\gamma_j && \text{if} \quad \tau_{\mathrm{crit}} = \mu\mathrm{p}. \end{aligned} \tag{5.2.3–3}$$

In this case the unsymmetric terms may have a strong effect on the speed of convergence of the Newton scheme. Hence, use of the unsymmetric equations solver is strongly recommended for the analysis of problems in which sliding friction occurs. In the case of dynamics with the Hilber-Hughes-Taylor time integration operator, Δt is scaled by the Hilber-Hughes-Taylor time integration operator parameters.

Frictional stress due to forced sliding in static analysis

If the user specifies different velocities for two bodies in contact as a predefined field, it is assumed that the slip velocity follows from the prescribed motion and is independent of the displacement values. The frictional stresses then follow from

$$\tau_i = \tau_{eq}\frac{\dot{\gamma}_i}{\dot{\gamma}_{eq}},$$

where

$$\tau_{eq} = \mu p,$$

and

$$\dot{\gamma}_{eq} = \sqrt{(\dot{\gamma}_1)^2 + (\dot{\gamma}_2)^2}.$$

$\dot{\gamma}_i$ are the tangential slip velocities derived from the user-defined fields ($i = 1, 2$).

Friction contributions for complex eigenvalue extraction analysis

At the contact nodes at which the velocity differential is imposed, the linearized shear stress can be expressed in the following form:

$$\mathrm{d}\tau_i = (\delta_{ij} - n_i n_j)\,\frac{\tau_{crit}}{\gamma_{eq}^{pr}}\mathrm{d}\dot{\gamma}_j + n_i\left(\mu + p\frac{\partial \mu}{\partial p}\right)\mathrm{d}p + n_i n_j p\frac{\partial \mu}{\partial \dot{\gamma}_{eq}}\mathrm{d}\dot{\gamma}_j.$$

In the equation above the first term is generated by the friction forces acting in the direction perpendicular to the direction of slip. The third term exists if the friction coefficient depends on velocity. In the complex eigenvalue extraction analysis both terms contribute to the damping matrix. The second term contributes to the stiffness matrix.

Viscous stick formulation for steady-state transport

For steady-state transport a viscous stick formulation is used. In this case the "viscous" tangential slip rate, $\dot{\gamma}_i^v$, is related to the interface shear stress by

$$\tau_i = \kappa_s \dot{\gamma}_i^v,$$

where κ_s is the "stick viscosity," which follows from the relation

$$\kappa_s = \frac{\tau_{crit}}{\dot{\gamma}_{crit}}.$$

The allowable viscous slip is defined as a fraction of the relative tangential velocity

$$\dot{\gamma}_{crit} = 2F_f \omega R,$$

where F_f is a user-defined slip tolerance, ω is the angular spinning velocity, and R is the radius of a point on the contact surface in the undeformed configuration.

Exact stick formulation

In Abaqus/Standard Lagrange multipliers are used to enforce exact sticking conditions. A constraint term enforced with Lagrange multipliers δq_i is added to the virtual work statement:

$$\delta\Pi^* = \int_S (\tau_i \delta\gamma_i + \Delta\gamma_i \delta q_i) dS, \qquad (5.2.3\text{–}4)$$

where δq_i is a Lagrange multiplier. By taking the rate of change of Equation 5.2.3–4, the rate of virtual work is obtained:

$$\mathrm{d}\delta\Pi^* = \int_S (\mathrm{d}\tau_i \delta\gamma_i + \mathrm{d}\gamma_i \delta q_i + \tau_i \mathrm{d}\delta\gamma_i) dS, \tag{5.2.3–5}$$

where the last term results from a nonlinear relation between nodal displacements and relative motion γ_i. This term is nonzero only for contact with finite sliding (see "Finite-sliding interaction between deformable bodies," Section 5.1.2, and "Finite-sliding interaction between a deformable and a rigid body," Section 5.1.3).

To obtain a complete formulation, a relationship between $\Delta\gamma_i$, q_i, and τ_i must be defined. For sticking conditions a suitable relation is

$$\tau_i = q_i + k_0 \Delta\gamma_i,$$

with k_0 a reference stiffness selected internally in Abaqus. The reference stiffness can be considered to model a spring that is in parallel to the sticking constraint. Since the constraint prevents relative motion, the reference stiffness has no physical significance and is added only to eliminate zero terms on the diagonal of the stiffness matrix that could cause equation solver problems. The variational and rate forms of the equation for τ_i are readily obtained as

$$\delta\tau_i = \delta q_i + k_0 \delta\gamma_i \qquad \text{and} \qquad \mathrm{d}\tau_i = \mathrm{d}q_i + k_0 \mathrm{d}\gamma_i.$$

Substitution in Equation 5.2.3–5 yields

$$\mathrm{d}\delta\Pi^* = \int_S (k_0 \delta\gamma_i d\gamma_i + \delta\gamma_i \mathrm{d}q_i + \delta q_i \mathrm{d}\gamma_i + \tau_i \mathrm{d}\delta\gamma_i) dS. \tag{5.2.3–6}$$

If the element is slipping, $\Delta\gamma_i$ can take an arbitrary value. Consequently the terms involving δq_i in Equation 5.2.3–4 and Equation 5.2.3–5 vanish. With the backward difference method, the frictional stress is obtained as

$$\tau_i = \tau_{crit} \frac{\Delta\gamma_i}{\Delta\gamma_{eq}}, \tag{5.2.3–7}$$

and the linearized form is

$$\begin{aligned} \mathrm{d}\tau_i &= \frac{\tau_{crit}}{\Delta\gamma_{eq}} (\delta_{ij} - n_i n_j) \mathrm{d}\gamma_j && \text{if} \quad \tau_{\mathrm{crit}} = \tau_{\max}, \\ \mathrm{d}\tau_i &= \frac{\tau_{crit}}{\Delta\gamma_{eq}} (\delta_{ij} - n_i n_j) \mathrm{d}\gamma_j + n_i \left(\mu + p \frac{\partial\mu}{\partial p} \right) dp + n_i n_j \frac{p}{\Delta t} \frac{\partial\mu}{\partial\dot{\gamma}_{eq}} \mathrm{d}\gamma_j && \text{if} \quad \tau_{\mathrm{crit}} = \mu \mathrm{p}. \end{aligned} \tag{5.2.3–8}$$

The rate of virtual work can, hence, be written in the form

$$
\begin{aligned}
\mathrm{d}\delta\Pi^* &= \int_S \Big[\frac{\tau_{crit}}{\Delta\gamma_{eq}}(\delta_{ij} - n_i n_j)\delta\gamma_i \mathrm{d}\gamma_j + \tau_i \mathrm{d}\delta\gamma_i \Big] dS \qquad \text{if} \quad \tau_{\mathrm{crit}} = \tau_{\max}, \\
\mathrm{d}\delta\Pi^* &= \int_S \Big[\frac{\tau_{crit}}{\Delta\gamma_{eq}}(\delta_{ij} - n_i n_j)\delta\gamma_i \mathrm{d}\gamma_j + \left(\mu + p\frac{\partial\mu}{\partial p}\right) n_i \delta\gamma_i \mathrm{d}p \\
&\quad + \frac{p}{\Delta t}\frac{\partial\mu}{\partial\dot{\gamma}_{eq}} n_i n_j \delta\gamma_i \mathrm{d}\gamma_j + \tau_i \mathrm{d}\delta\gamma_i \Big] dS \qquad \text{if} \quad \tau_{\mathrm{crit}} = \mu \mathrm{p}.
\end{aligned}
\tag{5.2.3–9}
$$

Observe that for two-dimensional problems the term involving $(\delta_{ij} - n_i n_j)$ vanishes. In the case of dynamics with the Hilber-Hughes-Taylor time integration operator, Δt is scaled by the Hilber-Hughes-Taylor time integration operator parameters.

To determine whether an element sticks or slips, the following tests are used. If the element is currently sticking, it is checked whether the magnitude of the frictional stress exceeds the critical value. Hence, if

$$\tau_{eq} > \tau_{crit},$$

the state is changed to slipping.

If the element is slipping, the direction of slip is compared with the frictional stress $\tau_i(t)$ applied at the start of the iteration. If at the end of the iteration

$$\Delta\gamma_i \tau_i(t) < 0,$$

the state is changed to sticking.

Anisotropic friction

In anisotropic friction the friction coefficients in the two (principal) directions are different; hence, the critical shear stresses are different:

$$\tau_1^{crit} = \mu_1 p \qquad \text{and} \qquad \tau_2^{crit} = \mu_2 p.$$

The two critical shear stresses are at the extreme points of the friction ellipse, given by the equation

$$\left(\frac{\tau_1}{\mu_1}\right)^2 + \left(\frac{\tau_2}{\mu_2}\right)^2 = p^2.$$

This suggests definition of the scaled shear stresses,

$$\overline{\tau_1} \stackrel{\mathrm{def}}{=} \frac{\overline{\mu}}{\mu_1}\tau_1 \qquad \text{and} \qquad \overline{\tau_2} \stackrel{\mathrm{def}}{=} \frac{\overline{\mu}}{\mu_2}\tau_2, \tag{5.2.3–10}$$

where $\overline{\mu}$ is the average friction coefficient defined by

$$\overline{\mu} = \sqrt{\frac{1}{2}(\mu_1^2 + \mu_2^2)}.$$

Hence, the (scaled) equivalent frictional stress is defined by

$$\overline{\tau}_{eq} = \sqrt{\overline{\tau_1}^2 + \overline{\tau}_2^2},$$

and the average critical stress

$$\overline{\tau}_{crit} = min(\overline{\mu}p, \tau_{max}),$$

where τ_{max} is the user-specified limiting shear stress, which is applied to the scaled shear stresses. No slip will occur if $\overline{\tau}_{eq} < \overline{\tau}_{crit}$, and slip can occur if $\overline{\tau}_{eq} = \overline{\tau}_{crit}$.

If slip occurs, it is assumed that the direction of slip is governed by an associated slip law:

$$\dot{\gamma}_1 = \lambda \frac{\partial \overline{\tau}_{eq}}{\partial \tau_1} = \lambda \frac{\partial \overline{\tau}_{eq}}{\partial \overline{\tau}_1} \frac{d\overline{\tau}_1}{d\tau_1} = \lambda \frac{\overline{\tau}_1}{\overline{\tau}_{eq}} \frac{\overline{\mu}}{\mu_1} \qquad \text{and} \qquad \dot{\gamma}_2 = \lambda \frac{\partial \overline{\tau}_{eq}}{\partial \tau_2} = \lambda \frac{\partial \overline{\tau}_{eq}}{\partial \overline{\tau}_2} \frac{d\overline{\tau}_2}{d\tau_2} = \lambda \frac{\overline{\tau}_2}{\overline{\tau}_{eq}} \frac{\overline{\mu}}{\mu_2}.$$

Hence, scaled rates of slip are defined by

$$\dot{\overline{\gamma}}_1 \overset{\text{def}}{=} \frac{\mu_1}{\overline{\mu}} \dot{\gamma}_1 \qquad \text{and} \qquad \dot{\overline{\gamma}}_2 \overset{\text{def}}{=} \frac{\mu_2}{\overline{\mu}} \dot{\gamma}_2, \tag{5.2.3–11}$$

and the (scaled) equivalent slip rate

$$\dot{\overline{\gamma}}_{eq} = \sqrt{\dot{\overline{\gamma}}_1^2 + \dot{\overline{\gamma}}_2^2}.$$

As a result, the direction of the scaled slip and the scaled shear stress coincide:

$$\frac{\overline{\tau}_i}{\overline{\tau}_{eq}} = \frac{\dot{\overline{\gamma}}_i}{\dot{\overline{\gamma}}_{eq}}.$$

Since these equations have exactly the same structure as the equations for isotropic friction, the same solution algorithms can be used to get the incremental and linearized equations in terms of the scaled stress and slip. The actual stresses are readily obtained from the scaled stresses with Equation 5.2.3–10, and the linearized equations are scaled with the same factors.

In the anisotropic elastic stick formulation, it is assumed that the stiffness in stick in terms of the *scaled* stress and slip is constant:

$$\overline{k}_s = \frac{\overline{\tau}_{crit}}{\gamma_{crit}};$$

hence, the scaled stress is related to the scaled elastic slip by the relation

$$\overline{\tau}_i = \overline{k}_s \overline{\gamma}_i^{el}.$$

In terms of the actual stress and strain in the 1- and 2-directions, this implies

$$\tau_1 = k_1 \gamma_1^{el} \qquad \text{and} \qquad \tau_2 = k_2 \gamma_2^{el},$$

with

$$k_1 = \left(\frac{\mu_1}{\overline{\mu}}\right)^2 \overline{k}_s, \qquad k_2 = \left(\frac{\mu_2}{\overline{\mu}}\right)^2 \overline{k}_s.$$

Hence, the anisotropy in terms of the stiffness in stick is more pronounced than the anisotropy in terms of critical stress.

Viscous damping

In addition to the friction models described above, Abaqus allows for the definition of a "viscous" shear stress τ_i^v that is proportional to the relative tangential velocity $\dot{\gamma}_i$. This viscous damping in the tangential direction occurs if

- damping in the contact direction is included and the tangent fraction is nonzero; or
- contact controls are used to stabilize rigid body modes automatically in multi-body contact analysis.

The viscous damping stress is proportional to the tangential damping coefficient μ^v, which is a function of the overclosure as follows:

$$\mu^v(h) = \begin{cases} 0 & \text{for } h < -c_0 \\ \mu_0^v (h + c_0)/(1-\eta)c_0 & \text{for } -c_0 \le h < -\eta c_0 \\ \mu_0^v & \text{for } h \ge -\eta c_0, \end{cases}$$

where μ_0^v is the value of the tangential damping coefficient at zero overclosure and η is the fraction of the overclosure interval $[-c_0, 0]$ over which the damping coefficient is equal to μ_0^v.

The virtual work contribution associated with the viscous shear stress is

$$\delta\Pi = \tau_i^v \delta\gamma_i \, .$$

The contribution to the stiffness matrix for the Newton solution is given by the linearized form of the virtual work contribution:

$$\begin{aligned} \mathrm{d}\delta\Pi = \delta\gamma_i \mathrm{d}\tau_i^v &= \delta\gamma_i \left(\frac{\partial \tau_i^v}{\partial h} \mathrm{d}h + \frac{\partial \tau_i^v}{\partial \dot{\gamma}_j} \frac{\partial \dot{\gamma}_j}{\partial \gamma_k} \mathrm{d}\gamma_k \right) \\ &= \delta\gamma_i \frac{d\mu^v}{dh} \dot{\gamma}_i \mathrm{d}h + \delta\gamma_i \mu^v \frac{\partial \dot{\gamma}_i}{\partial \gamma_k} \mathrm{d}\gamma_k, \end{aligned}$$

where

$$\frac{d\mu^v}{dh} = \begin{cases} 0 & \text{for } h < -c_0 \\ \mu_0^v/(1-\eta)c_0 & \text{for } -c_0 \le h < -\eta c_0 \\ 0 & \text{for } h \ge -\eta c_0. \end{cases}$$

In static analysis the velocity is defined as the displacement increment divided by the time increment. Therefore, $\dot{\gamma}_j = \Delta\gamma_j/\Delta t$, and the stiffness contribution reduces to

$$\mathrm{d}\delta\Pi = \delta\gamma_i \frac{d\mu^v}{dh}\frac{\Delta\gamma_i}{\Delta t}\mathrm{d}h + \delta\gamma_i \frac{\mu^v}{\Delta t}\mathrm{d}\gamma_i.$$

The previous expression also applies to dynamics with the backward Euler time integration operator. In the case of dynamics with the Hilber-Hughes-Taylor time integration operator, $\partial\dot{\gamma}_j/\partial\gamma_k$ is defined by the dynamic time integration operator and the stiffness contribution can be written as

$$\mathrm{d}\delta\Pi = \delta\gamma_i \frac{d\mu}{dh}\dot{\gamma}_i\mathrm{d}h + \delta\gamma_i \frac{\gamma\mu}{\beta\Delta t}\mathrm{d}\gamma_i,$$

where γ and β are the Hilber-Hughes-Taylor time integration operator parameters. Viscous damping cannot be used in a Riks analysis since velocity is not defined.

Reference

- "Frictional behavior," Section 37.1.5 of the Abaqus Analysis User's Guide

5.2.4 THERMAL INTERFACE DEFINITION

Products: Abaqus/Standard Abaqus/Explicit

Abaqus/Standard allows heat flow across an interface via conduction or radiation. Generally, both modes of heat transfer are present to some degree. Regardless of the mode, heat transfer across the interface is assumed to occur only in the normal direction.

Conduction

Heat conduction across the interface is assumed to be defined by

$$q = k(\theta_A - \theta_B),$$

where q is the heat flux per unit area crossing the interface from point A on one surface to point B on the other, θ_A and θ_B are the temperatures of the points on the surfaces, and k is the gap conductance.

The derivatives of q are

$$\frac{\partial q}{\partial \theta_A} = k + \frac{1}{2}\frac{\partial k}{\partial \bar{\theta}}(\theta_A - \theta_B)$$

and

$$\frac{\partial q}{\partial \theta_B} = -k + \frac{1}{2}\frac{\partial k}{\partial \bar{\theta}}(\theta_A - \theta_B),$$

where

$$\bar{\theta} = \frac{1}{2}(\theta_A + \theta_B).$$

Radiation

The heat flow per unit area between corresponding points is assumed to be given by

$$q = F\left[(\theta_A - \theta^Z)^4 - (\theta_B - \theta^Z)^4\right],$$

where θ^Z is the value of absolute zero temperature on the temperature scale being used; q is the heat flux per unit surface area crossing the gap at this point, from surface A to surface B; θ_A and θ_B are the temperatures of the two surfaces; and F is the gap radiation constant derived from the emissivities of the two surfaces.

The derivatives of q are

$$\frac{\partial q}{\partial \theta_A} = 4F(\theta_A - \theta^Z)^3$$

and

$$\frac{\partial q}{\partial \theta_B} = -4F(\theta_B - \theta^Z)^3.$$

Jacobian matrix

The contribution to the variational statement of thermal equilibrium is

$$\delta\Pi = qA\delta\theta_A - qA\delta\theta_B,$$

where A is the area. The contribution to the Jacobian matrix for the Newton solution is

$$\mathrm{d}\delta\Pi = \mathrm{d}qA\delta\theta_A - \mathrm{d}qA\delta\theta_B,$$

where

$$\mathrm{d}q = \frac{\partial q}{\partial \theta_A}\mathrm{d}\theta_A + \frac{\partial q}{\partial \theta_B}\mathrm{d}\theta_B.$$

For "tied" thermal contact the temperature at point A is constrained to have the same temperature as point B. The Lagrange multiplier method is used to impose the constraint by augmenting the thermal equilibrium statement as follows:

$$\delta\Pi = \delta\lambda(\theta_A - \theta_B) + \lambda(\delta\theta_A - \delta\theta_B),$$

where λ is the Lagrange multiplier. The contribution to the Jacobian matrix for the Newton solution is

$$\mathrm{d}\delta\Pi = \delta\lambda(\mathrm{d}\theta_A - \mathrm{d}\theta_B) + \mathrm{d}\lambda(\delta\theta_A - \delta\theta_B).$$

Reference

- "Thermal contact properties," Section 37.2.1 of the Abaqus Analysis User's Guide

5.2.5 HEAT GENERATION CAUSED BY FRICTIONAL SLIDING

Products: Abaqus/Standard Abaqus/Explicit

For coupled temperature-displacement and coupled thermal-electrical-structural analyses in Abaqus/Standard the user can introduce a factor, η, which defines the fraction of frictional work converted to heat. The fraction of generated heat into the first and second surface, f_1 and f_2, respectively, can also be defined. This heat generation capability is to be used only in coupled temperature-displacement and coupled thermal-electrical-structural analyses.

The heat fraction, η, determines the fraction of the energy dissipated during frictional slip that enters the contacting bodies as heat. Heat is instantaneously conducted into each of the contacting bodies depending on the values of f_1 and f_2. The contact interface is assumed to have no heat capacity and may have properties for the exchange of heat by conduction and radiation.

Refer to "Small-sliding interaction between bodies," Section 5.1.1, and "Finite-sliding interaction between deformable bodies," Section 5.1.2, for explanations of the notation used for the shape functions and contact parameters involved in the small-sliding and slide line theory. Note that the shape functions for interpolation of the temperature field may be different from the interpolation functions for the displacements; for example, if the underlying elements are of second order, the displacements are interpolated using quadratic functions, whereas the temperature field is interpolated using linear shape functions. Hence, the temperature interpolator will be denoted with the symbol M and the displacement interpolation will be denoted with the symbol N. Only the heat transfer terms will be discussed in the following.

The heat flux densities—q_1, going out the surface on side 1, and q_2, going out the surface on side 2—are given by

$$q_1 = q_k + q_r - f_1 q_g$$

and

$$q_2 = -q_k - q_r - f_2 q_g,$$

where q_g is the heat flux density generated by the interface element due to frictional heat generation, q_k is the heat flux due to conduction, and q_r is the heat flux due to radiation.

The heat flux density generated by the interface element due to frictional heat generation is given by

$$q_g = \eta \tau \dot{s} = \eta \tau \frac{\Delta s}{\Delta t},$$

where τ is the frictional stress, Δs is the incremental slip, and Δt is the incremental time. The frictional stress is dependent on the contact pressure, p; the friction coefficient, μ; and the temperatures on either side of the interface.

The heat flux due to conduction is assumed to be of the form

$$q_k = \kappa(h, p, \overline{\theta})(\theta_1 - \theta_2) = \kappa(h, p, \overline{\theta})\Delta\theta,$$

where the heat transfer coefficient $\kappa(h, p, \overline{\theta})$ is a function of the average temperature at the contact point, $\overline{\theta} = \frac{1}{2}(\theta_1 + \theta_2)$; the overclosure, h; and the contact pressure, p. θ_1 and θ_2 are the temperatures of side 1 and side 2, respectively.

The heat flux due to radiation is assumed to be of the form

$$q_r = F\left[(\theta_1 - \theta^Z)^4 - (\theta_2 - \theta^Z)^4\right],$$

where F is the gap radiation constant (derived from the emissivities of the two surfaces) and θ^Z is the absolute zero on the temperature scale used.

Using the Galerkin method the weak form of the equations can be written as

$$\int_S \delta\theta_1 q_1 dS = \int_S \delta\theta_1 (q_k + q_r - f_1 q_g) dS, \qquad \int_S \delta\theta_2 q_2 dS = \int_S \delta\theta_2 (-q_k - q_r - f_2 q_g) dS.$$

The contribution to the variational statement of thermal equilibrium is

$$\delta\Pi = \int_S (\delta\theta_1 q_1 + \delta\theta_2 q_2) dS = \int_S [\delta\Delta\theta (q_k + q_r) - \delta\hat{\theta}\ q_g] dS,$$

where $\hat{\theta} = f_1\theta_1 + f_2\theta_2$. The contribution to the Jacobian matrix for the Newton solution is

$$d\delta\Pi = \int_S [d\delta\Delta\theta (q_k + q_r) + \delta\Delta\theta (dq_k + dq_r) - d\delta\hat{\theta}\ q_g - \delta\hat{\theta}\ dq_g] dS. \tag{5.2.5–1}$$

At a contact point the temperatures can be interpolated with

$$\theta_1(s) = M_1^N(s)\theta^N, \qquad \theta_2(s) = M_2^N(s)\theta^N,$$

where θ^N is the temperature at the Nth node associated with the interface element. Note that the summation convention will be used for all superscripts. Therefore, the temperature variables can be written as follows:

$$\Delta\theta(s) = \Delta M^N(s)\theta^N, \qquad \hat{\theta}(s) = \hat{M}^N(s)\theta^N, \qquad \overline{\theta}(s) = \overline{M}^N(s)\theta^N,$$

where $\hat{M}^N(s) = f_1 M_1^N(s) + f_2 M_2^N(s)$ and $\overline{M}^N(s) = \frac{1}{2}(M_1^N(s) + M_2^N(s))$. Substituting the above expressions into Equation 5.2.5–1, we obtain

$$\begin{aligned} d\delta\Pi = \delta\theta^N \int_S \Big[& (q_k + q_r)\frac{d\Delta M^N}{ds} ds + \Delta M^N \left(\frac{\partial q_r}{\partial \theta_1} d\theta_1 + \frac{\partial q_r}{\partial \theta_2} d\theta_2 \right. \\ & \left. + \frac{\partial q_k}{\partial \Delta\theta} d\Delta\theta + \frac{\partial q_k}{\partial \overline{\theta}} d\overline{\theta} + \frac{\partial q_k}{\partial h} dh + \frac{\partial q_k}{\partial p} dp \right) \\ & - q_g \frac{d\hat{M}^N}{ds} ds - \hat{M}^N \left(\frac{\partial q_g}{\partial \tau} d\tau + \frac{\partial q_g}{\partial s} ds \right) \Big] dS. \end{aligned}$$

After rearranging and expanding terms, we obtain

$$
\begin{aligned}
d\delta\Pi = \delta\theta^N \int_S \Big[& \Big((q_k + q_r)\frac{d\Delta M^N}{ds} - q_g \frac{d\hat{M}^N}{ds} \Big) ds \\
& + \Delta M^N \left(\frac{\partial q_r}{\partial \theta_1} M_1^M + \frac{\partial q_r}{\partial \theta_2} M_2^M + \frac{\partial q_k}{\partial \Delta\theta} \Delta M^M + \frac{\partial q_k}{\partial \overline{\theta}} \overline{M}^M \right) d\theta^M \\
& + \Delta M^N \left(\frac{\partial q_k}{\partial h} dh + \frac{\partial q_k}{\partial p} dp \right) \\
& + \Delta M^N \left(\frac{\partial q_r}{\partial \theta_1} \frac{dM_1^K}{ds} \theta^K + \frac{\partial q_r}{\partial \theta_2} \frac{dM_2^K}{ds} \theta^K + \frac{\partial q_k}{\partial \Delta\theta} \frac{d\Delta M^K}{ds} \theta^K + \frac{\partial q_k}{\partial \overline{\theta}} \frac{d\overline{M}^K}{ds} \theta^K \right) ds \\
& - \hat{M}^N \left(\frac{\partial q_g}{\partial \tau} \left(\frac{\partial \tau}{\partial p} dp + \frac{\partial \tau}{\partial \theta_1} d\theta_1 + \frac{\partial \tau}{\partial \theta_2} d\theta_2 \right) + \frac{\partial q_g}{\partial s} ds \right) \Big] dS.
\end{aligned}
$$

Expanding the terms involving frictional heat generation yields

$$
\begin{aligned}
d\delta\Pi = \delta\theta^N \int_S \Big[& \Big((q_k + q_r)\frac{d\Delta M^N}{ds} - q_g \frac{d\hat{M}^N}{ds} \Big) ds \\
& + \Delta M^N \left(\frac{\partial q_r}{\partial \theta_1} M_1^M + \frac{\partial q_r}{\partial \theta_2} M_2^M + \frac{\partial q_k}{\partial \Delta\theta} \Delta M^M + \frac{\partial q_k}{\partial \overline{\theta}} \overline{M}^M \right) d\theta^M \\
& + \Delta M^N \left(\frac{\partial q_k}{\partial h} dh + \frac{\partial q_k}{\partial p} dp \right) \\
& + \Delta M^N \left(\frac{\partial q_r}{\partial \theta_1} \frac{dM_1^K}{ds} \theta^K + \frac{\partial q_r}{\partial \theta_2} \frac{dM_2^K}{ds} \theta^K + \frac{\partial q_k}{\partial \Delta\theta} \frac{d\Delta M^K}{ds} \theta^K + \frac{\partial q_k}{\partial \overline{\theta}} \frac{d\overline{M}^K}{ds} \theta^K \right) ds \\
& - \hat{M}^N \frac{\partial q_g}{\partial \tau} \left(\frac{\partial \tau}{\partial \theta_1} M_1^M + \frac{\partial \tau}{\partial \theta_2} M_2^M \right) d\theta^M \\
& - \hat{M}^N \frac{\partial q_g}{\partial \tau} \left(\frac{\partial \tau}{\partial \theta_1} \frac{dM_1^K}{ds} \theta^K + \frac{\partial \tau}{\partial \theta_2} \frac{dM_2^K}{ds} \theta^K \right) ds \\
& - \hat{M}^N \left(\frac{\partial q_g}{\partial \tau} \frac{\partial \tau}{\partial p} dp + \frac{\partial q_g}{\partial s} ds \right) \Big] dS.
\end{aligned}
$$

The derivatives of q_r, q_k, and q_g, are as follows:

$$
\frac{\partial q_r}{\partial \theta_1} = 4F(\theta_1 - \theta^Z)^3, \qquad \frac{\partial q_r}{\partial \theta_2} = -4F(\theta_2 - \theta^Z)^3,
$$

$$
\frac{\partial q_k}{\partial \Delta\theta} = \kappa(h, p, \overline{\theta}), \qquad \frac{\partial q_k}{\partial \overline{\theta}} = \frac{\partial \kappa(h, p, \overline{\theta})}{\partial \overline{\theta}} \Delta\theta,
$$

$$
\frac{\partial q_k}{\partial h} = \frac{\partial \kappa(h, p, \overline{\theta})}{\partial h} \Delta\theta, \qquad \frac{\partial q_k}{\partial p} = \frac{\partial \kappa(h, p, \overline{\theta})}{\partial p} \Delta\theta,
$$

$$\frac{\partial q_g}{\partial \tau}\frac{\partial \tau}{\partial p} = \eta \frac{|\Delta s|}{\Delta t}\mu, \qquad \frac{\partial q_g}{\partial s} = \eta \frac{\tau}{\Delta t}.$$

For contact pairs and slide line elements, each integration point is associated with a unique slave node. If we associate M_1^N with the slave surface, then M_1^N will again have only a single nonzero entry equal to one and the derivatives of M_1^N with respect to s vanish. In contrast, on the master surface there will be multiple nonzero entries in M_2^N, which are a function of the position on the master surface at which contact occurs.

For GAPUNIT and DGAP elements each contact (integration point) is directly associated with a node pair. Hence, M_1^N and M_2^N each have one nonzero entry that is equal to one, and all terms involving derivatives of M_1^N and M_2^N with respect to s vanish.

The variations of overclosure, h, and slip, s, can be written as linear functions of the variations of displacement. These expressions, which determine the form of the β matrix for contact elements, have been derived in "Small-sliding interaction between bodies," Section 5.1.1, and "Finite-sliding interaction between deformable bodies," Section 5.1.2.

References

- "Fully coupled thermal-stress analysis," Section 6.5.3 of the Abaqus Analysis User's Guide
- "Fully coupled thermal-electrical-structural analysis," Section 6.7.4 of the Abaqus Analysis User's Guide
- "Thermal contact properties," Section 37.2.1 of the Abaqus Analysis User's Guide

5.2.6 HEAT GENERATION CAUSED BY ELECTRICAL CURRENT

Product: Abaqus/Standard

For coupled thermal-electrical and coupled thermal-electrical-structural analyses in Abaqus/Standard the user can introduce a factor, η, which defines the fraction of electrical dissipation in the contact zone converted to heat. The fraction of generated heat into the first and second surface, f_1 and f_2, respectively, can also be defined.

The heat fraction, η, determines the fraction of the energy dissipated due to electrical current that enters the contacting bodies as heat. Heat is instantaneously conducted into each of the contacting bodies depending on the values of f_1 and f_2. The contact interface is assumed to have no heat capacity and may have properties for the exchange of heat by conduction and radiation.

The heat flux densities, q_1, going out the surface on side 1, and q_2, going out the surface on side 2, are given by

$$q_1 = q_k + q_r - f_1 q_g$$

and

$$q_2 = -q_k - q_r - f_2 q_g,$$

where q_g is the heat flux density generated by the interface element due to electrical current, q_k is the heat flux due to conduction, and q_r is the heat flux due to radiation.

The heat flux due to conduction is assumed to be of the form

$$q_k = \kappa(h, \overline{\theta})(\theta_1 - \theta_2) = \kappa(h, \overline{\theta})\Delta\theta,$$

where the heat transfer coefficient $\kappa(h, \overline{\theta})$ is a function of the average temperature at the contact point, $\overline{\theta} = \frac{1}{2}(\theta_1 + \theta_2)$, and overclosure, h. θ_1 and θ_2 are the temperatures of side 1 and side 2, respectively.

The heat flux due to radiation is assumed to be of the form

$$q_r = F\left[(\theta_1 - \theta^Z)^4 - (\theta_2 - \theta^Z)^4\right],$$

where F is the gap radiation constant (derived from the emissivities of the two surfaces) and θ^Z is the absolute zero on the temperature scale used.

The electrical flux density, J, in the interface element is given in terms of the difference in the electric potential, $\Delta\varphi$, across the interface:

$$J = \sigma_g(h, \overline{\theta})(\varphi_1 - \varphi_2) = \sigma_g(h, \overline{\theta})\Delta\varphi,$$

where the gap electrical conductance $\sigma_g(h, \overline{\theta})$ is a function of the overclosure, h, and the average temperature at the contact point, $\overline{\theta}$. φ_1 and φ_2 are the electric potentials of side 1 and side 2, respectively.

In a steady-state analysis the heat flux density generated by the interface element due to electrical current is given by

$$q_g = \eta\sigma_g(h,\overline{\theta})(\varphi_1 - \varphi_2)^2 = \eta\sigma_g(h,\overline{\theta})(\Delta\varphi)^2,$$

where η is the fraction of dissipated energy converted to heat. In a transient analysis the average heat flux density is given by

$$\begin{aligned} q_g &= \frac{\eta}{\Delta t}\int_t^{t+\Delta t} \sigma_g(h,\overline{\theta})(\Delta\varphi)^2 dt \\ &\approx \frac{1}{3}\eta\sigma_g(h,\overline{\theta})\left[(\Delta\varphi_{t+\Delta t})^2 + \Delta\varphi_{t+\Delta t}\Delta\varphi_t + (\Delta\varphi_t)^2\right], \end{aligned}$$

where t is the time at the start of an increment and Δt is the time increment.

Using the Galerkin method, the weak form of the equations can be written as

$$\int_S \delta\theta_1 q_1 dS = \int_S \delta\theta_1(q_k + q_r - f_1 q_g)dS, \qquad \int_S \delta\theta_2 q_2 dS = \int_S \delta\theta_2(-q_k - q_r - f_2 q_g)dS.$$

The contribution to the variational statement of thermal equilibrium is

$$\delta\Pi = \int_S (\delta\theta_1 q_1 + \delta\theta_2 q_2)dS = \int_S [\delta\Delta\theta(q_k + q_r) - \delta\hat{\theta}\, q_g]dS,$$

where $\hat{\theta} = f_1\theta_1 + f_2\theta_2$. The contribution to the Jacobian matrix for the Newton solution is

$$d\delta\Pi = \int_S [\delta\Delta\theta(dq_k + dq_r) - \delta\hat{\theta}\, dq_g]dS. \tag{5.2.6–1}$$

At a contact point the temperatures can be interpolated with

$$\theta_1(s) = M_1^N(s)\theta^N \qquad \text{and} \qquad \theta_2(s) = M_2^N(s)\theta^N,$$

where θ^N is the temperature at the Nth node associated with the interface element. Note that the summation convention will be used for all superscripts. Therefore, the temperature variables can be written as follows:

$$\Delta\theta(s) = \Delta M^N(s)\theta^N, \qquad \hat{\theta}(s) = \hat{M}^N(s)\theta^N, \qquad \overline{\theta}(s) = \overline{M}^N(s)\theta^N,$$

where $\hat{M}^N(s) = f_1 M_1^N(s) + f_2 M_2^N(s)$ and $\overline{M}^N(s) = \frac{1}{2}(M_1^N(s) + M_2^N(s))$. Substituting the above expressions to Equation 5.2.6–1, we obtain

$$\begin{aligned} d\delta\Pi = \delta\theta^N \int_S \Big[&\Delta M^N \left(\frac{\partial q_r}{\partial\theta_1}d\theta_1 + \frac{\partial q_r}{\partial\theta_2}d\theta_2 + \frac{\partial q_k}{\partial\Delta\theta}d\Delta\theta + \frac{\partial q_k}{\partial\overline{\theta}}d\overline{\theta}\right) \\ &- \hat{M}^N\left(\frac{\partial q_g}{\partial\Delta\varphi}d\Delta\varphi + \frac{\partial q_g}{\partial\overline{\theta}}d\overline{\theta}\right)\Big]dS. \end{aligned}$$

The derivatives of q_r, q_k, and q_g, are as follows:

$$\frac{\partial q_r}{\partial \theta_1} = 4F(\theta_1 - \theta^Z)^3, \qquad \frac{\partial q_r}{\partial \theta_2} = -4F(\theta_2 - \theta^Z)^3,$$

$$\frac{\partial q_k}{\partial \Delta\theta} = \kappa(h, \overline{\theta}), \qquad \frac{\partial q_k}{\partial \overline{\theta}} = \frac{\partial \kappa(h, \overline{\theta})}{\partial \overline{\theta}} \Delta\theta,$$

and in a steady-state analysis

$$\frac{\partial q_g}{\partial \Delta\varphi} = 2\eta\sigma_g(h, \overline{\theta})\Delta\varphi, \qquad \frac{\partial q_g}{\partial \overline{\theta}} = \eta \frac{\partial \sigma_g(h, \overline{\theta})}{\partial \overline{\theta}} (\Delta\varphi)^2,$$

while in a transient analysis

$$\frac{\partial q_g}{\partial \Delta\varphi} = \frac{1}{3}\eta\sigma_g(h, \overline{\theta})(2\Delta\varphi + \Delta\varphi_t), \qquad \frac{\partial q_g}{\partial \overline{\theta}} = \frac{1}{3}\eta \frac{\partial \sigma_g(h, \overline{\theta})}{\partial \overline{\theta}} \left[(\Delta\varphi)^2 + \Delta\varphi\Delta\varphi_t + (\Delta\varphi_t)^2\right].$$

Similarly, the contribution to the variational statement of electrical equilibrium is

$$\delta\Pi = \int_S (\delta\varphi_1 J - \delta\varphi_2 J) dS = \int_S \delta\Delta\varphi J dS,$$

and the contribution to the Jacobian matrix for the Newton solution is

$$d\delta\Pi = \int_S \delta\Delta\varphi \left(\frac{\partial J}{\partial \overline{\theta}} d\overline{\theta} + \frac{\partial J}{\partial \Delta\varphi} d\Delta\varphi \right) dS.$$

The derivatives of J are

$$\frac{\partial J}{\partial \overline{\theta}} = \frac{\partial \sigma_g(h, \overline{\theta})}{\partial \overline{\theta}} \Delta\varphi \qquad \text{and} \qquad \frac{\partial J}{\partial \Delta\varphi} = \sigma_g(h, \overline{\theta}).$$

References

- "Coupled thermal-electrical analysis," Section 6.7.3 of the Abaqus Analysis User's Guide
- "Fully coupled thermal-electrical-structural analysis," Section 6.7.4 of the Abaqus Analysis User's Guide
- "Electrical contact properties," Section 37.3.1 of the Abaqus Analysis User's Guide

5.2.7 SURFACE-BASED ACOUSTIC-STRUCTURAL MEDIUM INTERACTION

Products: Abaqus/Standard Abaqus/Explicit

Abaqus provides two alternatives for modeling interaction between acoustic and structural media: surface-based interaction or acoustic interface (ASI*n*) elements. Both are available in Abaqus/Standard, but only the surface-based capability is available in Abaqus/Explicit. If the special-purpose interface elements (ASI*n*) are used, interacting structural and acoustic nodes must be shared by the two meshes. The surface-based capability can be used for structural and acoustic meshes that have different node numbering and whose surface meshes may not be spatially coincident. The ease of use and low computational cost of the surface-based procedure make it preferable to the element-based approach.

The equations on the contact interface between the structure and the acoustic medium

In the surface-based method the tractions and volumetric acceleration fluxes are computed between structural and acoustic media. In place of consistent distributed tractions or fluxes on both media, one side (identified as the "slave") receives point tractions/fluxes based on interpolation with the shape functions from the other ("master") side. Either the acoustic fluid or the structural solid can be the slave or master, and no Lagrange multipliers are introduced in the formulation. The basis for deciding which to make slave or master is discussed in the Abaqus Analysis User's Guide.

The transient expressions for the coupled acoustic-structural problem are

$$\int_{V_f}\left[\delta p\left(\frac{1}{\mathrm{K}_f}\,\ddot{p}+\frac{r}{\rho_f \mathrm{K}_f}\,\dot{p}\right)+\frac{1}{\rho_f}\frac{\partial \delta p}{\partial \mathbf{x}}\cdot\frac{\partial p}{\partial \mathbf{x}}\right]dV+\int_{S_{\mathrm{fs}}\cup S_{\mathrm{frs}}}\delta p\,\mathbf{n}^{-}\cdot\frac{\partial p}{\partial \mathbf{x}}\,dS-\int_{S_{\mathrm{ft}}}\delta p\,a_{in}\,dS$$
$$+\int_{S_{\mathrm{fr}}\cup S_{\mathrm{frs}}}\delta p\left(\frac{r}{\rho_f}\frac{1}{c_1}\,p+\left(\frac{r}{\rho_f}\frac{1}{k_1}+\frac{1}{c_1}\right)\dot{p}+\frac{1}{k_1}\,\ddot{p}\right)dS=0$$

(acoustic medium) and

$$\int_V \delta\boldsymbol{\varepsilon}:\boldsymbol{\sigma}\,dV+\int_V \alpha_c\rho\delta\mathbf{u}^m\cdot\dot{\mathbf{u}}^m\,dV+\int_V \rho\delta\mathbf{u}^m\cdot\ddot{\mathbf{u}}^m\,dV$$
$$+\int_{S_{\mathrm{fs}}\cup S_{\mathrm{frs}}}\delta\mathbf{u}^m\cdot\mathbf{n}^{-}\,p\,dS-\int_{S_{\mathrm{t}}}\delta\mathbf{u}^m\cdot\mathbf{t}\,dS=0$$

(structural medium), where $\mathbf{n}^{-}$ is the normal vector pointing into the fluid. The fluid-solid surface consists of the union of the directly coupled fluid-solid region, S_{fs}, and a region coupled via a "reactive" acoustic surface or impedance boundary, S_{frs}. Of primary interest here are those terms integrated over $S_{\mathrm{fs}}\cup S_{\mathrm{frs}}$, which couple the two variational equations. The fluid impedance integral, over $S_{\mathrm{fr}}\cup S_{\mathrm{frs}}$, depends only on the acoustic pressure field and its variations, so it is unaffected by the contact with the solid. The derivation for the steady-state case is formally identical to the transient case and will not be discussed here. For details of the differences in transient and steady-state acoustics in Abaqus, see "Coupled acoustic-structural medium analysis," Section 2.9.1.

When ASI*n* elements are used (see "Acoustic interface elements," Section 32.13.1 of the Abaqus Analysis User's Guide), the formulation requires that the fluid and solid elements be geometrically and nodally conformal so that the shape functions for the structural displacements and the acoustic pressures are identical. The shape functions are integrated using standard methods to yield element matrices of dimension equal to the number of surface nodes on the element. The complete fluid-solid coupling matrices are formed by the sum over the element faces; that is, a standard element assembly operation. The two final coupling matrices have the sparsity pattern of the coupled fluid-solid element faces.

In surface-based coupling the interaction surface $S_{\text{fs}} \cup S_{\text{frs}}$ is formed by the boundary between possibly nonconforming structural and acoustic meshes. Therefore, the fluid-solid coupling matrix cannot be broken up into a sum over element faces as simply as in the ASI*n* case. To derive the coupling matrices in the surface-based procedure, we use a variation of the master-slave procedure used in small-sliding contact (see "Small-sliding interaction between bodies," Section 5.1.1). At the start of an analysis, the projections $\mathbf{x}_{\text{N}}$ of slave nodes onto the master surface are found, and the areas A_N and normals $\mathbf{n}^{-}(\mathbf{x}_{\text{N}})$ associated with the slave nodes are computed. The projections are points $\mathbf{p}(\mathbf{x}_{\text{N}})$ on the master surface; master nodes in the vicinity of this projection are identified. Variables at the slave nodes $\mathbf{x}_{\text{N}}$ are then interpolated from variables at the identified master surface nodes near the projection $\mathbf{p}(\mathbf{x}_{\text{N}})$.

Since the physical degrees of freedom for the fluid and solid meshes are different, two cases must be treated. The two cases handle the discretization of the coupling terms differently.

Solid/structural master, fluid slave

If the fluid medium surface is designated as the slave, we constrain values at each fluid node to be an average of the values at nearby master surface nodes (see Figure 5.2.7–1). The pointwise fluid-solid coupling condition,

$$\frac{1}{\rho_f}\frac{\partial p}{\partial \mathbf{x}} \cdot \mathbf{n}^{-} + \ddot{\mathbf{u}}^{m} \cdot \mathbf{n}^{-} = 0,$$

is enforced at the slave nodes, resulting in displacement degrees of freedom added to the fluid slave surface. These slave displacements are constrained by the master displacements and thereby eliminated; the slave pressures are not constrained directly. Hence, the fluid equation coupling term

$$\int_{S_{\text{fs}} \cup S_{\text{frs}}} \delta p \, \mathbf{n}^{-} \cdot \frac{\partial p}{\partial \mathbf{x}} \, dS$$

is equal to

$$-\int_{S_{\text{fs}} \cup S_{\text{frs}}} \delta p \, \mathbf{n}^{-} \cdot \ddot{\mathbf{u}}^{m} \, dS.$$

This term is now approximated, at the slave node level, by the interpolated values of structural displacements at the nearby master nodes times the area of the slave node:

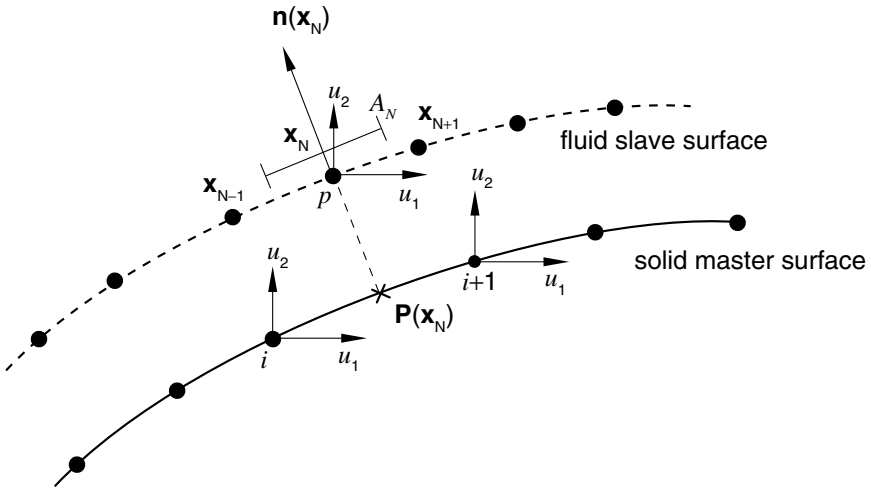


Figure 5.2.7–1 Fluid slave.

$$\int_{S_{\mathrm{fs}} \cup S_{\mathrm{frs}}} \delta p \, \mathbf{n}^{-} \cdot \ddot{\mathbf{u}}^{m} \, dS \approx A_N \left[\sum_i \mathbf{n}^{-} \left(\mathbf{x}_{\mathrm{N}} \right) \cdot \mathrm{N}^{i} \left(\mathbf{p} \left(\mathbf{x}_{\mathrm{N}} \right) \right) \ddot{\mathbf{u}}_i^{m} \right],$$

where $\mathrm{N}^{i} \left(\mathbf{p} \left(\mathbf{x}_{\mathrm{N}} \right) \right)$ is the master surface interpolant evaluated at the projection of the slave node, $\ddot{\mathbf{u}}_i^{m}$ are the structural accelerations at the master nodes, and $\mathbf{n}^{-} \left(\mathbf{x}_{\mathrm{N}} \right)$ is the normal vector, pointing into the fluid, evaluated at the slave node. The summation extends over all master nodes i in the vicinity of the slave node projection $\mathbf{p} \left(\mathbf{x}_{\mathrm{N}} \right)$ (see Figure 5.2.7–1). The computation is repeated for each slave node $\mathbf{x}_{\mathrm{N}}$ on the surface $S_{\mathrm{fs}} \cup S_{\mathrm{frs}}$ and assembled to form the entire coupling matrix.

Similarly, the contribution to the pressure coupling term in the structural equation due to slave node $\mathbf{x}_{\mathrm{N}}$ is approximated by

$$\int_{S_{\mathrm{fs}} \cup S_{\mathrm{frs}}} \delta \mathbf{u}^{m} \cdot \mathbf{n}^{-} \, p \, dS \approx p_N A_N \left[\sum_i \mathbf{n}^{-} \left(\mathbf{x}_{\mathrm{N}} \right) \cdot \mathrm{N}^{i} \left(\mathbf{p} \left(\mathbf{x}_{\mathrm{N}} \right) \right) \right],$$

where p_N is the acoustic pressure at slave node $\mathbf{x}_{\mathrm{N}}$, and the sum is again over the master nodes i in the vicinity of the slave node projection.

These expressions for the coupling terms result in matrices that are the transpose of each other. The normal vectors at the slave surface are used, so these vectors must be well-defined (see "Surfaces: overview," Section 2.3.1 of the Abaqus Analysis User's Guide).

Fluid master, solid/structural slave

If the solid medium is designated as the slave, the values on this surface are constrained to equal values interpolated from the master surface. The pointwise fluid-solid coupling condition is again enforced

at the slave nodes, resulting in acoustic pressure degrees of freedom added to the solid slave surface (see Figure 5.2.7–2). These slave pressures are constrained by the master surface acoustic pressures and eliminated; the slave displacements are not constrained directly. Hence, the contribution to the coupling term in the acoustic equation for a single slave node $\mathbf{x}_{\mathrm{N}}$ is approximated by

$$\int_{S_{\mathrm{fs}} \cup S_{\mathrm{frs}}} \delta p \, \mathbf{n}^{-} \cdot \ddot{\mathbf{u}}^{m} \, dS \approx \left[A_N \mathbf{n}^{-} \left(\mathbf{x}_{\mathrm{N}} \right) \cdot \ddot{\mathbf{u}}_N^{m} \right] \left[\sum_i \mathrm{H}^i \left(\mathbf{p} \left(\mathbf{x}_{\mathrm{N}} \right) \right) \right],$$

where $\ddot{\mathbf{u}}_N^m$ is the structural acceleration at the slave node and $\mathrm{H}^i \left(\mathbf{p} \left(\mathbf{x}_{\mathrm{N}} \right) \right)$ are the interpolants on the fluid (master) surface evaluated at the projection $\mathbf{p} \left(\mathbf{x}_{\mathrm{N}} \right)$. The summation extends over the master nodes i in the vicinity of the slave node projection and is repeated for all solid/structural nodes in the surface $S_{\mathrm{fs}} \cup S_{\mathrm{frs}}$ to compute the entire coupling matrix.

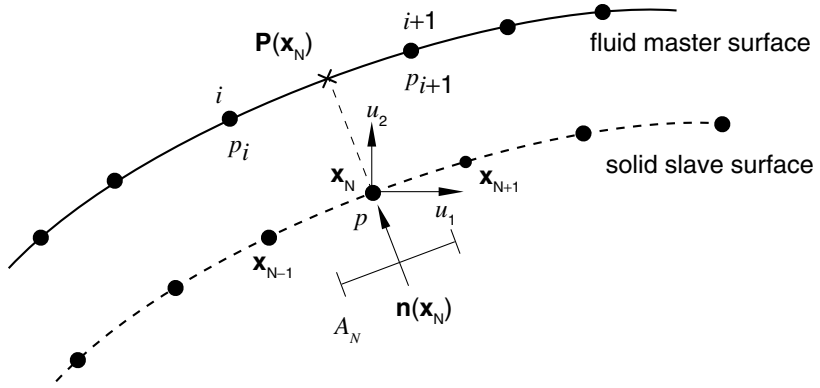


Figure 5.2.7–2 Fluid master.

The contribution of a slave node to the coupling term in the structural equation is approximated by

$$\int_{S_{\mathrm{fs}} \cup S_{\mathrm{frs}}} \delta \mathbf{u}^{m} \cdot \mathbf{n}^{-} \, p \, dS \approx A_N \sum_i \mathrm{H}^i \left(\mathbf{p} \left(\mathbf{x}_{\mathrm{N}} \right) \right) p_i,$$

where p_i is the pressure at master node i and the sum is again over the master nodes in the vicinity of the slave node projection $\mathbf{p} \left(\mathbf{x}_{\mathrm{N}} \right)$.

Reference

- "Acoustic, shock, and coupled acoustic-structural analysis," Section 6.10.1 of the Abaqus Analysis User's Guide

6. Loading and Constraints

Dynamic loading 6.1
Abaqus/Aqua loading 6.2
Incident wave loading 6.3
Pressure penetration loading 6.4
Load stiffness 6.5
Multi-point constraints 6.6

6.1 Dynamic loading

- “Centrifugal, Coriolis, and rotary acceleration forces,” Section 6.1.1
- “Baseline correction of accelerograms,” Section 6.1.2

6.1.1 CENTRIFUGAL, CORIOLIS, AND ROTARY ACCELERATION FORCES

Product: Abaqus/Standard

Many of the elements in Abaqus allow centrifugal, Coriolis, and rotary acceleration forces to be included. This section defines these load types.

It is assumed that the model (or that part of it to which these forces are applied) is described in a coordinate system that is rotating with an angular velocity, $\boldsymbol{\omega}$, and/or an angular (rotary) acceleration, $d\boldsymbol{\omega}/dt$. Let $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]$ be a right-handed set of unit, orthogonal vectors that form a basis for this system. Then, $d\mathbf{A}/dt = \boldsymbol{\omega} \times \mathbf{A}$ and $d^2\mathbf{A}/dt^2 = d\boldsymbol{\omega}/dt \times \mathbf{A} + \boldsymbol{\omega} \times d\mathbf{A}/dt$.

If the angular velocity is cast as

$$\boldsymbol{\omega} = \omega \mathbf{n},$$

where ω is the magnitude of $\boldsymbol{\omega}$ and $\mathbf{n}$ is the unit axis of rotation, the rotary acceleration is

$$\frac{d\boldsymbol{\omega}}{dt} = \frac{d\omega}{dt}\mathbf{n} + \omega\frac{d\mathbf{n}}{dt} = \alpha\, \mathbf{n}^{ROTA},$$

where the term $\omega d\mathbf{n}/dt$ represents the effect of the motion of the axis of rotation (precessional motion);

$\alpha = d\boldsymbol{\omega}/dt \cdot \mathbf{n}^{ROTA}$ is the magnitude of the rotary acceleration; and $\mathbf{n}^{ROTA}$ is the axis of rotary acceleration. If $d\mathbf{n}/dt = \mathbf{0}$, then $\alpha = d\omega/dt$ and $\mathbf{n}^{ROTA} = \mathbf{n}$. In component form

$$\omega^k = \omega n^k$$

and

$$\frac{d\omega^k}{dt} = \frac{d\omega}{dt}n^k + \omega\frac{dn^k}{dt}.$$

Let $\mathbf{x}_0$ be a point on the axis of rotation. The position of a material particle, $\mathbf{x}$, can be written

$$\mathbf{x} = \mathbf{x}_0 + y^i \mathbf{a}_i,$$

where y^i, $i = 1, 2, 3$, are the coordinates of the point in the rotating basis system. Taking time derivatives,

$$\frac{d\mathbf{x}}{dt} = \frac{d\mathbf{x}_0}{dt} + \frac{dy^i}{dt}\mathbf{a}_i + y^i\, \boldsymbol{\omega} \times \mathbf{a}_i$$

and

$$\frac{d^2\mathbf{x}}{dt^2} = \frac{d^2\mathbf{x}_0}{dt^2} + \frac{d^2y^i}{dt^2}\mathbf{a}_i + 2\frac{dy^i}{dt}\,\boldsymbol{\omega} \times \mathbf{a}_i + y^i\,\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{a}_i) + y^i\frac{d\boldsymbol{\omega}}{dt} \times \mathbf{a}_i.$$

We assume that the origin of the rotating system, x_0, is fixed, so that

$$\frac{d\mathbf{x}_0}{dt} = \frac{d^2\mathbf{x}_0}{dt^2} = 0.$$

With this simplification

$$\frac{d\mathbf{x}}{dt} = \frac{dy^i}{dt}\,\mathbf{a}_i + y^i\,\boldsymbol{\omega} \times \mathbf{a}_i$$

and

$$\frac{d^2\mathbf{x}}{dt^2} = \frac{d^2y^i}{dt^2}\mathbf{a}_i + 2\frac{dy^i}{dt}\,\boldsymbol{\omega} \times \mathbf{a}_i + y^i\,\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{a}_i) + y^i\frac{d\boldsymbol{\omega}}{dt} \times \mathbf{a}_i. \tag{6.1.1–1}$$

The virtual work contribution from the d'Alembert forces is

$$\delta W^A = -\int_{V^0} \rho^0 \frac{d^2\mathbf{x}}{dt^2} \cdot \delta\mathbf{x}\, dV^0,$$

where ρ^0 is the mass density of the body in its reference configuration, where its volume is V^0 and $\delta\mathbf{x}$ is a virtual variational field. For the part of the body described in the rotating system, the acceleration, $d^2\mathbf{x}/dt^2$, is given by Equation 6.1.1–1, while $\delta\mathbf{x} = \delta y^i\mathbf{a}_i$ only, since $\boldsymbol{\omega}$ and $d\boldsymbol{\omega}/dt$ are prescribed and $\mathbf{x}_0$ is fixed. Thus,

$$\delta W^A = -\int_{V^0} \rho^0 \left[\frac{d^2y^i}{dt^2}\delta y^i + \left\{2\frac{dy^i}{dt}\,\boldsymbol{\omega} \times \mathbf{a}_i + y^i\,\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{a}_i) + y^i\frac{d\boldsymbol{\omega}}{dt} \times \mathbf{a}_i\right\} \cdot \delta y^j\mathbf{a}_j\right] dV^0.$$

Simplifying,

$$\delta W^A = -\int_{V^0} \rho^0 \Bigg[\frac{d^2y^i}{dt^2}\delta y^i + 2\frac{dy^i}{dt}\delta y^j\,\boldsymbol{\omega} \cdot (\mathbf{a}_i \times \mathbf{a}_j) + y^i\delta y^j\,(\boldsymbol{\omega} \cdot \mathbf{a}_i\,\boldsymbol{\omega} \cdot \mathbf{a}_j - (\boldsymbol{\omega} \cdot \boldsymbol{\omega})\delta_{ij}) + y^i\delta y^j\frac{d\boldsymbol{\omega}}{dt} \times \mathbf{a}_i \cdot \mathbf{a}_j\Bigg] dV^0.$$

The terms in δW^A can be identified as follows. The first term,

$$-\int_{V^0} \rho^0 \frac{d^2y^i}{dt^2}\delta y^i\, dV^0,$$

is the usual "consistent mass matrix" term associated with acceleration of the material particles with respect to the rotating system.

Writing the angular velocity of the rotating basis system as its components in that system, $\boldsymbol{\omega} = \omega^i\mathbf{a}_i$, gives the second term as

$$-\int_{V^0} 2\rho^0\epsilon_{ijk}\frac{dy^i}{dt}\delta y^j\omega^k\, dV^0,$$

where ϵ_{ijk} is the alternator tensor. This term is the Coriolis force term and arises whenever there is velocity in the rotating system, which can happen in dynamic analysis or in quasi-static cases in which a constant velocity has been introduced (for example, by defining an initial velocity).

The third term is, likewise, rewritten as

$$-\int_{V^0} \rho^0 y^i \delta y^j (\omega^i \omega^j - \omega^k \omega^k \delta_{ij}) dV^0.$$

This term is the centrifugal load term.

Similarly, the fourth term is rewritten as

$$-\int_{V^0} \rho^0 \epsilon_{ijk} y^i \delta y^j \frac{d\omega^k}{dt} dV^0.$$

This term is the rotary acceleration load term.

In Abaqus/Standard the centrifugal load, Coriolis, and rotary acceleration terms contribute to the "load stiffness matrix." The centrifugal load term has a symmetric load stiffness matrix,

$$-\int_{V^0} \rho^0 dy^i \delta y^j (\omega^i \omega^j - \omega^k \omega^k \delta_{ij}) dV^0;$$

the Coriolis term has an antisymmetric "load damping matrix,"

$$\int_{V^0} 2\rho^0 \epsilon_{ijk} \frac{\partial}{\partial y^l} \left(\frac{dy^i}{dt} \right) dy^l \delta y^j \omega^k \, dV^0;$$

and the rotary acceleration term has an antisymmetric load stiffness matrix,

$$-\int_{V^0} \rho^0 \epsilon_{ijk} dy^i \delta y^j \frac{d\omega^k}{dt} dV^0.$$

Reference

- "Distributed loads," Section 34.4.3 of the Abaqus Analysis User's Guide

6.1.2 BASELINE CORRECTION OF ACCELEROGRAMS

Product: Abaqus/Standard

Abaqus/Standard offers baseline correction of acceleration records for time domain analysis. The correction technique is that proposed by Newmark (1973). An acceleration correction, $a_0(t)$, is added to the raw data record, $a(t)$, to produce a corrected acceleration record, $a_c(t) = a + a_0$, such that the mean square velocity over the time of the event is minimized. The acceleration correction is parabolic over any number of time intervals during the event:

$$a_0(t) = C_1 + C_2\left(\frac{t - T_1}{T_2}\right) + C_3\left(\frac{t - T_1}{T_2}\right)^2, \qquad T_1 < t < T_2,$$

where T_1, T_2 denote the limits of a time interval and C_k, $k = 1, 2, 3$, are constants obtained from the velocity minimization:

$$\frac{\partial}{\partial C_k}\int_{T_1}^{T_2} [v_c(t)]^2\, dt = 0,$$

where $v_c(t)$ is the corrected velocity record obtained by integrating the corrected acceleration record, $a_c(t)$.

It can be shown that the C_k are defined for each time interval by

$$\begin{Bmatrix} C_1 \\ C_2 \\ C_3 \end{Bmatrix} = \begin{bmatrix} -300. & 900. & -630. \\ 1800./\xi & -5760./\xi & 4200./\xi \\ -1890./\xi^2 & 6300./\xi^2 & -4725./\xi^2 \end{bmatrix} \begin{Bmatrix} A_1 + \big(v_c(T_1) - v(T_1)\big)/(2\Delta T) \\ A_2 + \big(v_c(T_1) - v(T_1)\big)/(3\Delta T) \\ A_3 + \big(v_c(T_1) - v(T_1)\big)/(4\Delta T) \end{Bmatrix},$$

where $\xi = \Delta T/T_2$; $\Delta T = T_2 - T_1$; and A_1, A_2, and A_3 are defined as

$$A_1 = \frac{1}{(\Delta T)^3}\int_0^{\Delta T} v(\tau + T_1)\tau\, d\tau,$$
$$A_2 = \frac{1}{(\Delta T)^4}\int_0^{\Delta T} v(\tau + T_1)\tau^2\, d\tau,$$
$$A_3 = \frac{1}{(\Delta T)^5}\int_0^{\Delta T} v(\tau + T_1)\tau^3\, d\tau.$$

Here $v(t)$ denotes the uncorrected velocity record obtained by integration of the uncorrected acceleration record, $a(t)$. These velocities are obtained by assuming the uncorrected and the corrected acceleration vary linearly over each time increment of the original acceleration history. This is not exact for the corrected

acceleration record (because of the parabolic variation of the correction in time), but it is assumed that the acceleration history is discretized at sufficiently small time increments so that this is an insignificant error.

Reference

- "Amplitude curves," Section 34.1.2 of the Abaqus Analysis User's Guide

6.2 Abaqus/Aqua loading

- “Drag, inertia, and buoyancy loading,” Section 6.2.1
- “Airy wave theory,” Section 6.2.2
- “Stokes wave theory,” Section 6.2.3

6.2.1 DRAG, INERTIA, AND BUOYANCY LOADING

Product: Abaqus/Aqua

For beam and truss structures immersed in fluid (e.g., offshore piping and riser problems), Abaqus/Standard provides a capability for introducing drag forces via Morison's equation, inertia loads, and buoyancy loads. Fluid drag is associated with velocities due to steady currents and any waves that may have been specified. Fluid inertia is associated with wave accelerations. Buoyancy has two components: the hydrostatic pressure measured to the mean fluid level and the dynamic pressure caused by the presence of waves. Partial submergence is done automatically for all fluid load types.

Drag and inertia loads are considered in two forms: distributed loads along the length of the element (distributed drag loading is further divided into a component normal to the element's axis and a component along the tangent to the element) and point drag and inertia loads where the beam changes cross-section.

Buoyancy loading is applied with a "closed-end" assumption; that is, it is assumed that the element's ends can support buoyancy loading normal to the element's cross-section. If the ends of the element are actually "open ended"—that is, the element's ends cannot support fluid pressure loads—point buoyancy forces are provided to remove the buoyancy forces at the ends of the element.

This section documents the form of these loadings. It is assumed that the fluid particle velocities and accelerations are known as functions of the current spatial location; they are defined by superimposing the steady current velocity and the wave velocity.

Continuously distributed drag and inertia loading

Distributed load types "FDD," "WDD," "FDT," and "FI" on beam, truss, or rigid beam elements provide continuously distributed drag and inertia via Morison's equations. To specify this loading, the following definitions are used:

- $\mathbf{v}_f$ is the fluid particle velocity (defined by the steady current input and possibly additional contributions from wave definitions),
- $\mathbf{a}_f$ is the fluid particle acceleration when the waves are defined,
- $\mathbf{v}_p$ is the velocity of a point on the element (nonzero during dynamic analysis steps only),
- $\mathbf{a}_p$ is the acceleration of a point on the element (nonzero during dynamic analysis steps only),
- $\Delta\mathbf{v}$ $= \mathbf{v}_f - \alpha_R\mathbf{v}_p$ is the relative fluid velocity,
- $\mathbf{t}$ is a unit vector defining the axial direction at a point in the element,
- $\Delta\mathbf{v}_t$ $= (\Delta\mathbf{v}\cdot\mathbf{t})\mathbf{t}$ is the relative tangential (axial) velocity of the fluid,
- $\Delta\mathbf{v}_n$ $= \Delta\mathbf{v} - \Delta\mathbf{v}_t$ is the relative transverse velocity of the fluid,
- C_T is the tangential (axial) drag coefficient,
- C_D is the transverse drag coefficient,
- C_M is the transverse inertia coefficient,
- C_A is the transverse added mass coefficient,
- ρ is the fluid density,

α_R is the structural velocity factor, and

h is the exponent for tangential drag.

Then, the transverse drag force per unit length on the member is

$$\mathbf{F}_D = \frac{1}{2}\rho C_D D \Delta\mathbf{v}_n (\Delta\mathbf{v}_n \cdot \Delta\mathbf{v}_n)^{\frac{1}{2}}.$$

The tangential drag force per unit length is

$$\mathbf{F}_T = \frac{1}{2}\rho\, C_T \pi D \Delta\mathbf{v}_t |\Delta\mathbf{v}_t|^{h-1},$$

and the inertia force per unit length is

$$\mathbf{F}_I = \frac{1}{4}\rho\pi D^2 \left[C_M(\mathbf{a}_f - \mathbf{a}_f \cdot \mathbf{t}\,\mathbf{t}) + C_A(\mathbf{a}_p - \mathbf{a}_p \cdot \mathbf{t}\,\mathbf{t})\right].$$

Only transverse drag is implemented for wind loading.

Point drag and inertia loading

At points where the section changes size and, thus, exposes an end area to the fluid (see Figure 6.2.1–1), additional drag and inertia forces arise.

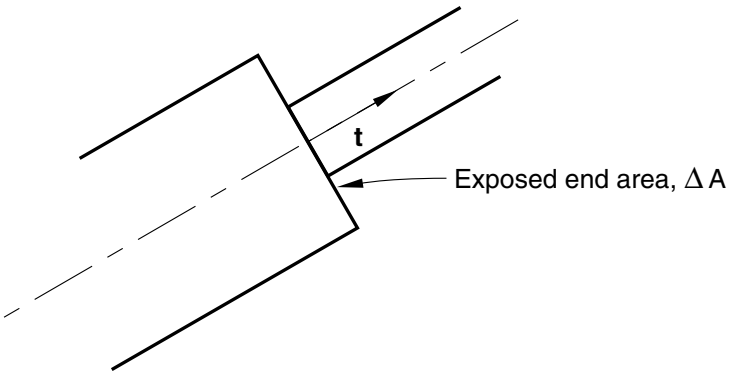


Figure 6.2.1–1 Change in section of an immersed beam.

For this case we need the additional definitions:

ΔA is the change in area of the section,

$\mathbf{t}$ is an outward normal to the exposed area (see Figure 6.2.1–1),

C_n is the drag coefficient associated with the discontinuity,

K_{ts} is the tangential inertia coefficient associated with the discontinuity,

L_{ts} is the tangential added mass coefficient associated with the discontinuity, and

F_{1s}, F_{2s} are fluid and structural acceleration shape factors for the tangential inertia term.

Then the drag force on the transition section is

$$\mathbf{F}_{ds} = \begin{cases} \frac{1}{2}\rho\, C_n \Delta A \Delta \mathbf{v}_{ft} |\Delta \mathbf{v}_{ft}| & \text{for} \quad \mathbf{t} \cdot \Delta \mathbf{v}_{ft} \leq 0, \\ \mathbf{0} & \text{for} \quad \mathbf{t} \cdot \Delta \mathbf{v}_{ft} > 0. \end{cases}$$

This force is nonzero only when the relative velocity of the fluid has a negative projection on the outward normal; i.e., when the fluid is flowing against the exposed surface. The inertia force is

$$\mathbf{F}_{Is} = \rho K_{ts} F_{1s} \mathbf{a}_f \cdot \mathbf{t}\, \mathbf{t} - \rho L_{ts} F_{2s} \mathbf{a}_p \cdot \mathbf{t}\, \mathbf{t}.$$

Distributed buoyancy

Abaqus assumes closed-end conditions when computing the distributed buoyancy loads (load type PB) for beams, pipes, rigid beams, and elbows. An open-end condition can be simulated by using load type TSB to remove the buoyancy load from the end of the element.

For truss elements the effect of buoyancy is simply Archimedes' principle; that is, a vertical force equal to the weight of the displaced fluid is applied to the element.

For buoyancy loading the beam is assumed to have a uniform circular pipe section with the effective diameter specified as part of the loading definition.

Consider a pressure field that varies with ***z***, the vertical coordinate, where $z = \mathbf{k} \cdot \mathbf{x}$ and $\mathbf{k}$ is the unit vector pointing in the vertical direction. Then

$$\hat{p} = p(z)\,.$$

For hydrostatic pressure the dependence on the vertical coordinate is linear in ***z***,

$$\gamma\hat{p} = \rho g(z_0 - z),$$

where z_0 is the vertical location of the free surface of the fluid (either the free surface elevation of the fluid inside the pipe or the elevation of the mean water level for external pressure), ρ is the density of the fluid, and ***g*** is the acceleration due to gravity. If waves are defined, the wave field generates a dynamic pressure effect. In this case the pressure field has a nonlinear variation with respect to location. Under the assumption of waves with small amplitude relative to the wave length and depth of the sea, we can assume that the dynamic pressure is slowly varying with respect to the wave direction; hence, derivatives of the pressure with respect to the coordinates in the plane parallel to the still water surface are neglected. However, the dynamic pressure field has a nonlinear dependence on ***z***.

The pressure field results in a nodal loading contribution that can be written as follows. Let $\delta\Pi_p$ be the pressure loading contribution to the weak form of the equilibrium equations. This contribution can be written in terms of a nodal load vector, $\mathbf{p}_N$, as

$$\delta\Pi_p = \mathbf{p}_N \cdot \delta\mathbf{x}_N\,,$$

where the nodal load vector is

$$\mathbf{p}_N = \int_0^l \pi R^2 \left[\hat{p} \frac{d\mathbf{x}}{ds} \frac{dN_N}{ds} + \frac{\partial \hat{p}}{\partial z} N_N \mathbf{k} \right] ds \,,$$

where R is the radius of the pipe (internal or external), $\hat{p}$ is the total pressure (including hydrostatic and dynamic pressure), N_N are the shape functions associated with the nodes on the element, and l is the current length of the element.

Point buoyancy

Point buoyancy is needed to apply discrete buoyant forces on an exposed surface. Point buoyancy can be used to remove the closed-end loading condition on beam, pipe, or rigid beam elements by applying a negative load on the exposed area. Use the following definitions:

z	is the elevation of the point considered,
ρ_w	is the mass density of the fluid outside the pipe,
ρ_f	is the mass density of the fluid inside the pipe,
z_w	is the free surface elevation of the fluid outside the pipe,
z_f	is the free surface elevation of the fluid inside the pipe,
$\mathbf{t}$	is the outward normal to the exposed area, and
$\mathbf{g}$	is the gravitational acceleration. In Abaqus it is assumed that $\mathbf{g} = -g\,\mathbf{k}$.

The buoyancy force is

$$\mathbf{F}_{pb} = -g \Delta A \big[f_1 \rho_w (z_w - z) - f_2 \rho_f (z_f - z) \big] \mathbf{t},$$

where

$$f_1 = \begin{cases} 0 & \text{if the elevation is above } z_w \\ 1 & \text{otherwise} \end{cases}$$

and

$$f_2 = \begin{cases} 0 & \text{if the elevation is above } z_f \\ 1 & \text{otherwise.} \end{cases}$$

Load stiffness

To ensure the quadratic convergence of the Newton method in Abaqus, it is necessary to calculate the changes in the above forces with respect to changes in the kinematic solution for the structure. Thus, a load stiffness is calculated for all of these load types.

Reference

- "Abaqus/Aqua analysis," Section 6.11.1 of the Abaqus Analysis User's Guide

6.2.2 AIRY WAVE THEORY

Product: Abaqus/Aqua

This is a linearized wave theory based on irrotational flow of an inviscid incompressible fluid. The linearization is achieved by assuming the wave height a is small compared to the wavelength λ and the still water depth. It is also assumed that the fluid is of uniform depth (that is, the bottom is flat).

Since we have irrotational flow, there exists a flow potential, ϕ, obeying

$$\nabla^2 \phi = 0 \tag{6.2.2–1}$$

and giving the fluid particle velocities as

$$\mathbf{v} = \frac{\partial \phi}{\partial \mathbf{x}}. \tag{6.2.2–2}$$

Now assume there is a potential energy per unit mass, G, (in this case associated with the gravity field). Then, equilibrium is given by

$$\rho \left[\frac{\partial \mathbf{v}}{\partial t} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}} \cdot \mathbf{v} \right] = -\rho \frac{\partial G}{\partial \mathbf{x}} - \frac{\partial p}{\partial \mathbf{x}}, \tag{6.2.2–3}$$

where ρ is the fluid density and p is the pressure. Substituting Equation 6.2.2–2 in Equation 6.2.2–3, we obtain

$$\rho \left[\frac{\partial^2 \phi}{\partial \mathbf{x} \partial t} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}} \cdot \mathbf{v} \right] = -\rho \frac{\partial G}{\partial \mathbf{x}} - \frac{\partial p}{\partial \mathbf{x}}.$$

This equation can be integrated with respect to position to give

$$\rho \left[\frac{\partial \phi}{\partial t} + \frac{1}{2} \mathbf{v} \cdot \mathbf{v} \right] = -\rho G - (p - p_0) + F(t), \tag{6.2.2–4}$$

since the fluid is assumed to be incompressible; thus, ρ is constant. Here F is an arbitrary function of t and p_0 is the pressure in the air just above the free surface. For convenience we choose $F(t) = 0$.

The term $1/2\mathbf{v} \cdot \mathbf{v}$ can be neglected compared to the other terms (this can be shown, from the resulting solution, to be consistent with the order of approximation that the wave height is small compared to the wavelength). By choosing the z-coordinate to point vertically upward, the gravity potential is conveniently chosen as

$$G = g(z - z_s),$$

where z_s is the undisturbed surface level. Equation 6.2.2–4 then becomes

$$\rho\frac{\partial\phi}{\partial t} + \rho g(z - z_s) + p - p_0 = 0. \tag{6.2.2–5}$$

From this equation the total pressure at a point below the instantaneous fluid surface is

$$p = p_0 + \rho g(z_s - z) + p_{dyn}\,.$$

Hence, the total pressure is the air pressure plus the hydrostatic pressure plus the dynamic pressure, p_{dyn}, where p_{dyn} is given by

$$p_{dyn} = -\rho\frac{\partial\phi}{\partial t}\,.$$

Let η be the elevation of the fluid surface at time t above the mean (undisturbed) fluid surface level, z_s. Since the position of the free surface is a part of the solution, there are two boundary conditions that must be applied on $z = z_s + \eta$. The first is a dynamic equilibrium condition at the interface between the fluid and the air. Since the interface is assumed to have no mass, the forces normal to the interface in the fluid and the air must be equal. If the surface tension of the interface is neglected, the pressure in the water and the air must be equal at the interface. Assuming that the pressure due to the motion of the air is negligible (which can be shown to be reasonable), the air pressure can be approximated by its undisturbed value (Whitham, 1974). The dynamic boundary condition then implies $p = p_0$, where p is the pressure in the water at the free surface and p_0 is the pressure in the undisturbed air. Since η is assumed to be small with respect to the depth of fluid, the boundary condition can be made linear by applying it at $z = z_s$ instead of at $z = z_s + \eta$.

With these assumptions Equation 6.2.2–5 provides the boundary term

$$\eta = -\frac{1}{g}\frac{\partial\phi}{\partial t}\bigg|_{z=z_s}. \tag{6.2.2–6}$$

The second boundary condition on the free surface comes from the kinematics of the free surface. Let the free surface be given by

$$f(x_1, x_2, z, t) = \eta(x_1, x_2, t) - (z - z_s) = 0.$$

The velocity of the fluid normal to the surface must be equal to the velocity of the surface normal to itself.

Differentiating this expression with respect to time yields

$$\frac{\partial\eta}{\partial t} - \frac{\partial z}{\partial t} = 0.$$

If we assume that the wave height is small compared to the wavelength ($\eta_{max} << \lambda$), then we can approximate $\frac{\partial z}{\partial t}$ by the velocity v_z, so that

$$\frac{\partial\eta}{\partial t} - v_z = \frac{\partial\eta}{\partial t} - \frac{\partial\phi}{\partial z} = 0 \text{ at } z = z_s. \tag{6.2.2–7}$$

Eliminating η between Equation 6.2.2–6 and Equation 6.2.2–7 gives

$$\frac{\partial^2 \phi}{\partial t^2} + g\frac{\partial \phi}{\partial z} = 0 \text{ at } z = z_s. \tag{6.2.2–8}$$

The boundary condition on the bottom, $z = z_b$, is

$$v_z = \frac{\partial \phi}{\partial z} = 0 \text{ at } z = z_b. \tag{6.2.2–9}$$

The problem is now defined by Equation 6.2.2–1, Equation 6.2.2–8, Equation 6.2.2–9, and the requirement that the solution be a plane wave periodic in the horizontal plane, such that

$$\phi(x_1, x_2, z, t) = \phi(\mathbf{d} \cdot \mathbf{x} - c\,t, z),$$

where $\mathbf{d}$ is the direction of wave propagation and $\boldsymbol{c}$ is the wave speed.

We solve the problem by assuming that $\phi = P(z)\Phi(x_1, x_2, t)$. Since $\boldsymbol{P}$ and Φ are independent functions, Equation 6.2.2–1 provides the two equations

$$\frac{d^2 P}{dz^2} - k^2 P = 0,$$
$$\frac{\partial^2 \Phi}{\partial x_1^2} + \frac{\partial^2 \Phi}{\partial x_2^2} + k^2 \Phi = 0,$$

where $\boldsymbol{k}$ is a constant.

Let $\mathbf{d} \cdot \mathbf{x} = s$ (so that $\boldsymbol{s}$ measures distance in the direction of travel of the wave). The solution to these equations is $P = \big(A_1 \cosh kz + A_2 \sinh kz\big)$ and $\Phi = \sin\big[k(s - c\,t) + \frac{2\pi}{360}\alpha\big]$; hence,

$$\phi = \big(A_1 \cosh kz + A_2 \sinh kz\big)\sin\big[k(s - c\,t) + \frac{2\pi}{360}\alpha\big], \tag{6.2.2–10}$$

where A_1 and A_2 are constants and α is the phase angle of the wave in degrees (α provides an arbitrary choice of origin in time and is chosen so that the vertical displacement of a fluid particle is a minimum when $s = 0$, $t = 0$, and $\alpha = 0$).

There is no motion at the bottom of the fluid in the vertical direction, so by Equation 6.2.2–9 we find

$$A_1 = -\frac{A_2}{\tanh kz_b}.$$

Substituting this into Equation 6.2.2–10 gives

$$\phi = C \cosh\big[k(z - z_b)\big] \sin\big[k(s - c\,t) + \frac{2\pi}{360}\alpha\big], \tag{6.2.2–11}$$

where $\boldsymbol{C}$ is a constant.

The dispersion relation can be obtained by substituting Equation 6.2.2–11 into Equation 6.2.2–8 and setting $z = z_s$, giving

$$c^2 = \frac{g}{k}\tanh\big[k(z_s - z_b)\big]. \tag{6.2.2–12}$$

The wave frequency ω is related to the wave period τ by $\omega = 2\pi/\tau$. The constant k is called the wave number and is related to the wavelength λ by $k = 2\pi/\lambda$, so that $\omega = ck$.

The free surface elevation above the undisturbed fluid surface, η, is given by Equation 6.2.2–6:

$$\eta = C\frac{kc}{g}\cosh\big[k(z_s - z_b)\big]\cos\big[k(s - c\,t) + \frac{2\pi}{360}\alpha\big]. \tag{6.2.2–13}$$

Writing the wave amplitude (half the wave height) as a, this defines

$$a = -C\frac{kc}{g}\cosh\big[k(z_s - z_b)\big]$$

so that Equation 6.2.2–11 can be rewritten

$$\begin{aligned}\phi &= -\frac{ga}{kc}\frac{\cosh\big[k(z - z_b)\big]}{\cosh\big[k(z_s - z_b)\big]}\sin\big[k(s - c\,t) + \frac{2\pi}{360}\alpha\big] \\ &= -\frac{ga\tau}{2\pi}\frac{\cosh\big[(2\pi/\lambda)(z - z_b)\big]}{\cosh\big[(2\pi/\lambda)(z_s - z_b)\big]}\sin 2\pi\big(\frac{s - c\,t}{\lambda} + \frac{\alpha}{360}\big).\end{aligned} \tag{6.2.2–14}$$

This solution provides fluid particle velocities $\mathbf{v} = \partial\phi/\partial\mathbf{x}$ and accelerations throughout the fluid for this one wave component. The term $\frac{1}{2}(\mathbf{v}\cdot\mathbf{v})$ in Equation 6.2.2–4 was neglected because the wave amplitude a is small compared to the wavelength λ. This implies, from Equation 6.2.2–3, that the fluid particle acceleration is approximated as $\mathbf{a} = \partial\mathbf{v}/\partial t$; that is, the convective part of the acceleration, $\partial\mathbf{v}/\partial\mathbf{x}\cdot\mathbf{v}$, is neglected.

Since the theory is linear, any set of waves can be superposed by linear superposition of its components:

$$\phi = \sum_{\text{components}} \phi_N,$$

where ϕ_N is the potential Equation 6.2.2–10 of the Nth wave component. Hence, the theory can be summarized as follows.

For $z \le z_s$:

Velocity potential:

$$\phi = -\frac{g}{2\pi}\sum_{\text{components}} a_N\tau_N\frac{\cosh\big[(2\pi/\lambda_N)(z - z_b)\big]}{\cosh\big[(2\pi/\lambda_N)(z_s - z_b)\big]}\sin 2\pi\left(\frac{s_N}{\lambda_N} - \frac{t}{\tau_N} + \frac{\alpha_N}{360}\right).$$

Fluid particle displacements:

horizontally,

$$u_i = \frac{g}{2\pi} \sum_{\text{components}} d_{Ni} \frac{a_N \tau_N^2}{\lambda_N} \frac{\cosh\left[(2\pi/\lambda_N)(z - z_b)\right]}{\cosh\left[(2\pi/\lambda_N)(z_s - z_b)\right]} \sin 2\pi \left(\frac{s_N}{\lambda_N} - \frac{t}{\tau_N} + \frac{\alpha_N}{360} \right),$$

and vertically,

$$u_z = -\frac{g}{2\pi} \sum_{\text{components}} \frac{a_N \tau_N^2}{\lambda_N} \frac{\sinh\left[(2\pi/\lambda_N)(z - z_b)\right]}{\cosh\left[(2\pi/\lambda_N)(z_s - z_b)\right]} \cos 2\pi \left(\frac{s_N}{\lambda_N} - \frac{t}{\tau_N} + \frac{\alpha_N}{360} \right).$$

Fluid particle velocities:

horizontally,

$$v_i = -g \sum_{\text{components}} d_{Ni} \frac{a_N \tau_N}{\lambda_N} \frac{\cosh\left[(2\pi/\lambda_N)(z - z_b)\right]}{\cosh\left[(2\pi/\lambda_N)(z_s - z_b)\right]} \cos 2\pi \left(\frac{s_N}{\lambda_N} - \frac{t}{\tau_N} + \frac{\alpha_N}{360} \right),$$

and vertically,

$$v_z = -g \sum_{\text{components}} \frac{a_N \tau_N}{\lambda_N} \frac{\sinh\left[(2\pi/\lambda_N)(z - z_b)\right]}{\cosh\left[(2\pi/\lambda_N)(z_s - z_b)\right]} \sin 2\pi \left(\frac{s_N}{\lambda_N} - \frac{t}{\tau_N} + \frac{\alpha_N}{360} \right).$$

Fluid particle accelerations:

horizontally,

$$a_i = -2\pi g \sum_{\text{components}} d_{Ni} \frac{a_N}{\lambda_N} \frac{\cosh\left[(2\pi/\lambda_N)(z - z_b)\right]}{\cosh\left[(2\pi/\lambda_N)(z_s - z_b)\right]} \sin 2\pi \left(\frac{s_N}{\lambda_N} - \frac{t}{\tau_N} + \frac{\alpha_N}{360} \right),$$

and vertically,

$$a_z = 2\pi g \sum_{\text{components}} \frac{a_N}{\lambda_N} \frac{\sinh\left[(2\pi/\lambda_N)(z - z_b)\right]}{\cosh\left[(2\pi/\lambda_N)(z_s - z_b)\right]} \cos 2\pi \left(\frac{s_N}{\lambda_N} - \frac{t}{\tau_N} + \frac{\alpha_N}{360} \right).$$

Free surface profile:

$$z_s + \eta = z_s - \sum_{\text{components}} a_N \cos 2\pi \left(\frac{s_N}{\lambda_N} - \frac{t}{\tau_N} + \frac{\alpha_N}{360} \right).$$

Dispersion relation for each mode:

$$\frac{2\pi}{\tau_N^2} = g \frac{1}{\lambda_N} \tanh\left[\frac{2\pi}{\lambda_N}(z_s - z_b)\right].$$

Dynamic pressure:

$$p_{dyn} = -\rho g \sum_{\text{components}} a_N \frac{\cosh\left[(2\pi/\lambda_N)(z - z_b)\right]}{\cosh\left[(2\pi/\lambda_N)(z_s - z_b)\right]} \cos 2\pi \left(\frac{s_N}{\lambda_N} - \frac{t}{\tau_N} + \frac{\alpha_N}{360} \right),$$

$$\frac{\partial p_{dyn}}{\partial z} = -2\pi\rho g \sum_{\text{components}} \frac{a_N}{\lambda_N} \frac{\sinh\left[(2\pi/\lambda_N)(z - z_b)\right]}{\cosh\left[(2\pi/\lambda_N)(z_s - z_b)\right]} \cos 2\pi \left(\frac{s_N}{\lambda_N} - \frac{t}{\tau_N} + \frac{\alpha_N}{360} \right),$$

and

$$\frac{\partial^2 p_{dyn}}{\partial z^2} = -4\pi^2\rho g \sum_{\text{components}} \frac{a_N}{\lambda_N^2} \frac{\cosh\left[(2\pi/\lambda_N)(z - z_b)\right]}{\cosh\left[(2\pi/\lambda_N)(z_s - z_b)\right]} \cos 2\pi \left(\frac{s_N}{\lambda_N} - \frac{t}{\tau_N} + \frac{\alpha_N}{360} \right).$$

Airy wave theory is a linearized theory; however, the wave amplitude can be large compared with the size of a structure. We, therefore, must make an assumption about the wave kinematics below a crest and above the mean water level. The assumption used here follows modified Airy wave theory as described in Hansen (1988) and Faltinsen (1990). The free surface boundary condition has been made linear in Equation 6.2.2–6. Above the mean or undisturbed surface level z_s the velocity, acceleration, and dynamic pressure are extrapolated from their values at the mean surface level. Hence, for $z_s < z \leq \eta$

$$\mathbf{v} = \mathbf{v}\big|_{z_s}, \ \mathbf{a} = \mathbf{a}\big|_{z_s}, \text{ and } p_{dyn} = p_{dyn}\big|_{z_s}.$$

Accordingly,

$$\frac{\partial p_{dyn}}{\partial z} = 0 \quad \text{and} \quad \frac{\partial^2 p_{dyn}}{\partial z^2} = 0\,.$$

When a gravity wave is defined, the penetration of the structure into the fluid must be calculated. Although the Airy wave theory assumes that the fluid displacements are small with respect to the wavelength and the fluid depth, they cannot be small with respect to the dimensions of the structure immersed in the fluid. Hence, the instantaneous fluid surface is used to determine if a point on the structure sees loads due to the presence of the fluid.

The Airy wave field is a spatial description of the wave field. The wave field defines velocity, acceleration, and dynamic pressure at spatial locations for all values of time. Hence, the velocity, acceleration, and dynamic pressure are determined by using the current (for geometrically nonlinear analysis) or reference (for geometrically linear analysis) location of the structure at the current time in the appropriate equations. The time used in the wave field equations is the total time for the analysis, which accumulates over all steps in the analysis (static, dynamic, etc.).

Reference

- "Abaqus/Aqua analysis," Section 6.11.1 of the Abaqus Analysis User's Guide

6.2.3 STOKES WAVE THEORY

Product: Abaqus/Aqua

Assume that an infinite series of plane, uniform waves travels through the fluid in the positive *S*-direction. The *z*-coordinate is chosen to be positive in the vertical direction, so the gravity potential is $G = g(z - z_0)$, where z_0 is an arbitrary datum.

Assume that the fluid is inviscid and incompressible. The fluid particle velocities are derivable from a flow potential

$$\nabla^2 \phi = 0 \tag{6.2.3–1}$$

$$\mathbf{v} = -\frac{\partial \phi}{\partial \mathbf{x}}.$$

Equilibrium is

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}} \cdot \mathbf{v}\right) = -\rho \frac{\partial G}{\partial \mathbf{x}} - \frac{\partial p}{\partial \mathbf{x}},$$

where ρ is the fluid density and p is the pressure. Writing $\partial \mathbf{v}/\partial t$ in terms of the flow potential then gives

$$\rho\left(\frac{\partial^2 \phi}{\partial \mathbf{x} \partial t} - \frac{\partial \mathbf{v}}{\partial \mathbf{x}} \cdot \mathbf{v}\right) = \rho \frac{\partial G}{\partial \mathbf{x}} + \frac{\partial p}{\partial \mathbf{x}}.$$

Integrating with respect to $\mathbf{x}$ (note that ρ is constant since the fluid is incompressible) gives the Bernoulli equation

$$\frac{\partial \phi}{\partial t} - \frac{1}{2}\mathbf{v} \cdot \mathbf{v} = G + \frac{p - p_0}{\rho} + gF(t),$$

where $F(t)$ is an arbitrary function (which for convenience is set to zero) and p_0 is the atmospheric pressure. Substituting in the gravity potential, this is

$$\frac{\partial \phi}{\partial t} - \frac{1}{2}\mathbf{v} \cdot \mathbf{v} = g(z - z_s) + \frac{p - p_0}{\rho},$$

where z_s is the undisturbed surface level. From this equation the total pressure at a point below the instantaneous fluid surface is

$$p = p_0 + \rho g(z_s - z) + p_{dyn}\,.$$

Hence, the total pressure is the air pressure plus the hydrostatic pressure plus the dynamic pressure, p_{dyn}, where p_{dyn} is given by

$$p_{dyn} = \rho \left(\frac{\partial \phi}{\partial t} - \frac{1}{2} \mathbf{v} \cdot \mathbf{v} \right) .$$

Let η be the elevation of the free surface above this level. At the free surface the Bernoulli equation is

$$\left(\frac{\partial \phi}{\partial t} - \frac{1}{2} \mathbf{v} \cdot \mathbf{v} \right) \Big|_{z=\eta+z_s} = g\eta,$$

assuming the pressure at the surface is negligible.

Assuming the waves are uniform, of wavelength λ and period τ, and that they travel in the positive S-direction means that the solution as a function of S and t must appear in terms of a phase angle

$$\theta = 2\pi \left(\frac{S}{\lambda} - \frac{t}{\tau} + \frac{\alpha}{360} \right) = \frac{2\pi}{\lambda} \left(S - \bar{c}t + \frac{\lambda \alpha}{360} \right),$$

where $\bar{c} = \frac{\lambda}{\tau}$ is the wave celerity. This means that, for any function in the solution,

$$\frac{\partial}{\partial t} = -\bar{c} \frac{\partial}{\partial s}.$$

Thus, at the free surface boundary

$$\frac{\partial \phi}{\partial t} = -\bar{c} \frac{\partial \phi}{\partial s} = \bar{c} v_s,$$

and the Bernoulli equation at the free surface is

$$(-\bar{c} v_s + 1/2(\mathbf{v} \cdot \mathbf{v})) \big|_{z=\eta+z_S} = -g\eta$$

or

$$(v_z^2 + (v_s - \bar{c})^2) \big|_{z=\eta+z_S} = \bar{c}^2 - zg\eta. \tag{6.2.3–2}$$

A further boundary condition at the free surface is that the fluid particle velocity relative to the wave celerity must be tangential to the slope of the wave:

$$\frac{v_z}{v_s - \bar{c}} |_{z=\eta+z_s} = \frac{\partial \eta}{\partial s}. \tag{6.2.3–3}$$

At the seabed ($z = z_b$), there is no fluid motion in the vertical direction:

$$-v_z = \frac{\partial \phi}{\partial z} |_{z=z_b} = 0.$$

The problem now consists of finding a potential function, ϕ, that satisfies Equation 6.2.3–1—the boundary condition at the seabed—as well as the boundary conditions at the surface—Equation 6.2.3–2 and Equation 6.2.3–3.

Stokes proposed a power series solution to this problem, and Skjelbreia and Hendrickson (1960) have obtained that solution to fifth-order. The potential function is assumed to be

$$\begin{aligned}\frac{\beta\phi}{\bar{c}} = \frac{2\pi\phi}{\lambda\bar{c}} &= (\mu A_{11} + \mu^3 A_{13} + \mu^5 A_{15})\cosh[\beta(z-z_b)]\sin\theta \\ &- (\mu^2 A_{22} + \mu^4 A_{24})\cosh[2\beta(z-z_b)]\sin 2\theta \\ &+ (\mu^3 A_{33} + \mu^5 A_{35})\cosh[3\beta(z-z_b)]\sin 3\theta \\ &- \mu^4 A_{44}\cosh[4\beta(z-z_b)]\sin 4\theta \\ &+ \mu^5 A_{55}\cosh[5\beta(z-z_b)]\sin 5\theta,\end{aligned} \tag{6.2.3–4}$$

where $\beta = 2\pi/\lambda$, the A_{ij} are constants that depend on the ratio of water depth to wavelength $(z_s - z_b)/\lambda$, and μ is a parameter. The wave profile, $\eta(\theta)$, is assumed to be

$$\begin{aligned}\beta\eta = &- \mu\cos\theta \\ &+ (\mu^2 B_{22} + \mu^4 B_{24})\cos 2\theta \\ &- (\mu^3 B_{33} + \mu^5 B_{35})\cos 3\theta \\ &+ \mu^4 B_{44}\cos 4\theta \\ &- \mu^5 B_{55}\cos 5\theta,\end{aligned} \tag{6.2.3–5}$$

where the B_{ij} are constants for a given water depth and wavelength. Finally, it is assumed that

$$\beta\bar{c}^2 = c_0^2(1 + \mu^2 C_1 + \mu^4 C_2) \tag{6.2.3–6}$$

and that

$$\beta F = \mu^2 C_3 + \mu^4 C_4.$$

Skjelbreia and Hendrickson obtain the 18 constants $A_{ij},\ B_{ij}, C_i$, and c_0 from matching terms in equal powers of μ and $\cos\theta$ in the free surface boundary conditions, Equation 6.2.3–2 and Equation 6.2.3–3. They give the constants as functions of $s = \sinh[\beta(z_s - z_b)],\ c = \cosh[\beta(z_s - z_b)]$, as

$$c_0^2 = g\,s/c,$$
$$A_{11} = 1/s,$$

$$A_{13} = -c^2 \frac{5c^2+1}{8s^5},$$

$$A_{15} = -\frac{1184c^1 - 1440c^8 - 1992c^6 + 2641c^4 - 249c^2 + 18}{1536s^{11}},$$

$$A_{22} = \frac{3}{8s^4},$$

$$A_{24} = \frac{192c^8 - 424c^6 - 312c^4 + 480c^2 - 17}{768s^{10}},$$

$$A_{33} = \frac{13 - 4c^2}{64s^7},$$

$$A_{35} = \frac{512c^{12} + 4224c^{10} - 6800c^8 - 12808c^6 + 167704c^4 - 3154c^2 + 107}{4096s^{13}(6c^2-1)},$$

$$A_{44} = \frac{80c^6 - 816c^4 + 1338c^2 - 197}{1536s^{10}(6c^2-1)},$$

$$A_{55} = -\frac{2880c^{10} - 72480c^8 + 324000c^6 - 432000c^4 + 163470c^2 - 16245}{61440s^{11}(6c^2-1)(8c^4-11c^2+3)},$$

$$B_{22} = c\frac{2c^2+1}{4s^3},$$

$$B_{24} = c\frac{272c^8 - 504c^6 - 192c^4 + 322c^2 + 21}{384s^9},$$

$$B_{33} = 3\frac{8c^6+1}{64s^6},$$

$$B_{35} = \frac{88128c^{14} - 208224c^{12} + 70848c^{10} + 54000c^8 - 21816c^6 + 6264c^4 - 54c^2 - 81}{12288s^{12}(6c^2-1)},$$

$$B_{44} = c\frac{768c^{10} - 448c^8 - 48c^6 + 48c^4 + 106c^2 - 21}{384s^9(6c^2-1)},$$

$$B_{55} = \frac{192000c^{16} - 262720c^{14} + 83680c^{12} + 20160c^{10} - 7280c^8}{12288s^{10}(6c^2-1)(8c^4-11c^2+3)}$$

$$+ \frac{7160c^6 - 1800c^4 - 1050c^2 + 225}{12288s^{10}(6c^2-1)(8c^4-11c^2+3)},$$

$$C_1 = \frac{8c^4 - 8c^2 + 9}{8s^4},$$

$$C_2 = \frac{3840c^{12} - 4096c^{10} - 2592c^8 - 1008c^6 + 5944c^4 - 1830c^2 + 147}{512s^{10}(6c^2-1)}.$$

Skjelbreia and Hendrickson (1960) have a factor +2592 multiplying c^8 in the equation for C_2. This was corrected to −2592 by Nishimura et al. (1970).

They then obtain equations for $\beta(= 2\pi/\lambda)$ and μ. The wave height is

$$H = \eta_{\text{crest}} - \eta_{\text{trough}} = \eta|_{\theta=\pi} - \eta|_{\theta=0},$$

so Equation 6.2.3–5 gives

$$H = \frac{\lambda}{\pi}[\mu + \mu^3 B_{33} + \mu^5(B_{35} + B_{55})]. \tag{6.2.3–7}$$

Also, the form assumed for the wave celerity gives

$$\beta\bar{c}^2 = \frac{2\pi\lambda}{\tau^2} = c_0^2(1 + \mu^2 C_1 + \mu^4 C_2). \tag{6.2.3–8}$$

Given the wave period, wave height, and water depth, Equation 6.2.3–7 and Equation 6.2.3–8 must be solved simultaneously for the wavelength, λ, and the parameter μ. This is done with a Newton method, using the Airy (linear) wave solution as an initial guess.

Fluid particle velocities and accelerations for Stokes 5th order wave

The flow potential has been approximated as

$$\phi = \frac{\lambda\bar{c}}{2\pi}\sum_{n=1}^{5} D_n \cosh\frac{2\pi n}{\lambda}(z - z_b)\sin 2\pi n\left(\frac{S}{\lambda} - \frac{t}{\tau} + \frac{\alpha}{360}\right),$$

where

$$\begin{aligned}
D_1 &= \mu A_{11} + \mu^3 A_{13} + \mu^5 A_{15},\\
D_2 &= -(\mu^2 A_{22} + \mu^4 A_{24}),\\
D_3 &= \mu^3 A_{33} + \mu^5 A_{35},\\
D_4 &= -\mu^4 A_{44},\\
D_5 &= \mu^5 A_{55}.
\end{aligned}$$

The fluid particle velocities are

$$v_s = -\frac{\partial\phi}{\partial s} \quad \text{and} \quad v_z = -\frac{\partial\phi}{\partial z},$$

and the fluid particle accelerations are

$$a_s = \frac{\partial v_s}{\partial t} + \frac{\partial v_s}{\partial s}v_s + \frac{\partial v_s}{\partial z}v_z \quad \text{and} \quad a_z = \frac{\partial v_z}{\partial t} + \frac{\partial v_z}{\partial s}v_s + \frac{\partial v_z}{\partial z}v_z.$$

Since the solution appears in terms of the phase angle θ, it follows that

$$\frac{\partial \mathbf{v}}{\partial t} = -\bar{c}\frac{\partial \mathbf{v}}{\partial s} = -\frac{\lambda}{\tau}\frac{\partial \mathbf{v}}{\partial s}.$$

This allows the acceleration components to be written as

$$a_s = (v_s - \bar{c})\,\frac{\partial v_s}{\partial s} + v_z\frac{\partial v_s}{\partial z} \quad \text{and} \quad a_z = (v_s - \bar{c})\,\frac{\partial v_z}{\partial s} + v_z\frac{\partial v_z}{\partial z}.$$

Recall the expression for the dynamic pressure:

$$p_{dyn} = \rho\left(\frac{\partial \phi}{\partial t} - \frac{1}{2}\mathbf{v}\cdot\mathbf{v}\right).$$

Substitution of the expression for ϕ yields:

$$\begin{aligned}
v_s &= -\bar{c}e_1,\\
v_z &= -\bar{c}e_2,\\
a_s &= \frac{2\pi\bar{c}}{\tau}\big[-(1+e_1)e_3 + e_2e_4\big],\\
a_z &= \frac{2\pi\bar{c}}{\tau}\big[(1+e_1)e_4 + e_2e_3\big],\\
p_{dyn} &= -\rho\left(\frac{\lambda\bar{c}}{\tau}e_1 + \frac{1}{2}\mathbf{v}\cdot\mathbf{v}\right),\\
\frac{\partial p_{dyn}}{\partial z} &= -\frac{\rho 2\pi\bar{c}}{\tau}\left(e_4 + e_1e_4 + e_2e_3\right),\\
\frac{\partial^2 p_{dyn}}{z^2} &= -\frac{\rho 4\pi^2\bar{c}}{\tau\lambda}\left(e_6 + e_1e_6 + e_4e_4 + e_2e_5 + e_3e_3\right),
\end{aligned}$$

where

$$\begin{aligned}
e_1 &= \sum_{n=1}^{5} nD_n \cosh\frac{2\pi n}{\lambda}(z - z_b)\cos 2\pi n\left(\frac{S}{\lambda} - \frac{t}{\tau} + \frac{\alpha}{360}\right),\\
e_2 &= \sum_{n=1}^{5} nD_n \sinh\frac{2\pi n}{\lambda}(z - z_b)\sin 2\pi n\left(\frac{S}{\lambda} - \frac{t}{\tau} + \frac{\alpha}{360}\right),\\
e_3 &= \sum_{n=1}^{5} n^2D_n \cosh\frac{2\pi n}{\lambda}(z - z_b)\sin 2\pi n\left(\frac{S}{\lambda} - \frac{t}{\tau} + \frac{\alpha}{360}\right),\\
e_4 &= \sum_{n=1}^{5} n^2D_n \sinh\frac{2\pi n}{\lambda}(z - z_b)\cos 2\pi n\left(\frac{S}{\lambda} - \frac{t}{\tau} + \frac{\alpha}{360}\right),
\end{aligned}$$

$$e_5 = \sum_{n=1}^{5} n^3 D_n \sinh \frac{2\pi n}{\lambda}(z - z_b) \sin 2\pi n \left(\frac{S}{\lambda} - \frac{t}{\tau} + \frac{\alpha}{360} \right),$$

and

$$e_6 = \sum_{n=1}^{5} n^3 D_n \cosh \frac{2\pi n}{\lambda}(z - z_b) \cos 2\pi n \left(\frac{S}{\lambda} - \frac{t}{\tau} + \frac{\alpha}{360} \right).$$

Finally, the surface position is given as

$$\eta = \frac{\lambda}{2\pi} \sum_{n=1}^{5} E_n \cos 2\pi n \left(\frac{S}{\lambda} - \frac{t}{\tau} + \frac{\alpha}{360} \right),$$

where

$$\begin{aligned} E_1 &= -\mu, \\ E_2 &= \mu^2 B_{22} + \mu^4 B_{24}, \\ E_3 &= -(\mu^3 B_{33} + \mu^5 B_{35}), \\ E_4 &= \mu^4 B_{44}, \\ E_5 &= -\mu^5 B_{55}. \end{aligned}$$

The Stokes wave field is a spatial description of the wave field. All wave field quantities are calculated up to the instantaneous fluid level. The wave field defines velocity, acceleration, and dynamic pressure at spatial locations for all values of time. Hence, the velocity, acceleration, and dynamic pressure are determined by using the current (for geometrically nonlinear analysis) or reference (for geometrically linear analysis) location of the structure at the current time in the appropriate equations. The time used in the wave field equations is the total time for the analysis, which accumulates over all steps in the analysis (static, dynamic, etc.).

Reference

- "Abaqus/Aqua analysis," Section 6.11.1 of the Abaqus Analysis User's Guide

6.3 Incident wave loading

- “Loading due to an incident dilatational wave field,” Section 6.3.1

6.3.1 LOADING DUE TO AN INCIDENT DILATATIONAL WAVE FIELD

Products: Abaqus/Standard Abaqus/Explicit

Abaqus provides a capability for introducing generalized forces on acoustic and solid media associated with the arrival of dilatational waves. This capability applies to acoustic scattering problems and problems involving blast loads in air or water. Thus, the capability is available in transient dynamic procedures in Abaqus/Standard and Abaqus/Explicit.

Consider a dynamic problem involving fluid and coupled solid domains, excited by a propagating wave in the fluid arriving from outside these domains. When the mechanics of a fluid can be described as linear, wave fields in the fluid can be superimposed. Therefore, the observed total pressure in the fluid can be decomposed into two components: the incident wave itself, which is known, and the wave field excited in the fluid due to reflections at the fluid boundaries and interactions with the solid. To compute the latter, "scattered" solution, it is sufficient to apply loads at the boundaries of the fluid and solid domains corresponding to the effects of the incident wave field.

The fluid mechanical behavior is nonlinear when the fluid is capable of undergoing cavitation. In that case superposition of the incident wave and the response due to the boundaries, the solid, and the cavitating fluid regions to the incident wave loading is not valid. A total wave formulation (see "Coupled acoustic-structural medium analysis," Section 2.9.1) is used in Abaqus/Explicit to handle the incident wave loads on an acoustic medium capable of undergoing cavitation. In the total wave formulation the incident wave loading is applied as traction on the boundary of the modeled acoustic domain as the wave impinges on this domain from an external source. The default scattered wave formulation applicable in the absence of cavitation is presented below.

Scattering formulation

The equations for coupled fluid-solid interaction in Abaqus are developed in "Coupled acoustic-structural medium analysis," Section 2.9.1. Here, we proceed from the coupled system:

$$\begin{aligned}(M_{\mathrm{f}}^{PQ}+M_{\mathrm{fr}}^{PQ})\ddot{p}^{Q}+(C_{\mathrm{f}}^{PQ}+C_{\mathrm{fr}}^{PQ})\dot{p}^{Q}+(K_{\mathrm{f}}^{PQ}+K_{\mathrm{fr}}^{PQ}+K_{\mathrm{fi}}^{PQ})p^{Q} &= +\left[S_{\mathrm{fs}}^{PM}\right]T^{M}+P_{f}^{P}\\ M^{NM}\ddot{u}^{M}+C_{(m)}^{NM}\dot{u}^{M}+I^{N} &= -\left[S_{\mathrm{fs}}^{QN}\right]^{T}p^{Q}+P^{N}\end{aligned} \tag{6.3.1–1}$$

These equations couple the total pressure in the fluid to the displacements in the solid. They are valid for any combination of fluid and solid domains in a particular model: the matrix S_{fs} is defined on all the interacting fluid and solid surfaces. The fluid traction

$$T=\frac{-1}{\rho_f}\frac{\partial p}{\partial \mathbf{x}}\cdot\mathbf{n}^{-}=\left(\ddot{\mathbf{u}}_f+\frac{\gamma}{\rho_f}\dot{\mathbf{u}}_f\right)\cdot\mathbf{n}^{-} \tag{6.3.1–2}$$

is a quantity (with dimensions of acceleration) that describes the mechanism by which the solid motion drives the fluid; the fluid drives the solid by the pressure on the solid surface. In Equation 6.3.1–2, $\mathbf{n}^-$ is the inward normal on the boundary.

To proceed, we decompose the total pressure into the known incident wave component and the unknown scattered component:

$$p \equiv p_I + p_S.$$

Then

$$(M_{\mathrm{f}}^{PQ} + M_{\mathrm{fr}}^{PQ})\ddot{p}_S^Q + (C_{\mathrm{f}}^{PQ} + C_{\mathrm{fr}}^{PQ})\dot{p}_S^Q + (K_{\mathrm{f}}^{PQ} + K_{\mathrm{fr}}^{PQ} + K_{\mathrm{fi}}^{PQ})p_S^Q = + \left[S_{\mathrm{fs}}^{PM}\right]\, T_S^M + {P_f}^P - \left((M_{\mathrm{f}}^{PQ} + M_{\mathrm{fr}}^{PQ})\ddot{p}_I^Q + (C_{\mathrm{f}}^{PQ} + C_{\mathrm{fr}}^{PQ})\dot{p}_I^Q + (K_{\mathrm{f}}^{PQ} + K_{\mathrm{fr}}^{PQ} + K_{\mathrm{fi}}^{PQ})p_I^Q\right) + \left[S_{\mathrm{fs}}^{PM}\right]\, T_I^M, \tag{6.3.1–3}$$

and

$$M^{NM}\ddot{u}^M + C_{(m)}^{NM}\dot{u}^M + I^N = -\left[S_{\mathrm{fs}}^{QN}\right]^T\, \left[{p_S}^Q + {p_I}^Q\right] + P^N. \tag{6.3.1–4}$$

We see that the displacements in the solid are driven by the sum of the incident pressure, which forms an applied boundary traction, and the scattered pressure, the unknown field in the fluid.

The incident field is independent of the scattered field by convention. Therefore, it can be shown that it is a solution to the equation

$$(M_{\mathrm{f}}^{PQ} + M_{\mathrm{fr}}^{PQ})\ddot{p}_I^Q + (C_{\mathrm{f}}^{PQ} + C_{\mathrm{fr}}^{PQ})\dot{p}_I^Q + (K_{\mathrm{f}}^{PQ} + K_{\mathrm{fr}}^{PQ} + K_{\mathrm{fi}}^{PQ})p_I^Q = \left[S_{\mathrm{fs}}^{PM}\right]\, T_I^M,$$

so the fluid domain equation reduces to

$$\begin{aligned}(M_{\mathrm{f}}^{PQ} + M_{\mathrm{fr}}^{PQ})\ddot{p}_S^Q + (C_{\mathrm{f}}^{PQ} + C_{\mathrm{fr}}^{PQ})\dot{p}_S^Q + (K_{\mathrm{f}}^{PQ} + K_{\mathrm{fr}}^{PQ} + K_{\mathrm{fi}}^{PQ})p_S^Q \\ = \left[S_{\mathrm{fs}}^{PM}\right]\, T_S^M + {P_f}^P.\end{aligned} \tag{6.3.1–5}$$

This equation, using the scattered pressure as the unknown field variable, is solved together with Equation 6.3.1–4 using explicit or implicit integration or using the steady-state formulation ("Coupled acoustic-structural medium analysis," Section 2.9.1).

The scattered fluid traction, T_S, depends on the incident pressure through the decomposition above and the solid motion at the boundary. In the absence of an impedance condition at this boundary, this results in the relation

$$
\begin{aligned}
T_S &= \frac{-1}{\rho_f}\frac{\partial p_S}{\partial \mathbf{x}}\cdot\mathbf{n}^- = \left(\ddot{\mathbf{u}}+\frac{\gamma}{\rho_f}\dot{\mathbf{u}}\right)\cdot\mathbf{n}^- - T_I \\
&= \left(\ddot{\mathbf{u}}+\frac{\gamma}{\rho_f}\dot{\mathbf{u}}\right)\cdot\mathbf{n}^- - \left(\ddot{\mathbf{u}}_I+\frac{\gamma}{\rho_f}\dot{\mathbf{u}}_I\right)\cdot\mathbf{n}^- \\
&= \left(\ddot{\mathbf{u}}+\frac{\gamma}{\rho_f}\dot{\mathbf{u}}\right)\cdot\mathbf{n}^- + \frac{1}{\rho_f}\frac{\partial p_I}{\partial \mathbf{x}}\cdot\mathbf{n}^-.
\end{aligned}
\tag{6.3.1–6}
$$

If an impedance condition is applied at the fluid-solid boundary, application of the total pressure decomposition and Equation 2.9.1–13 results in the relation

$$
\begin{aligned}
T_S = &\left(\ddot{\mathbf{u}}+\frac{\gamma}{\rho_f}\dot{\mathbf{u}}\right)\cdot\mathbf{n}^- + \frac{1}{\rho_f}\frac{\partial p_I}{\partial \mathbf{x}}\cdot\mathbf{n}^- \\
&- \frac{\gamma}{\rho_f}\frac{1}{c_1}p_S - \left(\frac{\gamma}{\rho_f}\frac{1}{k_1}+\frac{1}{c_1}\right)\dot{p}_S - \frac{1}{k_1}\ddot{p}_S \\
&- \frac{\gamma}{\rho_f}\frac{1}{c_1}p_I - \left(\frac{\gamma}{\rho_f}\frac{1}{k_1}+\frac{1}{c_1}\right)\dot{p}_I - \frac{1}{k_1}\ddot{p}_I.
\end{aligned}
\tag{6.3.1–7}
$$

Virtual mass formulation

The "virtual mass approximation" is a simplification of the general scattered-field form of the coupled system Equation 6.3.1–1. Physically, it corresponds to considering the fluid wave speed to be very large compared to the characteristic structural wave speeds of interest; that is, fluid disturbances propagate and distribute in the field at an extremely high rate compared to the disturbances in the structure. This can be modeled by imposing an incompressibility condition on the fluid. Formally, applying such a condition results in the suppression of the terms related to time derivatives, volumetric drag, and impedance in the fluid equation for the coupled system:

$$
\begin{aligned}
K_{\mathrm{f}}^{PQ}\,p^Q &= +\left[S_{\mathrm{fs}}^{PM}\right]\,\ddot{u}^M + \frac{1}{\rho_f}\left[S_{\mathrm{fs}}^{PM}\right]\,\mathbf{n}^-\cdot\frac{\partial p_I^M}{\partial \mathbf{x}} \\
M^{NM}\ddot{u}^M + C_{(m)}^{NM}\dot{u}^M + I^N &= -\left[S_{\mathrm{fs}}^{QN}\right]^T\,p^Q - \left[S_{\mathrm{fs}}^{QN}\right]^T\,p_I + P^N.
\end{aligned}
\tag{6.3.1–8}
$$

The fluid will also be assumed to have homogeneous mass density, ρ_f. The fluid pressures, p^Q, can be eliminated from the structural equation by inverting the fluid "stiffness," K_{f}^{PQ} to yield

$$
\begin{aligned}
M^{NM}\ddot{u}^M + C_{(m)}^{NM}\dot{u}^M + I^N = &-\left[S_{\mathrm{fs}}^{QN}\right]^T\,(K_{\mathrm{f}}^{PQ})^{-1}\,\left[S_{\mathrm{fs}}^{PM}\right]\,\ddot{u}^M \\
&-\left[S_{\mathrm{fs}}^{QN}\right]^T\,p_I - \left[S_{\mathrm{fs}}^{QN}\right]^T\,(K_{\mathrm{f}}^{PQ})^{-1}\,\left[S_{\mathrm{fs}}^{PM}\right]\,\frac{1}{\rho_f}\,\mathbf{n}^-\cdot\frac{\partial p_I^M}{\partial \mathbf{x}} + P^N.
\end{aligned}
\tag{6.3.1–9}
$$

Defining

$$M_{\mathrm{VMA}}^{NM} \equiv \left[S_{\mathrm{fs}}^{QN}\right]^T \left(K_{\mathrm{f}}^{PQ}\right)^{-1} \left[S_{\mathrm{fs}}^{PM}\right], \tag{6.3.1–10}$$

the equation for the structure can be written

$$\left(M^{NM} + M_{\mathrm{VMA}}^{NM}\right) \ddot{u}^M + C_{(m)}^{NM} \dot{u}^M + I^N = -\left[S_{\mathrm{fs}}^{QN}\right]^T p_I - \frac{1}{\rho_f} M_{\mathrm{VMA}}^{NM}\, \mathbf{n}^- \cdot \frac{\partial p_I^M}{\partial \mathbf{x}} + P^N. \tag{6.3.1–11}$$

This equation makes the term "virtual mass" clear: the effect of an incompressible fluid surrounding a structure is to add inertia to the structural motion, M_{VMA}^{NM}. For some shapes—in particular, cylinders, disks, and spheres—the virtual mass is *a priori* known from classical solutions.

The matrix operator S_{fs}^{QN} is the discrete form of the integral over the wetted or loaded surface (see "Coupled acoustic-structural medium analysis," Section 2.9.1). If the structure is long and thin, it is often modeled with beam elements. Then it is appropriate to replace the surface integral that generates the $\left[S_{\mathrm{fs}}^{QN}\right]^T p_I$ term with a more convenient volume integral term. Since

$$[S_{\mathrm{fs}}]^T p_I = \int_{S_{\mathrm{fs}}} H^N p_I\, \mathbf{n}^- dS, \tag{6.3.1–12}$$

where H^N is a suitable interpolation function, Green's theorem can be applied to yield

$$[S_{\mathrm{fs}}]^T p_I = -\int_{V_{\mathrm{wetted}}} H^N \frac{\partial p_I}{\partial \mathbf{x}}\, dS, \tag{6.3.1–13}$$

where V_{wetted} is the volume enclosed by the wetted surface. If H^N is chosen to be the same interpolation function used for the beam and if the loading is similarly interpolated, this term becomes

$$\begin{aligned} [S_{\mathrm{fs}}]^T p_I &= -\int_{V_{\mathrm{solid}}} H^N H^M\, dS\, \frac{\partial p_I^M}{\partial x} \\ &= -\frac{1}{\rho_s} M^{NM} \frac{\partial p_I^M}{\partial x}. \end{aligned} \tag{6.3.1–14}$$

Therefore, the virtual mass equations for a beam immersed in an incompressible fluid are

$$\left(M^{NM} + M_{\mathrm{VMA}}^{NM}\right) \ddot{u}^M + C_{(m)}^{NM} \dot{u}^M + I^N = -\left(\frac{1}{\rho_f} M_{\mathrm{VMA}}^{NM} + \frac{1}{\rho_s} M^{NM}\right) \frac{\partial p_I^M}{\partial x} + P^N. \tag{6.3.1–15}$$

This equation states that an immersed beam is loaded by a term due to the pressure gradient in the fluid due to the incident wave and that it experiences an additional amount of inertia due to the presence of the fluid.

Scattering load amplitude

The incident field in the fluid is assumed known; moreover, it is assumed to be associated with a separable solution to the scalar wave equation of the form

$$p_I(\mathbf{x}, \tau) \equiv p_t(\tau) p_x(\mathbf{x}),$$

where τ is the "retarded time," which is a function of the true time, t, and the position, $\mathbf{x}$. In applications of interest here, the incident wave is produced by an excitation outside the computational domain. However, its temporal part, $p_t(t)$, is specified through an amplitude time history at a single point inside the computational domain, referred to as the "standoff point," $\mathbf{x_0}$. The spatial term, $p_x(\mathbf{x})$, is restricted to forms modeling plane and spherical wavefronts only, emanating from a specified "source" point, $\mathbf{x_s}$.

An incident wave produces a time-varying pressure at a spatial point of interest $\mathbf{x}_j$ of the form

$$p_I(\mathbf{x}_j, t) \equiv p_t(t) p_x(\mathbf{x}_j). \tag{6.3.1–16}$$

The spatial variation of the field at point $\mathbf{x}_j$ is expressed by either

$$p_x(\mathbf{x}_j) = \left[\frac{\|\mathbf{x_s} - \mathbf{x_0}\|}{\|\mathbf{x_s} - \mathbf{x}_j\|} \right]^{\alpha}$$

for spherically symmetric waves or simply

$$p_x(\mathbf{x}_j) = 1$$

for plane waves. The amplitude observed at a point $\mathbf{x}_j$ is related to that at the standoff point through this spatial variation, delayed by the time required for the wave to travel from the standoff point to the point $\mathbf{x}_j$:

$$\begin{aligned} p_I(\mathbf{x}_j, t) &= p_t\left(t - \frac{R_j - R_0}{c_0}\right) p_x(\mathbf{x}_j) \\ &\equiv p_t(\tau_j) p_x(\mathbf{x}_j), \end{aligned} \tag{6.3.1–17}$$

where c_0 is the wave speed in the fluid. We have defined

$$R_0 \equiv \|\mathbf{x_s} - \mathbf{x_0}\|,$$

and

$$R_j \equiv \|\mathbf{x_s} - \mathbf{x}_j\|$$

for spherical waves and

$$R_j \equiv \frac{|(\mathbf{x}_j - \mathbf{x_s}) \cdot (\mathbf{x_0} - \mathbf{x_s})|}{\|\mathbf{x_s} - \mathbf{x_0}\|}$$

for plane waves. τ_j is referred to as "retarded time" because it includes a shift corresponding to the time required for the wave to move from the standoff point to $\mathbf{x}_j$. Of course, the delay may be positive or negative, depending on the relative distances between the point of interest, $\mathbf{x}_j$, and the source and between the standoff point and the source.

The incident wave boundary tractions in the solid equation Equation 6.3.1–4 result from direct substitution of Equation 6.3.1–17. The fluid boundary tractions in Equation 6.3.1–4 and the tractions in Equation 6.3.1–15 require the evaluation of the gradient of the incident wave pressure, assuming a source point with negligible speed:

$$\frac{\partial p_I(\mathbf{x}_j, t)}{\partial \mathbf{x}} = \left[-p_x \dot{p}_t \frac{1}{c_0} + p_t \frac{\partial p_x}{\partial R_j} \right] \frac{\partial R_j}{\partial \mathbf{x}}.$$

First consider the case of generalized spatial decay, spherically symmetric waves with an exponent varying with the distance:

$$p_x = \left(\frac{R_0}{R_j} \right)^{\alpha(R_j, R_0)},$$

where the exponent is chosen to be

$$\alpha(R_j, R_0) \equiv \frac{(A+1)R_j}{CR_0 + (B+1)R_j}.$$

This function reverts to acoustic decay for spherical waves if the constants A, B, and C are zero and approaches plane wave propagation as $A \to -1$; however, it allows some degree of flexibility in adapting the decay to experimental data. The gradient of this term is

$$\frac{\partial p_x}{\partial R_j} = -\left(\frac{R_0}{R_j} \right)^{\alpha} \left(\frac{\alpha}{R_j} \right) \left[1 - \log\left(\frac{R_0}{R_j} \right) \frac{CR_0}{CR_0 + (B+1)R_j} \right].$$

The gradient of the time-varying incident pressure simplifies for spherical acoustic waves to

$$\frac{\partial p_I(\mathbf{x}_j, t)}{\partial \mathbf{x}} = \left[\frac{-1}{c_0} \dot{p}_t - \frac{1}{R_j} p_t \right] \frac{R_0}{R_j^2} (\mathbf{x}_j - \mathbf{x}_s)$$

and to

$$\frac{\partial p_I(\mathbf{x}_j, t)}{\partial \mathbf{x}} = \left[\frac{-1}{c_0} \dot{p}_t \right] \frac{1}{R_0} (\mathbf{x_0} - \mathbf{x}_s)$$

for plane waves. The direction of the wave travel is

$$\frac{1}{R_j}\left(\mathbf{x}_j - \mathbf{x}_s\right)$$

for spherical waves and equal to the constant vector

$$\frac{1}{R_0}\left(\mathbf{x_0} - \mathbf{x}_s\right)$$

for plane waves. The time derivatives of the incident pressure field can be written

$$\dot{p}_I(\mathbf{x}_j, t) = \dot{p}_t p_x$$

and

$$\ddot{p}_I(\mathbf{x}_j, t) = \ddot{p}_t p_x,$$

again neglecting the small terms associated with source point motion.

Steady-state dynamics

Development of incident wave loading in steady-state dynamics is as for the transient case, with a few exceptions. First, no source point motion is included in the steady-state formulation. Second, only spherical and planar sources are supported. The Fourier transform is applied to the excitation: with the usual assumptions on $p_t(\tau_j)$, this is formally equivalent to substitution of a complex exponential in the steady-state formulation:

$$p_t(\tau_j) \mapsto \tilde{p_t} e^{i\Omega\tau_j},$$

where $\tilde{p_t}$ is the complex magnitude of the incident pressure at the standoff point, as a function of frequency. Then

$$p_I(\mathbf{x}_j, t) \mapsto \tilde{p_t} e^{i\Omega\tau_j} e^{-i\Omega\left[\frac{R_j - R_0}{c_0}\right]} p_x(\mathbf{x}_j);$$

and the gradient becomes

$$\frac{\partial p_I(\mathbf{x}_j, t)}{\partial \mathbf{x}} \mapsto \left[\frac{-i\Omega}{c_0} - \frac{1}{R_j}\right] \tilde{p_t} e^{-i\Omega\left[\frac{R_j - R_0}{c_0}\right]} \frac{R_0}{R_j^2}\left(\mathbf{x_s} - \mathbf{x}_j\right)$$

for spherical waves and

$$\frac{\partial p_I(\mathbf{x}_j, t)}{\partial \mathbf{x}} \mapsto \left[\frac{-i\Omega}{c_0}\right] \tilde{p_t} \frac{1}{R_0}\left(\mathbf{x_0} - \mathbf{x}_s\right)$$

for plane waves. The time derivatives of incident pressure, needed for the computation of the fluid traction on impedance boundaries, are

$$\dot{p}_t(t) \mapsto i\Omega\tilde{p}_t e^{i\Omega\tau_j} e^{-i\Omega\left[\frac{R_j - R_0}{c_0}\right]} p_x(\mathbf{x}_j),$$

and

$$\ddot{p}_t(t) \mapsto -\Omega^2\tilde{p}_t e^{i\Omega\tau_j} e^{-i\Omega\left[\frac{R_j - R_0}{c_0}\right]} p_x(\mathbf{x}_j).$$

The value of the load applied at a point $\mathbf{x}_j$ depends on the specified value $\tilde{p}_t$ and the spatial variation but includes a phase shift relative to the standoff point, $e^{-i\Omega\left[\frac{R_j - R_0}{c_0}\right]}$, corresponding to the time shifts in transient analysis.

Effect of structural motion

Generally, the effects of the motion of the source point, $\mathbf{x_s}(t)$, are included in the defined amplitude $p_t(t)$ of the incident wave field (specified at the standoff point $\mathbf{x_0}$). That is, it is assumed that the stated amplitude reflects the time history at a specific fixed standoff point, including the effects of the source point motion, if any. For example, in underwater shock loading the incident wave field is produced by a pulsating gas bubble, which migrates toward the free surface of the water. Therefore, the corresponding incident wave amplitude model for this effect includes a moving source point.

The structural model affected by the incident wave may be in motion, as well. For example, a ship can move with respect to an explosive charge source during the loading. Because the wave motion is usually extremely fast compared to the ship motion and because incident wave pulse durations are typically short, some approximations apply. First, it can be assumed that the motion of the standoff point, $\mathbf{x_0}$, can be described by its position at time zero and by the average velocity during the incident wave loading, $\mathbf{v_0}$. In addition, the magnitude of this velocity can be assumed small compared to the speed of wave propagation, so the wave equation itself is unaffected by the relative motion. Furthermore, it can be assumed that the specified amplitude history, $p_t(t)$, corresponds to observations made at the position of the standoff point at time zero, $\mathbf{x_0}(0)$.

Under these assumptions the effect of structural motion during incident wave loading can be modeled in a reference frame fixed with the standoff point, so the source point appears to move with $\mathbf{v_0}$:

$$\mathbf{x_s}(t) \mapsto \mathbf{x_s}(0) - \mathbf{v_0}t.$$

This motion also affects the perceived arrival time of the wave at the standoff point. Defining the time of arrival at $\mathbf{x_0}(0)$ as t and the time of arrival at $\mathbf{x_0}(t)$ as t_I, we can express the difference between these times in terms of $\mathbf{v_0}$:

$$c_0|t_I - t| = \|\mathbf{x_s}(t) - \mathbf{v_0}t - \mathbf{x_0}(0)\| - \|\mathbf{x_s}(t) - \mathbf{x_0}(0)\|.$$

This time shift is maximum when the movement $\mathbf{v_0}$ is aligned with $\mathbf{x_s}(t) - \mathbf{x_0}(0)$. Therefore,

$$c_0|t_I - t| \le \|\mathbf{x_s}(t) - \frac{t\|\mathbf{v_0}\|\,(\mathbf{x_s}(t) - \mathbf{x_0}(0))}{\|\mathbf{x_s}(t) - \mathbf{x_0}(0)\|} - \mathbf{x_0}(0)\| - \|\mathbf{x_s}(t) - \mathbf{x_0}(0)\|$$
$$\le |t|\|\mathbf{v_0}\|.$$

Since we have assumed that the standoff motion is slow compared to the wave speed, we neglect this small time shift.

Underwater explosion—bubble load amplitude

An underwater explosion can lead to the formation of a highly compressed bubble that propels the surrounding water radially outward, generating an outward-propagating shockwave. As the bubble expands, the pressure inside decreases until it is considerably below the ambient pressure. After reaching a maximum radius with a minimum pressure, the bubble begins to contract. The contraction proceeds until the bubble collapses to a minimum radius. Because of a large pressure generated inside the bubble during this stage, the bubble begins to expand again, generating a second outward-propagating wave. Once the bubble expands to a second maximum radius, it contracts again. The same expansion-contraction sequence can repeat many times. At the same time during this pulsation process, the bubble migrates upward under the force of buoyancy. As long as the bubble does not reach the free water surface, the expansion-contraction sequence continues, each time with reduced amplitude.

Geers-Hunter model

Geers and Hunter (2002) proposed a phenomenological model that treats an underwater explosion as a single bubble event comprised of a shockwave phase and an oscillation phase, with the first phase providing initial conditions to the second. This model was slightly refined in a later paper, Geers and Park (2005).

Based on this model, the volume acceleration during the shockwave phase is given by

$$\ddot{V}(t) = \frac{4\pi a_c}{\rho_f} P_c[0.8251\exp(-1.338t/T_c) + 0.1749\exp(-0.1805t/T_c)], \tag{6.3.1–18}$$

in which $P_c = K(m_c^{1/3}/a_c)^{1+A}$ and $T_c = km_c^{1/3}(a_c/m_c^{1/3})^B$, where m_c and a_c are the mass and radius of the explosive charge, respectively, and $\boldsymbol{K}$, $\boldsymbol{k}$, $\boldsymbol{A}$, and $\boldsymbol{B}$ are constants for the charge material; ρ_f is the mass density of the fluid.

Integration with $\dot{V}(0) = 0$ and further integration with $V(0) = (4/3)\pi a_c^3$ then yields

$$\dot{V}(t) = \frac{4\pi a_c}{\rho_f} P_c[1.5857 - 0.6167\exp(-1.338t/T_c) - 0.9690\exp(-0.1805t/T_c)]T_c,$$
$$V(t) = \frac{4}{3}\pi a_c^3 + \frac{4\pi a_c}{\rho_f} P_c[1.5857t/T_c - 5.8293 + 0.4609\exp(-1.338t/T_c) +$$
$$+ 5.3684\exp(-0.1805t/T_c)]T_c^2,$$

and radial displacement and velocity follow as

$$a = \left(\frac{3}{4\pi}V\right)^{1/3} \quad \text{and} \quad \dot{a} = \frac{1}{4\pi}\dot{V}/a^2.$$

These expressions are evaluated at $t_I = 7T_c$ to determine the initial conditions for the subsequent bubble response calculations during the oscillation phase. This choice was validated since, for a single set of charge constants, the initial condition values for values of t_I between $3T_c$ and $7T_c$ produce essentially the same response during the oscillation phase, as demonstrated by Geers and Hunter (2002).

The following are the equations of motion for the doubly asymptotic approximation model to describe the evolution of the bubble radius, a, and migration, u, during the oscillation phase:

$$\begin{aligned}
\dot{a} &= -\frac{\phi_{l0}}{a} - \frac{1}{c_f}\left(\dot{\phi}_{l0} - \dot{a}^2 - \frac{1}{3}\dot{u}^2 - \frac{2}{3}\dot{u}\frac{\phi_{l1}}{a}\right),\\
\dot{u} &= -2\frac{\phi_{l1}}{a} - \frac{1}{c_f}\left(\dot{\phi}_{l1} - 2\dot{a}\dot{u}\right),\\
\dot{\phi}_{l0} &= \frac{1}{(1+\zeta)}\left[\left(\frac{1}{2} + \frac{1}{2}\frac{\rho_g}{\rho_f} + \zeta\right)\left(\dot{a}^2 + \frac{1}{3}\dot{u}^2\right) - \frac{\rho_g}{\rho_f}c_g\frac{\phi_{l0}}{a} + \frac{2}{3}(1+\zeta)\dot{u}\frac{\phi_{l1}}{a} - Z\right],\\
\dot{\phi}_{l1} &= \frac{1}{(1+\zeta)}\left[\left(1 + \frac{\rho_g}{\rho_f} + 2\zeta\right)\dot{a}\dot{u} - \left(1 - \frac{\rho_g}{\rho_f}\right)ga - \frac{\rho_g}{\rho_f}c_g\left(2\frac{\phi_{l1}}{a} + \frac{\phi_{g1}}{a}\right) + \frac{3}{8}C_D|\dot{u}|^{E_D}\right],\\
\dot{\phi}_{g1} &= \frac{1}{(1+\zeta)}\left[\left(2 + \frac{c_g}{c_f} + \zeta\right)\dot{a}\dot{u} + \frac{c_g}{c_f}\left(1 - \frac{\rho_g}{\rho_f}\right)ga - c_g\left(2\frac{\phi_{l1}}{a} + \frac{\phi_{g1}}{a}\right) - \frac{c_g}{c_f}\frac{3}{8}C_D|\dot{u}|^{E_D}\right],
\end{aligned} \tag{6.3.1–19}$$

where

$$\zeta = \frac{\rho_g c_g}{\rho_f c_f}, \quad \text{and} \quad Z = \frac{1}{\rho_f}(P_g - p_I + \rho_f g u) + \frac{1}{3}\left[\left(\frac{\phi_{l1}}{a}\right)^2 - \frac{\rho_g}{\rho_f}\left(\frac{\phi_{g1}}{a}\right)^2\right];$$

$$P_g = K_c\left(\frac{V_c}{V}\right)^{\gamma}, \quad \rho_g = \rho_c\left(\frac{V_c}{V}\right), \quad c_g = c_c\left(\frac{V_c}{V}\right)^{\frac{1}{2}(\gamma-1)}, \quad \text{and} \quad c_c = \sqrt{\frac{\gamma K_c}{\rho_c}};$$

and ρ_c is the charge mass density, c_f is the sound speed in fluid, $V = (4/3)\pi a^3$ is the current volume of the bubble, K_c is the adiabatic charge constant, V_c is the volume of charge, γ is the ratio of specific heats for gas, $\boldsymbol{g}$ is the acceleration due to gravity, and $p_I = p_{atm} + \rho_f g d_I$ (where p_{atm} as the atmospheric pressure and d_I as the depth of the charge's center). C_D is an empirical flow drag parameter, which impedes the bubble's migration, and E_D is an exponent that can be tuned to match experimental migration data [Geers and Park (2005)].

Seven initial conditions are needed. The first two are $a(t_I) = a_I$, $\dot{a}(t_I) = \dot{a}_I$, the second two are $u(t_I) = 0$, $\dot{u}(t_I) = 0$, the fifth one is $\phi_{l1}(t_I) = 0$, and the remaining two are determined as

$$\phi_{l0}(t_I) = -a_I\dot{a}_I\left[1 - \frac{1}{2}\left(1 - \frac{\rho_{gI}}{\rho_f}\right)\frac{\dot{a}_I}{c_f} + \zeta_I\right] + \frac{a_I}{c_f}Z_I$$

and

$$\phi_{g1}(t_I) = -\zeta_I^{-1}\left(1 - \frac{\rho_{gI}}{\rho_f}\right)\frac{g}{c_f}a_I^2$$

with

$$Z_I = \rho_f^{-1}(P_{gI} - p_I) - \frac{1}{3}\left(1 - \frac{\rho_{gI}}{\rho_f}\right)^2 \frac{\rho_f}{\rho_{gI}}\left(\frac{ga_I}{c_{gI}}\right)^2.$$

Then Equation 6.3.1–19 can be solved by using any suitable method for nonlinear ordinary differential equations; in Abaqus, a fourth-order Runge-Kutta integrator with variable time steps is used.

The incident pressure induced during the bubble response can be expressed as

$$p_I(\mathbf{x}_j, t) \equiv p_t(t)p_x(\mathbf{x}_j), \tag{6.3.1–20}$$

where

$$p_x(\mathbf{x}_j) = \frac{1}{R_j}$$

and

$$p_t(t) = \frac{\rho_f}{4\pi}\left(\frac{a_c}{R_j}\right)^A \ddot{V}((a_c/R_j)^B t), \tag{6.3.1–21}$$

with $\ddot{V}$ given by Equation 6.3.1–18 for the shockwave phase ($t_I < 7T_c$), and

$$p_t(t) = \frac{\rho_f}{4\pi}\ddot{V}(t) = \rho_f(a^2\ddot{a} + 2a\dot{a}^2)$$

for the oscillation phase ($t_I \geq 7T_c$). Here, $R_j \equiv \|\mathbf{x_s} - \mathbf{x}_j\|$.

Since the constants $\boldsymbol{A}$ and $\boldsymbol{B}$ are both substantially smaller than one, $p_t(t)$ for the shockwave phase above is only weakly dependent on R_j. Thus, for simplicity, R_j can be treated as a constant when the gradient of the pressure is evaluated. In this way all formulations in the previous two sections can still be used without losing any significant accuracy.

"Waveless" model

The model of Geers and Hunter (2002) can be simplified to ignore the energy losses due to waves in the fluid and the gases. This "waveless" form of the equations more closely reproduces the results of earlier research on this phenomenon.

In the waveless model the shockwave phase of the loading is unchanged from the previous case. The equations of motion for the evolution of the bubble radius and migration during the oscillation phase are simplified, however, using the assumptions

$\frac{1}{c_f} \mapsto 0,$

$\frac{\rho_g}{\rho_f} \mapsto 0,$

and

$c_g \mapsto 0.$

Under these assumptions, the waveless equations of motion for the bubble dynamics are

$$\begin{aligned}
\dot{a} &= -\frac{\phi_{l0}}{a}, \\
\dot{u} &= -2\frac{\phi_{l1}}{a}, \\
\dot{\phi}_{l0} &= \frac{1}{2}\left(\dot{a}^2 + \frac{1}{3}\dot{u}^2\right) + \frac{2}{3}\dot{u}\frac{\phi_{l1}}{a} - Z, \\
\dot{\phi}_{l1} &= \dot{a}\dot{u} - ga + \frac{3}{8}C_D|\dot{u}|^{E_D}, \\
\dot{\phi}_{g1} &= 2\dot{a}\dot{u},
\end{aligned} \tag{6.3.1–22}$$

where

$$Z = \frac{1}{\rho_f}\left(P_g - p_I + \rho_f g u\right) + \frac{1}{3}\left(\frac{\phi_{l1}}{a}\right)^2;$$

$$P_g = K_c\left(\frac{V_c}{V}\right)^{\gamma}, \quad \rho_g = 0, \quad c_g = 0.$$

The initial conditions are determined as for the previously discussed case. Since the equation for ϕ_{g1} is now decoupled from the others, it does not need to be solved.

Reflections outside the computational domain

Abaqus allows the user to specify planes outside the computational domain that reflect the incident wave. This reflected wave is superposed, with a suitable time delay or phase shift, onto the wave arriving at the standoff point via the direct path, forming the total incident wave field for that source. This functionality is implemented for spherical or planar waves and reflection planes at any orientation, which may have arbitrary impedance properties. Only a single reflection from each plane is considered. However, multiple reflections inside the finite element computational domain are considered automatically.

Spherical waves

Consider the schematic arrangement of source point, standoff point, and reflection plane $\boldsymbol{A}$ as shown in Figure 6.3.1–1. The reflection plane can be specified by a normal vector, $\mathbf{n}_A$, and some point on the plane, $\mathbf{x}_A$. The distance from the source to the plane is $d_A \equiv \|(\mathbf{x}_A - \mathbf{x}_S) \cdot \mathbf{n}_A\|$, and the plane is assumed to have a specific characteristic impedance, Z_A. This information, together with the behavior of the source, is sufficient to characterize the reflected field in terms of an image source, located at the point $\mathbf{x}_{SA} \equiv \mathbf{x}_S + 2d_A\mathbf{n}_A$. The source will be derived in steady state and then extended to the transient case.

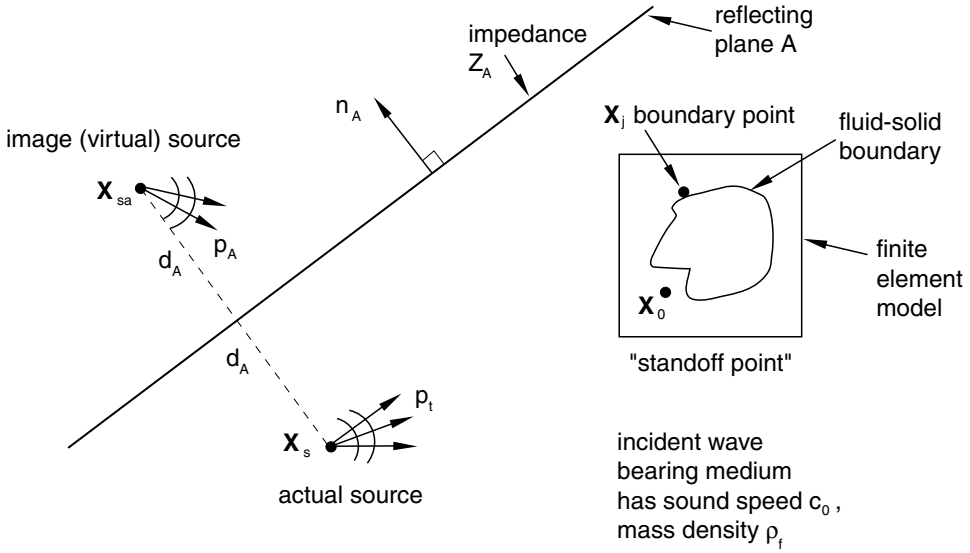


Figure 6.3.1–1 Schematic of incident wave loading.

The actual source is given by

$$P_S \equiv \tilde{p}_t e^{-i\Omega\left[\frac{\|\mathbf{x}_S-\mathbf{x}\|}{c_0}\right]} \frac{\|\mathbf{x}_S-\mathbf{x_0}\|}{\|\mathbf{x}_S-\mathbf{x}\|},$$

and the image source is given by

$$P_{SA} \equiv \tilde{p}_A e^{-i\Omega\left[\frac{\|\mathbf{x}_{SA}-\mathbf{x}\|}{c_0}\right]} \frac{\|\mathbf{x}_S-\mathbf{x_0}\|}{\|\mathbf{x}_{SA}-\mathbf{x}\|},$$

where the normalization length $\|\mathbf{x}_S - \mathbf{x_0}\|$ is chosen for convenience. Generally, a nonzero imaginary part of the plane's impedance Z_A will imply that $\tilde{p}_A$ will also have a nonzero imaginary part. The unknown $\tilde{p}_A$ is found by enforcing the impedance condition at the plane:

$$P_{SA} + P_S = i\Omega Z_A \left(\tilde{\mathbf{u}}_f \cdot \mathbf{n}_A\right),$$

and the conservation of linear momentum:

$$-\Omega^2 \tilde{\rho}\tilde{\mathbf{u}}_f = \frac{\partial}{\partial \mathbf{x}} \left[P_{SA} + P_S\right],$$

where $\tilde{\rho}$ is the complex density of the medium, including volumetric drag effects. Combining these two equations and simplifying, we have

$$\tilde{p}_A = \tilde{p}_t \left[\frac{1 - Z_A \left(\frac{1}{c_0 \tilde{\rho}} + \frac{1}{i\Omega \tilde{\rho} d_A} \right)}{-1 - Z_A \left(\frac{1}{c_0 \tilde{\rho}} + \frac{1}{i\Omega \tilde{\rho} d_A} \right)} \right].$$

In Abaqus the impedance property is specified using admittance constants $1/c_1$ and $1/k_1$,

$$\frac{1}{Z_A} = \frac{1}{c_1} + \frac{i\Omega}{k_1},$$

so it is convenient to use

$$\tilde{p}_A \equiv \left[\frac{Q_+}{Q_-} \right] \tilde{p}_t = \tilde{p}_t \left[\frac{\frac{1}{c_1} + \frac{i\Omega}{k_1} - \left(\frac{1}{c_0 \tilde{\rho}} + \frac{1}{i\Omega \tilde{\rho} d_A} \right)}{-\frac{1}{c_1} - \frac{i\Omega}{k_1} - \left(\frac{1}{c_0 \tilde{\rho}} + \frac{1}{i\Omega \tilde{\rho} d_A} \right)} \right].$$

This describes the source strength for a spherical image source in steady state, but in the transient case the complex density needs to be eliminated using

$\tilde{\rho} = \rho_f + \frac{\gamma}{i\Omega}$.

The expression for the image source can be expanded to

$$\left[-\frac{\gamma}{c_1} - \frac{1}{d_A} - i\Omega \frac{\rho_f}{c_1} - i\Omega \frac{\gamma}{k_1} - i\Omega \frac{1}{c_0} + \Omega^2 \frac{\rho_f}{k_1} \right] \tilde{p}_A$$

$$= \left[\frac{\gamma}{c_1} - \frac{1}{d_A} + i\Omega \frac{\rho_f}{c_1} + i\Omega \frac{\gamma}{k_1} - i\Omega \frac{1}{c_0} - \Omega^2 \frac{\rho_f}{k_1} \right] \tilde{p}_t.$$

In Abaqus the effects of fluid volumetric drag, the spreading loss due to the distance to the reflection plane, and the complex part of the admittance $1/k_1$ are ignored, so the amplitude of the reflected wave is related to the amplitude of the incident wave by

$$\tilde{p}_A \equiv \tilde{p}_t \left[\frac{\frac{1}{c_1} - \frac{1}{c_0 \rho_f}}{-\frac{1}{c_1} - \frac{1}{c_0 \rho_f}} \right] \equiv Q \tilde{p}_t.$$

Therefore, the reflected spherical load is similar to the direct load, with magnitude reduced by the reflection impedance effect and by the greater distance travelled. In the time domain inversion of the steady-state result, retaining a phase shift corresponding to the arrival time at the standoff point, yields

$$p_A(t, \mathbf{x}_j) \equiv Q p_t(\tau_j) \frac{\|\mathbf{x}_S - \mathbf{x_0}\|}{\|\mathbf{x}_{SA} - \mathbf{x}\|},$$

where

$$\tau_j \equiv t - \frac{\|\mathbf{x}_{SA} - \mathbf{x}_j\|}{c_0} + \frac{\|\mathbf{x}_S - \mathbf{x}_0\|}{c_0}.$$

The direction of this wave varies at each field point $\mathbf{x}_j$:

$$\mathbf{n}_{sa} \equiv \frac{\mathbf{x}_{SA} - \mathbf{x}_j}{\|\mathbf{x}_{SA} - \mathbf{x}_j\|}.$$

"Soft" and "hard" limits

If a reflecting plane is considered "soft," the pressure is zero there. Then $Z_A \equiv 0$ and

$$p_A = -p_t.$$

A "hard" reflecting plane refers to a zero fluid particle motion condition, implying $1/Z_A \equiv 0$. Then

$$p_A = p_t.$$

Planar waves

The schematic for reflected plane waves is similar to that for spherical waves, except for the geometry of the reflection. In contrast to the spherical case, when planar waves reflect from a boundary (see Figure 6.3.1–2), the direction of the reflected wave is common for all points on the wavefront. The location of the perceived source of the reflected wave is different, however. To calculate the phase shift for the reflected wave incident at an arbitrary point $\mathbf{x_j}$ correctly, the locations of these perceived sources need to be calculated.

The reflected plane wave load can be calculated on the basis of the reflected wave direction and the time delay associated with the additional path length of the reflected wave. Referring to Figure 6.3.1–2, we see that the reflected wave direction is given by

$$\mathbf{n}_v = \mathbf{n}_0 - 2\left(\mathbf{n}_0 \cdot \mathbf{n}_{sa}\right)\mathbf{n}_{sa}.$$

The direct path wave travels the distance

$$\|\hat{\mathbf{x}}_S - \mathbf{x}_S\| + \|\hat{\mathbf{x}}_S - \mathbf{x}_0\| + \left|(\mathbf{x}_j - \mathbf{x}_0) \cdot \mathbf{n}_0\right|,$$

and the reflected path travels the distance

$$\|\hat{\mathbf{x}}_S - \mathbf{x}_S\| + \|\hat{\mathbf{x}}_j - \mathbf{x}_v\|.$$

The time delay of interest for point $\mathbf{x}_j$, τ_j, is related to the difference of these distances via the acoustic wave speed:

$$c_0\tau_j = \left|(\mathbf{x}_j - \mathbf{x}_0) \cdot \mathbf{n}_0\right| + \|\hat{\mathbf{x}}_S - \mathbf{x}_0\| - \|\hat{\mathbf{x}}_j - \mathbf{x}_v\|.$$

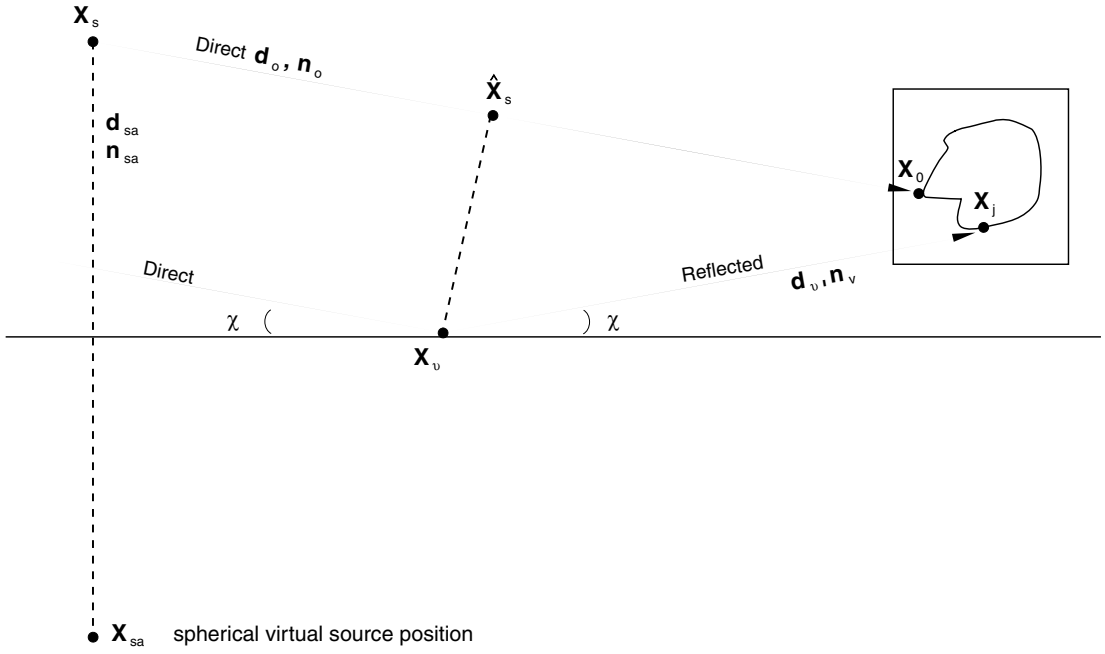


Figure 6.3.1–2 Schematic of incident wave loading using planar waves.

The first term on the right is the same as the delay due to the direct path; the rest is the increment due to the additional path length of the reflection. Some geometric manipulations allow this expression to be written in terms of the locations of $\mathbf{x}_j$, $\mathbf{x}_S$, $\mathbf{x}_0$, and the location of the spherical-wave virtual source, $\mathbf{x}_{sa}$:

$$c_0\tau_j = |(\mathbf{x}_j - \mathbf{x}_0)\cdot\mathbf{n}_0| + 2r_v\left(\frac{\mathbf{d}_{sa}\cdot\mathbf{d}_0}{\|\mathbf{d}_{sa}\|r_0}\right)^2,$$

where

$$r_0 = \|\mathbf{x}_S - \mathbf{x}_0\|,$$

and

$$r_v = \frac{\|\mathbf{d}_{sa}\|r_0}{|\mathbf{n}_0\cdot\mathbf{n}_{sa}|}\left(\frac{1}{2} - \frac{(\mathbf{x}_j - \hat{\mathbf{x}}_S)\cdot\mathbf{d}_{sa}}{\|\mathbf{d}_{sa}\|^2}\right).$$

Reflected planar waves are subject to reduction in amplitude due to impedance conditions at the reflection plane in the same manner as spherical waves, but there is no spatial attenuation. Unlike spherical waves, planar waves may yield no reflections for some orientations of the specified reflection plane: if the

direction of the wave is parallel to the reflection plane, or strikes at an obtuse angle, no reflections are generated.

Reference

- "Acoustic and shock loads," Section 34.4.6 of the Abaqus Analysis User's Guide

6.4 Pressure penetration loading

- “Pressure penetration loading with surface-based contact,” Section 6.4.1

6.4.1 PRESSURE PENETRATION LOADING WITH SURFACE-BASED CONTACT

Product: Abaqus/Standard

Abaqus/Standard allows for the simulation of fluid penetrating into the surface between two contacting bodies and application of the fluid pressure normal to the surfaces. The surface-based contact approach is used to model the interactions between the bodies, where one surface definition provides the "master" surface and the other surface definition provides the "slave" surface. Both surfaces can be deformable, or one can be rigid.

A single slave node–based penetration criterion is used. Fluid will penetrate into the surface between the contacting bodies from one or multiple locations, which are exposed to the fluid, until a point is reached where the contact pressure is greater than the critical value specified by the user, cutting off further penetration of the fluid. The critical contact pressure is introduced to account for the asperities on the contacting surfaces. The higher this value, the easier the fluid penetrates. The default value of the critical contact pressure is zero, in which case fluid penetration occurs only if the contact pressure is zero and contact is lost. When a node has a contact status of "OPEN," its nearest neighboring nodes are considered to be subjected to the fluid pressure as well. The nodes initially exposed to the fluid, which are specified by the user, will always be subjected to the fluid pressure irrespective of the contact status at these nodes.

The pressure penetration load will be applied normal to the element surface based on the pressure penetration criterion described above at the beginning of an increment and will remain constant over that increment even if the fluid penetrates further during that increment. In two dimensions, a nodal integration scheme is used to integrate the distributed pressure penetration load over an element; the variation of the distributed load over an element will be determined by the load magnitudes at the element's nodes, which are coincident with the base points. Consider the contact interaction of three nodes—101, 102, and 103—on the slave surface made up of faces of two first-order elements, 1 and 2, with a master surface made up of faces of two elements, 4 and 5, which are described by nodes 201, 202, and 203 as shown in Figure 6.4.1–1.

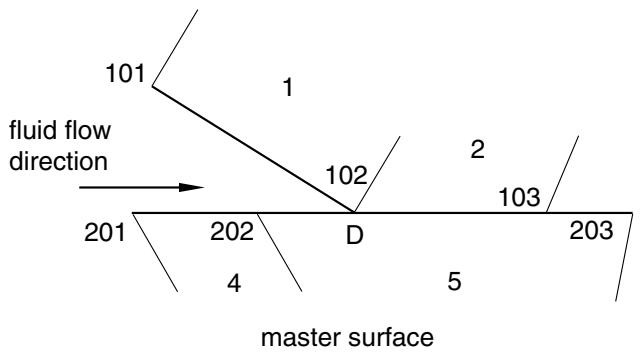


Figure 6.4.1–1 Pressure penetration with nonmatching meshes.

If the fluid with a pressure magnitude of f has penetrated up to node 102 on the slave surface, the variation of the distributed load over element 1 is given by

$$P_1 = fN_1 + fN_2 = f,$$

and the variation of the distributed load over element 2 is given by

$$P_2 = fN_1,$$

where N_1 and N_2 are the shape functions on the face of a first-order element.

For an element-based master surface we must also consider how the fluid pressure is applied to the master surface, which depends on the location of the "anchor" point on the master surface. The "anchor" point is chosen as the point on the master surface closest to the last slave node subjected to the fluid pressure (pressure penetration tip). All the nodes between the "anchor point" and the node initially exposed to the fluid on the master surface, as specified by the user, are considered to be subjected to the fluid pressure.

The pressure penetration capability does not require that the contacting surfaces have matching meshes; however, the best accuracy is obtained when meshes are initially matching. For initially nonmatching meshes the equivalent fluid pressure applied on the master surface can be evaluated based on equilibrium considerations. For the problem illustrated in Figure 6.4.1–1, the "anchor" point corresponding to slave node 102 is D. For the element associated with the anchor point (element 5) on the master surface, part of the load is transferred from element 1 and part of the load is transferred from element 2 on the slave surface. The load distribution on element 5 is illustrated in Figure 6.4.1–2, where L_5 is the element length.

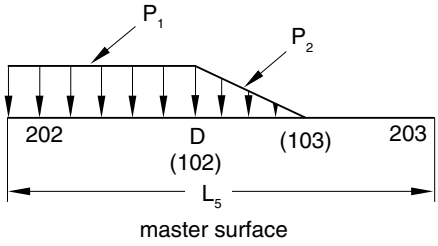


Figure 6.4.1–2 Stress distribution over an element on the master surface with nonmatching meshes.

Because the fluid pressure will be simulated as a distributed load on the contacting surfaces, the variation of the load over an element needs to be described. For a first-order element if Q_{202} and Q_{203} are the equivalent forces at nodes 202 and 203 due to the distributed load shown in Figure 6.4.1–2, the variation of the distributed load over element 5 on the master surface is given by

$$P_5 = \frac{4Q_{202} - 2Q_{203}}{L_5} N_1 + \frac{4Q_{203} - 2Q_{202}}{L_5} N_2.$$

A similar approach is used for the second-order elements.

Reference

- "Pressure penetration loading," Section 37.1.7 of the Abaqus Analysis User's Guide

6.5 Load stiffness

- “Pressure load stiffness,” Section 6.5.1
- “Load stiffness for beam elements,” Section 6.5.2
- “Pressure loadings on elbow elements,” Section 6.5.3

6.5.1 PRESSURE LOAD STIFFNESS

Product: Abaqus/Standard

In geometrically nonlinear analysis pressure loads are applied on the deformed structure. Hence, the equivalent nodal loads are dependent on the nodal displacements. This dependency leads to additional contributions to the Jacobian in the solution procedure used in Abaqus/Standard. The external virtual work is

$$\delta W^E = \int_A \delta \mathbf{u} \cdot p\, \mathbf{n}\, dA,$$

where A is the surface on which the pressure is applied; $\mathbf{n}$ is the normal to this surface, pointing into the material; $\delta \mathbf{u}$ is the virtual displacement field; and p is the pressure magnitude.

Pressure load stiffness on a surface in three-dimensional space

The expression $\mathbf{n}\, dA$ can be rewritten as follows:

$$\mathbf{n}\, dA = \frac{\partial \mathbf{x}}{\partial g} \times \frac{\partial \mathbf{x}}{\partial h}\, dg\, dh,$$

where $\mathbf{x}$ are the current coordinates of a point on the surface, and the surface parametric coordinates (g, h) are chosen to give the correct sign to $\mathbf{n}$ through the cross product. The external virtual work is then given by

$$\delta W^E = \int_{g,h} p\, \delta \mathbf{u} \cdot \frac{\partial \mathbf{x}}{\partial g} \times \frac{\partial \mathbf{x}}{\partial h}\, dg\, dh,$$

and the load stiffness matrix is obtained from

$$-d\delta W^E = -\int_{g,h} p\, \delta \mathbf{u} \cdot \left(\frac{\partial d\mathbf{u}}{\partial g} \times \frac{\partial \mathbf{x}}{\partial h} + \frac{\partial \mathbf{x}}{\partial g} \times \frac{\partial d\mathbf{u}}{\partial h} \right) dg\, dh,$$

where, for a solid, $d\mathbf{x} = d\mathbf{u}$.

Pressure load stiffness on a surface in two-dimensional space

Now

$$\mathbf{n}\, dA = \mathbf{k} \times \frac{\partial \mathbf{x}}{\partial g}\, t\, dg,$$

where $\mathbf{k} = (0, 0, 1)$ is a unit vector out of the plane of the model, t is the thickness of the two-dimensional solid (which is assumed to be constant), and the surface parametric coordinate g is chosen to give the correct sign to $\mathbf{n}$ through the cross product. The external work is then given by

$$\delta W^E = \int_g p\, \delta \mathbf{u} \cdot \mathbf{k} \times \frac{\partial \mathbf{x}}{\partial g}\, t\, dg,$$

and the load stiffness matrix is obtained from

$$-d\delta W^E = -\int_g p\, \delta \mathbf{u} \cdot \mathbf{k} \times \frac{\partial d\mathbf{u}}{\partial g}\, t\, dg.$$

Reference

- "Distributed loads," Section 34.4.3 of the Abaqus Analysis User's Guide

6.5.2 LOAD STIFFNESS FOR BEAM ELEMENTS

Product: Abaqus/Standard

Abaqus provides for loads per unit length in the beam cross-sectional directions as distributed load options for the beam elements (load types P1, P2). Since these are follower forces, they have a load stiffness; and this stiffness can sometimes be important especially in the case of buckling prediction by eigenvalue extraction. The symmetric form of this load stiffness is included in Abaqus/Standard (see Hibbitt, 1979, and Mang, 1980). This form is developed below. The external virtual work on the beam is

$$\delta \mathbf{W}^e = \int_S \mathbf{p} \cdot \delta \mathbf{u} \, dS,$$

where the pressure load, $\mathbf{p}$, is given by the externally prescribed pressure magnitude, p, as $\mathbf{p} = p\mathbf{n}_\alpha$, where $\alpha = 1$ or 2 defines the particular cross-sectional direction of the load. Therefore, $\mathbf{n}_\alpha = (-1)^\beta \mathbf{n}_\beta \times (d\mathbf{x}/dS)$, where $\beta = 2$ when $\alpha = 1$, and $\beta = 1$ when $\alpha = 2$ so that

$$\delta \mathbf{W}^e = (-1)^\beta \int_S p \left(\mathbf{n}_\beta \times \frac{d\mathbf{x}}{dS} \right) \cdot \delta \mathbf{u} \, dS,$$

where S is the material coordinate along the beam. Now assuming that the load magnitude, p, is externally prescribed so that it does not change with position, the rate of change of $\delta \mathbf{W}^e$ with change in position, $d\mathbf{u}$, is

$$d\delta \mathbf{W}^e = (-1)^\beta \int_S p \left(d\mathbf{n}_\beta \times \frac{d\mathbf{x}}{dS} + \mathbf{n}_\beta \times \frac{dd\mathbf{u}}{dS} \right) \cdot \delta \mathbf{u} \, dS.$$

Now

$$d\mathbf{n}_\beta = d\omega \times \mathbf{n}_\beta,$$

and so

$$d\mathbf{n}_\beta \times \frac{d\mathbf{x}}{dS} = (d\omega \times \mathbf{n}_\beta) \cdot \frac{d\mathbf{x}}{dS} = \left(d\omega \cdot \frac{d\mathbf{x}}{dS} \right) \mathbf{n}_\beta, \text{ neglecting } \mathbf{n}_\beta \cdot \frac{d\mathbf{x}}{dS}.$$

Thus,

$$d\delta \mathbf{W}^e = (-1)^\beta \int_S p \left[d\boldsymbol{\omega} \cdot \frac{d\mathbf{x}}{dS} \mathbf{n}_\beta \cdot \delta \mathbf{u} + \left(\mathbf{n}_\beta \times \frac{dd\mathbf{u}}{\mathrm{d}S} \right) \cdot \delta \mathbf{u} \right] dS.$$

This load stiffness is not symmetric, except in the case of a beam in a plane with fixed ends (or no ends, such as a ring), in which case the first term is exactly zero and the second gives the symmetric form

$$(-1)^\beta \int_S \frac{1}{2} p \mathbf{n}_\beta \cdot \left(\frac{dd\mathbf{u}}{\mathrm{d}S} \times \delta \mathbf{u} + \frac{d\mathbf{u}}{dS} \times d\delta \mathbf{u} \right) dS.$$

In Abaqus, even for the general beams in three dimensions, the load stiffness is introduced as the symmetric part of $d\delta\mathbf{W}^e$ above.

Reference

- "Distributed loads," Section 34.4.3 of the Abaqus Analysis User's Guide

6.5.3 PRESSURE LOADINGS ON ELBOW ELEMENTS

Product: Abaqus/Standard

Elbow elements are often used to model pipelines in which the curvature of the pipe can change significantly while the pipe is subjected to uniform or hydrostatic pressure. Therefore, pressure loadings that include large geometry changes are developed for these elements, as described in this section.

The virtual work contribution of pressure on the lateral surface of the elbow is

$$\delta W_L^p = \int_A p\, \delta\mathbf{x} \cdot \mathbf{n}\, dA,$$

where

- p is the pressure magnitude,
- $\delta\mathbf{x}$ is the variational displacement at a point on the midsurface of the lateral wall of the elbow,
- $\mathbf{n}$ is the normal to the lateral wall midsurface, and
- A is the area of space occupied by the lateral surface in the current configuration.

The product of the surface normal and the differential area can be rewritten in terms of material coordinates S^0 along the pipe and ϕ around the pipe section:

$$\mathbf{n}\, dA = \frac{1}{r^0}\frac{\partial\mathbf{x}}{\partial\phi} \times \frac{\partial\mathbf{x}}{\partial S^0} r^0\, d\phi\, dS^0,$$

where r^0 is the initial pipe radius, so that

$$\delta W_L^p = \int_{A^0} p\, \frac{1}{r^0}\delta\mathbf{x} \cdot \frac{\partial\mathbf{x}}{\partial\phi} \times \frac{\partial\mathbf{x}}{\partial S^0} r^0\, d\phi\, dS^0,$$

where A^0 is the area of space occupied by the lateral surface in the reference configuration.

For hydrostatic pressure, the pressure magnitude is a function of position:

$$p = (z^0 - \mathbf{x} \cdot \mathbf{k})\frac{p^1}{(z^0 - z^1)},$$

where

- p^1 is the reference pressure magnitude,
- z^0 is the zero pressure height,
- z^1 is the reference height, and
- $\mathbf{k}$ is $(0, 0, 1)$, a unit vector in the vertical direction.

In the elbow elements position on the lateral surface, $\mathbf{x}$, is interpolated as

$$\mathbf{x} = \bar{\mathbf{x}} + (r^0 + u^r)\mathbf{r} + u^t\mathbf{t} + y^3\mathbf{a}_3,$$

where

$\bar{\mathbf{x}}(S^0)$	is the position of a point on the pipe axis,
$u^r(\phi, S^0)$	is the radial displacement,
$u^t(\phi, S^0)$	is the tangential displacement,
$\mathbf{r}$	is $\mathbf{a}_1 \cos\phi + \mathbf{a}_2 \sin\phi$, and
$\mathbf{t}$	is $-\mathbf{a}_1 \sin\phi + \mathbf{a}_2 \cos\phi$, with $\mathbf{a}_1(S^0)$ and $\mathbf{a}_2(S^0)$ being the cross-sectional basis vectors.

The first variation of the position can now be expressed as

$$\delta\mathbf{x} \;=\; \delta\bar{\mathbf{x}} + \delta u^r\mathbf{r} + \delta u^t\mathbf{t} + \delta y^3\mathbf{a}_3 + \delta\boldsymbol{\omega} \times [(r^0 + u^r)\mathbf{r} + u^t\mathbf{t} + y^3\mathbf{a}_3],$$

and the derivatives of the position with respect to the parametrization are

$$\frac{\partial\mathbf{x}}{\partial\phi} \;=\; (\frac{\partial u^r}{\partial\phi} - u^t)\mathbf{r} + (r^0 + u^r + \frac{\partial u^t}{\partial\phi})\mathbf{t} + \frac{\partial y^3}{\partial\phi}\mathbf{a}_3$$

and

$$\begin{aligned}\frac{\partial\mathbf{x}}{\partial S^0} \;&=\; [\frac{d\bar{\mathbf{x}}}{dS^0}\cdot\mathbf{r} + \frac{\partial u^r}{\partial S^0} + u^t(\frac{\partial\mathbf{t}}{\partial S^0}\cdot\mathbf{r}) + y^3(\frac{\partial\mathbf{a}_3}{\partial S^0}\cdot\mathbf{r})]\;\mathbf{r}\\ &+\; [\frac{d\bar{\mathbf{x}}}{dS^0}\cdot\mathbf{t} + \frac{\partial u^t}{\partial S^0} + (r^0 + u^r)\frac{\partial\mathbf{r}}{\partial S^0}\cdot\mathbf{t} + y^3\frac{\partial\mathbf{a}_3}{\partial S^0}\cdot\mathbf{t}]\;\mathbf{t}\\ &+\; [\frac{d\bar{\mathbf{x}}}{dS^0}\cdot\mathbf{a}_3 + \frac{\partial y^3}{\partial S^0}\cdot\mathbf{a}_3 + (r^0 + u^r)\frac{\partial\mathbf{r}}{\partial S^0}\cdot\mathbf{a}_3 + u^t\frac{\partial\mathbf{t}}{\partial S^0}\cdot\mathbf{a}_3]\;\mathbf{a}_3.\end{aligned}$$

Assuming that (1) terms in $(\partial\mathbf{r}/\partial S^0)\cdot\mathbf{t}$ and $(\partial\mathbf{t}/\partial S^0)\cdot\mathbf{r}$ can be ignored due to negligible twist in the pipe, (2) terms in y^3 and its derivatives can be ignored due to negligible warping in the pipe, (3) $\mathbf{a}_3(dS/dS^0) = d\bar{\mathbf{x}}/dS^0$, (4) the stretch dS/dS^0 is unity, and (5) u^r and u^t are small compared to r^0, we arrive at the following expression for the integrand of δW_L^p:

$$\begin{aligned}\frac{p}{r^0}\delta\mathbf{x}\cdot\frac{\partial\mathbf{x}}{\partial\phi}\times\frac{\partial\mathbf{x}}{\partial S^0} \;&=\; \frac{p\,\delta\bar{\mathbf{x}}}{r^0}\cdot(1 + r^0\frac{\partial\mathbf{r}}{\partial S^0}\cdot\mathbf{a}_3)[(r^0 + u^r + \frac{\partial u^t}{\partial\phi})\mathbf{r} + (u^t - \frac{\partial u^r}{\partial\phi})\mathbf{t}]\\ &+\; \frac{p\,\delta\boldsymbol{\omega}}{r^0}\cdot(1 + r^0\frac{\partial\mathbf{r}}{\partial S^0}\cdot\mathbf{a}_3)[(r^0 + u^r)(u^t - \frac{\partial u^r}{\partial\phi})\mathbf{a}_3 - u^t(r^0 + u^r + \frac{\partial u^t}{\partial\phi})\mathbf{a}_3]\\ &+\; \frac{p\,\delta u^r}{r^0}(1 + r^0\frac{\partial\mathbf{r}}{\partial S^0}\cdot\mathbf{a}_3)(r^0 + u^r + \frac{\partial u^t}{\partial\phi})\\ &+\; \frac{p\,\delta u^t}{r^0}(1 + r^0\frac{\partial\mathbf{r}}{\partial S^0}\cdot\mathbf{a}_3)(u^t - \frac{\partial u^r}{\partial\phi}).\end{aligned}$$

For closed-end loading the virtual work contribution of pressure on the end-caps of the elbow is

$$\delta W_E^p = \int_{E^0}\frac{p}{r}\delta\mathbf{y}\cdot\frac{\partial\mathbf{y}}{\partial r}\times\frac{\partial\mathbf{y}}{\partial\phi}\,r\,d\phi\,dr,$$

where r and ϕ are the material coordinates in a two-dimensional cylindrical coordinate system of points on the end-caps of the elbow element, $\mathbf{y}$ represents position on the end-caps, and E^0 is the area of space occupied by the end-caps of the elbow element. We assume the following deformation for the end-caps:

$$\mathbf{y}(r,\phi\,;\,S^0) \;=\; \bar{\mathbf{x}}(S^0) \;+\; \frac{r}{r^0}\,[\mathbf{x}(\phi\,;\,S^0) - \bar{\mathbf{x}}(S^0)],$$

where S^0 is the parameter that identifies the end-cap being considered. The assumed deformation arises naturally on considering a deformation of the end-caps in which radial rays of the reference end-cap configuration remain straight lines under deformation. It can be shown easily that the assumed deformation of the end-caps is differentiable as long as the deformations of the circumferential curves of the end-caps are differentiable. For the end-cap boundary shapes that arise in applications (primarily ovalized modes), the assumed deformation will be locally invertible so that integration of functions over the deformed surface is not likely to be a problem.

Ignoring the terms due to warping in the expression for position on the lateral surface, the first variation of position on an end-cap is

$$\delta\mathbf{y} \;=\; \delta\bar{\mathbf{x}} + \frac{r}{r^0}[\,(\delta u^r - u^t\delta\boldsymbol{\omega}\cdot\mathbf{a}_3)\mathbf{r} \;+\; (\delta u^t + (r^0+u^r)\delta\boldsymbol{\omega}\cdot\mathbf{a}_3)\mathbf{t} \;+\; (u^t\delta\boldsymbol{\omega}\cdot\mathbf{r} - (r^0+u^r)\delta\boldsymbol{\omega}\cdot\mathbf{t})\mathbf{a}_3],$$

and the derivatives of $\mathbf{y}$ with respect to r and ϕ are

$$\frac{\partial\mathbf{y}}{\partial r} \;=\; \frac{1}{r^0}[\,(r^0+u^r)\mathbf{r} + u^t\mathbf{t}]$$

and

$$\frac{\partial\mathbf{y}}{\partial\phi} \;=\; \frac{r}{r^0}[\,(\frac{\partial u^r}{\partial\phi} - u^t)\mathbf{r} + (r^0 + u^r + \frac{\partial u^t}{\partial\phi})\mathbf{t}\,].$$

The integrand of the expression for the virtual work of pressure on the end-caps can now be expressed as

$$\begin{aligned}\frac{p}{r}\delta\mathbf{y}\cdot\frac{\partial\mathbf{y}}{\partial r}\times\frac{\partial\mathbf{y}}{\partial\phi} \;&=\; p\,\{{r^0}^2 + 2r^0u^r \;+\; (r^0+u^r)\frac{\partial u^t}{\partial\phi}\\ &\quad -\; u^t\frac{\partial u^r}{\partial\phi}\}[\,\frac{1}{{r^0}^2}\delta\bar{\mathbf{x}}\cdot\mathbf{n} \;+\; \frac{r}{{r^0}^3}\delta\boldsymbol{\omega}\cdot\{u^t\mathbf{r} - (r^0+u^r)\mathbf{t}\}\,\mathbf{a}_3\cdot\mathbf{n}\,],\end{aligned}$$

where $\mathbf{n}$ is $\mathbf{a}_3$ if the center of the end-cap is node 1 of the element and $-\mathbf{a}_3$ if the center is node 2 or 3 of the element.

The load stiffness for the pressure loading, which by definition is the first variation of the virtual work of the pressure load, is given by $d(\delta W_L^p + \delta W_E^p)$. The following expressions are required for its calculation:

$$\begin{aligned}d(\frac{p}{r^0}\delta\mathbf{x}\cdot\frac{\partial\mathbf{x}}{\partial\phi}\times\frac{\partial\mathbf{x}}{\partial S^0}) \;&=\; \frac{dp}{r^0}\delta\mathbf{x}\cdot\frac{\partial\mathbf{x}}{\partial\phi}\times\frac{\partial\mathbf{x}}{\partial S^0}\\ &\quad +\; \{\frac{p\delta\mathbf{x}\cdot\frac{\partial\mathbf{x}}{\partial\phi}\times\frac{\partial\mathbf{x}}{\partial S^0}}{(1+r^0\frac{\partial\mathbf{r}}{\partial S^0}\cdot\mathbf{a}_3)}\}\,[(\frac{\partial d\boldsymbol{\omega}}{\partial S^0}\times\mathbf{r} + d\boldsymbol{\omega}\times\frac{\partial\mathbf{r}}{\partial S^0})\cdot\mathbf{a_3} \;+\; \frac{\partial\mathbf{r}}{\partial S^0}\cdot d\boldsymbol{\omega}\times\mathbf{a_3}]\end{aligned}$$

$$
\begin{aligned}
&+ \frac{p}{r^0}(1 + r^0 \frac{\partial \mathbf{r}}{\partial S^0} \cdot \mathbf{a}_3)\delta\bar{\mathbf{x}} \cdot [\,(du^r + \frac{\partial du^t}{\partial \phi})\mathbf{r} + (r^0 + u^r + \frac{\partial u^t}{\partial \phi})d\boldsymbol{\omega} \times \mathbf{r} \\
&\qquad + (du^t - \frac{\partial du^r}{\partial \phi})\mathbf{t} + (u^t - \frac{\partial u^r}{\partial \phi})d\boldsymbol{\omega} \times \mathbf{t}\,] \\
&+ \frac{p}{r^0}(1 + r^0 \frac{\partial \mathbf{r}}{\partial S^0} \cdot \mathbf{a}_3)\delta\boldsymbol{\omega} \cdot [\,\{\, du^r(u^t - \frac{\partial u^r}{\partial \phi}) + (r^0 + u^r)(du^t - \frac{\partial du^r}{\partial \phi}) \\
&\qquad - du^t(r^0 + u^r + \frac{\partial u^t}{\partial \phi}) - u^t(du^r + \frac{\partial du^t}{\partial \phi})\,\}\mathbf{a}_3 \\
&\qquad + \{\,(r^0 + u^r)(u^t - \frac{\partial u^r}{\partial \phi}) - u^t(r^0 + u^r + \frac{\partial u^t}{\partial \phi})\,\}d\boldsymbol{\omega} \times \mathbf{a}_3\,] \\
&+ \frac{p}{r^0}(1 + r^0 \frac{\partial \mathbf{r}}{\partial S^0} \cdot \mathbf{a}_3)\delta u^r(\, du^r + \frac{\partial du^t}{\partial \phi}\,) \\
&+ \frac{p}{r^0}(1 + r^0 \frac{\partial \mathbf{r}}{\partial S^0} \cdot \mathbf{a}_3)\delta u^t(\, du^t - \frac{\partial du^r}{\partial \phi}\,)
\end{aligned}
$$

and

$$
\begin{aligned}
d(\frac{p}{r}\delta\mathbf{y} \cdot \frac{\partial \mathbf{y}}{\partial r} \times \frac{\partial \mathbf{y}}{\partial \phi}) \;&= \frac{dp}{r}\delta\mathbf{y} \cdot \frac{\partial \mathbf{y}}{\partial r} \times \frac{\partial \mathbf{y}}{\partial \phi} \\
&+ p[\,2r^0 du^r + du^r \frac{\partial u^t}{\partial \phi} + (r^0 + u^r)\frac{\partial du^t}{\partial \phi} - du^t \frac{\partial u^r}{\partial \phi} - u^t \frac{\partial du^r}{\partial \phi}\,] \\
&\qquad [\,\frac{1}{{r^0}^2}\delta\bar{\mathbf{x}} \cdot \mathbf{n} + \frac{r}{{r^0}^3}(\mathbf{a}_3 \cdot \mathbf{n})\delta\boldsymbol{\omega} \cdot \{u^t\mathbf{r} - (r^0 + u^r)\mathbf{t}\}\,] \\
&+ p[\,{r^0}^2 + 2r^0 u^r + (r^0 + u^r)\frac{\partial u^t}{\partial \phi} - u^t \frac{\partial u^r}{\partial \phi}\,] \\
&\qquad [\,\frac{r}{{r^0}^3}(\mathbf{a}_3 \cdot \mathbf{n})\delta\boldsymbol{\omega} \cdot \{du^t\mathbf{r} + u^t d\boldsymbol{\omega} \times \mathbf{r} - du^r\mathbf{t} - (r^0 + u^r)d\boldsymbol{\omega} \times \mathbf{t}\} \\
&\qquad + \frac{1}{{r^0}^2}\delta\bar{\mathbf{x}} \cdot d\boldsymbol{\omega} \times \mathbf{n}\,].
\end{aligned}
$$

In the above dp is nonzero only in the case of hydrostatic pressure, when it is given in the first case by

$$dp = -\frac{p^1}{(z^0 - z^1)}\mathbf{k} \cdot d\mathbf{x}$$

and in the second case by

$$dp = -\frac{p^1}{(z^0 - z^1)}\mathbf{k} \cdot d\mathbf{y}.$$

Reference

- "Pipes and pipebends with deforming cross-sections: elbow elements," Section 29.5.1 of the Abaqus Analysis User's Guide

6.6 Multi-point constraints

- “Sliding constraint,” Section 6.6.1
- “Shell to solid constraint,” Section 6.6.2
- “Revolute joint,” Section 6.6.3
- “Universal joint,” Section 6.6.4
- “Local velocity constraint,” Section 6.6.5
- “Kinematic coupling,” Section 6.6.6

6.6.1 SLIDING CONSTRAINT

Products: Abaqus/Standard Abaqus/Explicit

The sliding constraint has a variety of uses. For example, this MPC is used in conjunction with other MPC types to constrain a shell element mesh to a solid element mesh. The MPC maintains consistency with standard shell theory by forcing initially straight lines through the thickness to remain straight despite rotation and displacement. When applied to solid element nodes on the shell-solid interface, this MPC enforces a kinematic approximation of compatibility with the shell model.

The theory of this constraint is as follows:

Let P^1, P^n be the points defining the line; and let P^m be the node that must lie on this line. The direction of the line is given by

$$\mathbf{n} = (\mathbf{x}^n - \mathbf{x}^1)/l,$$

where

$$l = \left[(\mathbf{x}^n - \mathbf{x}^1) \cdot (\mathbf{x}^n - \mathbf{x}^1)\right]^{\frac{1}{2}}.$$

Let $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ be base vectors in the x-, y-, z-directions in the global coordinate system. Then, define a unit vector normal to the line as

$$\mathbf{t}^1 = \frac{\mathbf{n} \times \mathbf{i}}{|\mathbf{n} \times \mathbf{i}|}$$

unless $\mathbf{n} = \mathbf{i}(n_2 = n_3 = 0)$, in which case we use

$$\mathbf{t}^1 = \frac{\mathbf{n} \times \mathbf{k}}{|\mathbf{n} \times \mathbf{k}|}.$$

Now we can define an orthogonal normal as

$$\mathbf{t}^2 = \mathbf{n} \times \mathbf{t}^1.$$

$\mathbf{t}^1$, $\mathbf{t}^2$, and $\mathbf{n}$ now form a set of orthonormal base vectors with $\mathbf{t}^1$ and $\mathbf{t}^2$ normal to the line joining P^1 and P^n. The constraint can be imposed by the condition that the line joining the node m to node 1 be perpendicular to $\mathbf{t}^1$ and $\mathbf{t}^2$; that is,

$$(\mathbf{x}^m - \mathbf{x}^1) \cdot \mathbf{t}^1 = 0$$

and

$$(\mathbf{x}^m - \mathbf{x}^1) \cdot \mathbf{t}^2 = 0.$$

We now choose a local coordinate numbering system such that i is the global direction on which $\mathbf{t}^1$ has its largest projection:

$$|t_i^1| > |t_j^1| \text{ and } |t_i^1| \geq |t_k^1|, \text{ where } j \neq i, k \neq i.$$

Likewise, we choose global direction j such that $j \neq i$ and

$$|t_j^2| > |t_k^2|, \text{ where } j \neq k.$$

Using this definition of i, j, k the constraint conditions can be written explicitly in terms of coordinate components of node m as

$$x_i^m t_i^1 + x_j^m t_j^1 + x_k^m t_k^1 = \mathbf{x}^1 \cdot \mathbf{t}^1$$

and

$$x_i^m t_i^2 + x_j^m t_j^2 + x_k^m t_k^2 = \mathbf{x}^1 \cdot \mathbf{t}^2.$$

These equations can be used to eliminate x_i^m, x_j^m (note that the numbering of i, j, k avoids dividing through by zero in this elimination):

$$x_i^m [t_i^1 t_j^2 - t_i^2 t_j^1] = \mathbf{x} \cdot (\mathbf{t}^1 t_j^2 - \mathbf{t}^2 t_j^1) - x_k^m (t_k^1 t_j^2 - t_k^2 t_j^1)$$

and

$$x_j^m [t_i^1 t_j^2 - t_j^1 t_i^2] = \mathbf{x} \cdot (\mathbf{t}^2 t_i^1 - \mathbf{t}^1 t_j^1) - x_k^m (t_k^1 t_i^2 - t_k^2 t_i^1).$$

The above equations will enforce the desired constraint. We also need the derivatives of these constraints. These are

$$(d\mathbf{x}^m - d\mathbf{x}^1) \cdot \mathbf{t}^1 + (\mathbf{x}^m - \mathbf{x}^1) \cdot d\mathbf{t}^1 = 0$$

and

$$(d\mathbf{x}^m - d\mathbf{x}^1) \cdot \mathbf{t}^2 + (\mathbf{x}^m - \mathbf{x}^1) \cdot d\mathbf{t}^2 = 0,$$

where

$$d\mathbf{t}^1 = (-\mathbf{t}^1 \cdot d\mathbf{n})\mathbf{n}$$

and

$$d\mathbf{t}^2 = (-\mathbf{t}^2 \cdot d\mathbf{n})\mathbf{n}.$$

These equations reduce to

$$(d\mathbf{x}^m - d\mathbf{x}^1) \cdot \mathbf{t}^1 - (\mathbf{x}^m - \mathbf{x}^1) \cdot \mathbf{n}\mathbf{t}^1 \cdot d\mathbf{n} = 0$$

and

$$(d\mathbf{x}^m - d\mathbf{x}^1) \cdot \mathbf{t}^2 - (\mathbf{x}^m - \mathbf{x}^1) \cdot \mathbf{n}\mathbf{t}^2 \cdot d\mathbf{n} = 0.$$

$d\mathbf{n}$ can be obtained from the definition of $\mathbf{n}$ to give

$$d\mathbf{n} = \frac{1}{l}(\mathbf{i} - \mathbf{n}\mathbf{n}) \cdot (d\mathbf{x}^n - d\mathbf{x}^1),$$

and, therefore,

$$\mathbf{t}^1 \cdot d\mathbf{n} = (\mathbf{t}^1/l) \cdot (d\mathbf{x}^n - d\mathbf{x}^1)$$

and

$$\mathbf{t}^2 \cdot d\mathbf{n} = (\mathbf{t}^2/l) \cdot (d\mathbf{x}^n - d\mathbf{x}^1).$$

The incremental constraint equations become

$$d\mathbf{x}^m \cdot \mathbf{t}^1 - d\mathbf{x}^1 \cdot \mathbf{t}^1 - (\mathbf{x}^m - \mathbf{x}^1) \cdot \mathbf{n}(\mathbf{t}^1/l) \cdot (d\mathbf{x}^n - d\mathbf{x}^1) = 0$$

and

$$d\mathbf{x}^m \cdot \mathbf{t}^2 - d\mathbf{x}^1 \cdot \mathbf{t}^1 - (\mathbf{x}^m - \mathbf{x}^1) \cdot \mathbf{n}(\mathbf{t}^2/l) \cdot (d\mathbf{x}^n - d\mathbf{x}^1) = 0.$$

Let $P = 1/l(\mathbf{x}^m - \mathbf{x}^1) \cdot \mathbf{n}$. Then, the above equations, when written out in full with the same ordering of $i,\ j,\ k$ used above, are

$$dx_i^m t_i^1 + dx_j^m t_j^1 + dx_k^m t_k^1 - P d\mathbf{x}^n \cdot \mathbf{t}^1 - (1 - P) d\mathbf{x}^1 \cdot \mathbf{t}^1 = 0$$

and

$$dx_i^m t_i^2 + dx_j^m t_j^2 + dx_k^m t_k^2 - P d\mathbf{x}^n \cdot \mathbf{t}^2 - (1 - P) d\mathbf{x}^1 \cdot \mathbf{t}^2 = 0.$$

Solving for $dx_i^m,\ dx_j^m$ we obtain

$$\begin{aligned} &dx_i^m(t_i^1 t_j^2 - t_i^2 t_j^1) + dx_k^m(t_k^1 t_j^2 - t_k^2 t_j^1) + \\ &\quad - P d\mathbf{x}^n \cdot (\mathbf{t}^1 t_j^2 - \mathbf{t}^2 t_j^1) - (1 - P) d\mathbf{x}^1 \cdot (\mathbf{t}^1 t_j^2 - \mathbf{t}^2 t_j^1) = 0 \end{aligned}$$

and

$$dx_j^m(t_j^2 t_i^1 - t_j^1 t_i^2) + dx_k^m(t_k^2 t_i^1 - t_k^1 t_i^2) +$$
$$- Pd\mathbf{x}^m \cdot (\mathbf{t}^2 t_i^1 - \mathbf{t}^1 t_i^2) - (1-P)d\mathbf{x}^1 \cdot (\mathbf{t}^2 t_i^1 - \mathbf{t}^1 t_i^2) = 0.$$

In the two-dimensional case $\mathbf{n}$ lies in the *x*–*y* or *r*–*z* plane and $\mathbf{t}^2 = (0, 0, -1)$. This implies that the second constraint equation is satisfied automatically. The remaining constraint equation is

$$x_i^m t_i^1 = \mathbf{x}^1 \cdot \mathbf{t}^1 - x_k^m t_k^1,$$

and its derivative is

$$dx_i^m t_i^1 + dx_k^m t_k^1 - Pd\mathbf{x}^n \cdot \mathbf{t}^1 - (1-P)d\mathbf{x}^1 \cdot \mathbf{t}^1 = 0.$$

Reference

- "General multi-point constraints," Section 35.2.2 of the Abaqus Analysis User's Guide

6.6.2 SHELL TO SOLID CONSTRAINT

Products: Abaqus/Standard Abaqus/Explicit

The shell to solid constraints SS LINEAR, SS BILINEAR, and SSF BILINEAR are used to connect shell elements to a solid element mesh. These MPCs are used in conjunction with the sliding constraint SLIDER (see "Sliding constraint," Section 6.6.1). The SLIDER MPC maintains consistency with standard shell theory by forcing initially straight lines through the thickness to remain straight despite rotation and displacement. Thus, the shell to solid MPC must enforce the remaining constraints:

- The displacement of the shell node at the interface must be equal to the displacement of the corresponding point on a line of nodes through the thickness of the solid;
- The rotation of the shell node at the interface must be compatible with the rotation of the corresponding line of nodes through the thickness.

The three MPC types impose essentially the same constraints but use different weighting factors to reflect the nature of the interpolations in the solid elements. SS LINEAR is used with first-order elements, SS BILINEAR is used at the edges of second-order elements, and SSF BILINEAR is used for the middle of second-order elements. The MPCs can be used in two-dimensional as well as three-dimensional models. The degrees of freedom will automatically adapt to the dimensionality of the problem. The shell to solid MPCs can be used with any number of points through the thickness of the solid. The weighting functions for the nodes in the solid will be chosen based on this number.

The displacement constraint for the shell node is obtained by setting the displacement of the shell node, $\mathbf{u}_s$, equal to the weighted average of the displacements of the nodes in the solid:

$$\mathbf{u}_s = \sum_{i=1}^{n} w_i \mathbf{u}_c^i, \tag{6.6.2–1}$$

where the MPC selects the appropriate weighting factors, w_i, based on the MPC type and the location of the nodes. If the SLIDER MPC is used to keep the nodes in the solid on a straight line, the choice of weighting factors does not influence the solution.

For the formulation of the rotation constraint, we assume that the nodes on the solid remain on one line. Hence, this line of nodes can be represented by the normalized direction $\mathbf{N}$ in the undeformed configuration and $\mathbf{n}$ in the deformed configuration. Let the rotation of the shell node be given by the finite rotation vector, $\boldsymbol{\phi}$. Then $\mathbf{N}$, $\mathbf{n}$, and $\boldsymbol{\phi}$ are related by the equation

$$\mathbf{C} \cdot \mathbf{N} = \mathbf{n}, \tag{6.6.2–2}$$

where $C = \exp[\hat{\boldsymbol{\phi}}]$ with $\hat{\boldsymbol{\phi}}$ the skew symmetric matrix form of the rotation vector $\boldsymbol{\phi}$. See "Rotation variables," Section 1.3.1, for a more detailed discussion of finite rotations.

This constraint equation does not completely define the rotation of the shell node: any solution to Equation 6.6.2–2 can be augmented by a rotation $\boldsymbol{\phi}_f = \mathbf{n}\phi_f$ around the line of nodes in the solid, where ϕ_f

can be chosen arbitrarily. Hence, Equation 6.6.2–2 only constrains the finite rotation vector ϕ to within two components. The linearized form of the rotation constraint is

$$\delta\boldsymbol{\theta} \times \mathbf{n} = d\mathbf{n}, \tag{6.6.2–3}$$

where $\delta\boldsymbol{\theta}$ is the linearized rotation. See "Rotation variables," Section 1.3.1, for a more detailed discussion of the linearized rotation.

We now define two local directions, $\mathbf{s}$ and $\mathbf{t}$, so that $\mathbf{s}$, $\mathbf{t}$, and $\mathbf{n}$ form a right-handed, orthonormal, local coordinate system. We then project Equation 6.6.2–3 onto $\mathbf{s}$ and $\mathbf{t}$, which yields

$$(\delta\boldsymbol{\theta} \times \mathbf{n}) \cdot \mathbf{s} = d\mathbf{n} \cdot \mathbf{s}, \qquad (\delta\boldsymbol{\theta} \times \mathbf{n}) \cdot \mathbf{t} = d\mathbf{n} \cdot \mathbf{t}.$$

With some standard vector algebra these equations can be transformed into the form

$$\delta\boldsymbol{\theta} \cdot \mathbf{t} = d\mathbf{n} \cdot \mathbf{s}, \qquad \delta\boldsymbol{\theta} \cdot \mathbf{s} = -d\mathbf{n} \cdot \mathbf{t}. \tag{6.6.2–4}$$

The change in the normal can be expressed in terms of the displacement difference between the two extreme nodes 1 and n in the continuum:

$$\delta\mathbf{n} = \frac{1}{h}\delta\Delta\mathbf{u} \cdot (\mathbf{I} - \mathbf{nn}),$$

where $\Delta\mathbf{u} = \mathbf{u}_n - \mathbf{u}_1$ is the displacement difference and $h = \mathbf{n} \cdot (\mathbf{x}_n - \mathbf{x}_1)$ is the distance between the nodes. Substitution in Equation 6.6.2–4 yields the constraint equations

$$\delta\boldsymbol{\theta} \cdot \mathbf{t} = \frac{1}{h}\delta\Delta\mathbf{u} \cdot \mathbf{s}, \qquad \delta\boldsymbol{\theta} \cdot \mathbf{s} = -\frac{1}{h}\delta\Delta\mathbf{u} \cdot \mathbf{t}. \tag{6.6.2–5}$$

These constraint equations are formulated in terms of local components of $\delta\boldsymbol{\theta}$. To obtain the constraint in terms of global components of $\delta\boldsymbol{\theta}$, we choose a basis $(\mathbf{e}_i, \mathbf{e}_j, \mathbf{e}_k)$ that is a cyclic permutation of the global basis $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$, such that $n_k \geq n_i$ and $n_k \geq n_j$. The constraints can then be written in component form as follows:

$$\begin{aligned} \delta\theta_i t_i + \delta\theta_j t_j + \delta\theta_k t_k &= \frac{1}{h}\delta\Delta\mathbf{u} \cdot \mathbf{s}, \\ \delta\theta_i s_i + \delta\theta_j s_j + \delta\theta_k s_k &= -\frac{1}{h}\delta\Delta\mathbf{u} \cdot \mathbf{t}. \end{aligned} \qquad \text{(no summation)}$$

From these two equations we solve for $\delta\theta_i$ and $\delta\theta_j$:

$$\begin{aligned} \delta\theta_i &= \frac{n_i}{n_k}\delta\theta_k - \frac{1}{hn_k}(s_j\delta\Delta\mathbf{u} \cdot \mathbf{s} + t_j\delta\Delta\mathbf{u} \cdot \mathbf{t}), \\ \delta\theta_j &= \frac{n_j}{n_k}\delta\theta_k + \frac{1}{hn_k}(s_i\delta\Delta\mathbf{u} \cdot \mathbf{s} + t_i\delta\Delta\mathbf{u} \cdot \mathbf{t}). \end{aligned} \tag{6.6.2–6}$$

The linearized constraints shown above are used directly in geometrically linear analysis, linear perturbations, and Abaqus/Explicit. In geometrically nonlinear analysis in Abaqus/Standard, the linearized constraints are used to solve for the general nonlinear constraint Equation 6.6.2–2 with a Newton method. When the rotation is large, the permutation i, j, k may change to maintain the conditions that $n_k \geq n_i$ and $n_k \geq n_j$.

Reference

- "General multi-point constraints," Section 35.2.2 of the Abaqus Analysis User's Guide

6.6.3 REVOLUTE JOINT

Products: Abaqus/Standard Abaqus/Explicit

A revolute joint is a joint between two nodes in which the rotations of the nodes differ by a relative rotation about an axis that is fixed in and, therefore, rotates with the joint. A simple example is a hinge.

A revolute joint is implemented in Abaqus/Standard as a multi-point constraint, defining the total rotation of the constrained ("slave") node (the first node of the MPC), $\mathbf{C}(\boldsymbol{\phi}^S)$, as the total rotation of the "master node" (the second node of the MPC), $\mathbf{C}(\boldsymbol{\phi}^M)$, followed by the relative rotation ϕ^J, about the axis of the joint $\mathbf{a}$:

$$\mathbf{C}(\boldsymbol{\phi}^S) = \mathbf{C}(\phi^J \mathbf{a}) \cdot \mathbf{C}(\boldsymbol{\phi}^M).$$

The joint axis, $\mathbf{a}$, also rotates with the rotation of the master node:

$$\mathbf{a} = \mathbf{C}(\boldsymbol{\phi}^M) \cdot \mathbf{a}\big|_0.$$

The angular velocity of the slave node is

$$\dot{\boldsymbol{\phi}}^S = \dot{\boldsymbol{\phi}}^M + \dot{\phi}^J \mathbf{a},$$

and the virtual variations of the rotations are, likewise,

$$\delta\boldsymbol{\phi}^S = \delta\boldsymbol{\phi}^M + \delta\phi^J \mathbf{a}. \tag{6.6.3–1}$$

Thus, the joint imposes three constraints (each component of the angular velocity of the slave node is constrained) but introduces an additional degree of freedom in the form of the relative rotation ϕ^J. This means the joint provides a total of two constraints to the model if ϕ^J is not prescribed and three constraints if it is.

The virtual work contribution of the three nodes of the joint is

$$\mathbf{M}^S \cdot \delta\boldsymbol{\phi}^S + \mathbf{M}^M \cdot \delta\boldsymbol{\phi}^M + M^J \delta\phi^J = 0,$$

where $\mathbf{M}^S$ is the total moment at node S, $\mathbf{M}^M$ is the total moment at node M, and M^J is the moment in the joint. Applying the constraints (Equation 6.6.3–1), this is

$$(\mathbf{M}^S + \mathbf{M}^M) \cdot \delta\boldsymbol{\phi}^M + (\mathbf{M}^S \cdot \mathbf{a} + M^J)\delta\phi^J = 0.$$

If there are no further constraints associated with the nodes of the joint, $\delta\boldsymbol{\phi}^M$ and $\delta\phi^J$ are independent variations, so the constrained virtual work equation implies that

$$\mathbf{M}^S = -\mathbf{M}^M$$

and that

$$M^J = -\mathbf{M}^S \cdot \mathbf{a} = \mathbf{M}^M \cdot \mathbf{a}.$$

Because the revolute is implemented in this manner, the relative rotation in the joint ϕ^J appears as a degree of freedom in the model (degree of freedom 6 at the third node of the MPC). Thus, a moment, M^J, can be applied in the joint by specifying its value as a concentrated load; ϕ^J can be given a prescribed variation in time by specifying a boundary condition; or stiffness and/or damping can be associated with relative rotation of the joint by attaching a spring and/or dashpot to ground to this degree of freedom (a spring or dashpot to ground is used because the variable is a relative rotation).

Reference

- "General multi-point constraints," Section 35.2.2 of the Abaqus Analysis User's Guide

6.6.4 UNIVERSAL JOINT

Products: Abaqus/Standard Abaqus/Explicit

A universal joint is a joint between two nodes containing orthogonal hinges that provide two axes of relative rotation in the joint.

A universal joint is implemented in Abaqus/Standard as a multi-point constraint, defining the total rotation of the constrained ("slave") node (the first node of the MPC), $\mathbf{C}(\boldsymbol{\phi}^S)$, as the total rotation of the "master node" (the second node of the MPC), $\mathbf{C}(\boldsymbol{\phi}^M)$, followed by two relative rotations: ϕ_1 about the first axis of the joint $\mathbf{a}_1$, then ϕ_2 about the second axis of the joint $\mathbf{a}_2$ (which is orthogonal to $\mathbf{a}_1$):

$$\mathbf{C}(\boldsymbol{\phi}^S) = \mathbf{C}(\phi_2\mathbf{a}_2) \cdot \mathbf{C}(\phi_1\mathbf{a}_1) \cdot \mathbf{C}(\boldsymbol{\phi}^M).$$

The first joint axis, $\mathbf{a}_1$, rotates with the rotation of the master node:

$$\mathbf{a}_1 = \mathbf{C}(\boldsymbol{\phi}^M) \cdot \mathbf{a}_1\big|_0.$$

The second joint axis has this rotation plus the rotation about the first joint axis:

$$\mathbf{a}_2 = \mathbf{C}(\phi_1\mathbf{a}_1) \cdot \mathbf{C}(\boldsymbol{\phi}^M) \cdot \mathbf{a}_2\big|_0.$$

The angular velocity of the slave node is

$$\dot{\boldsymbol{\phi}}^S = \dot{\boldsymbol{\phi}}^M + \dot{\phi}_1\mathbf{a}_1 + \dot{\phi}_2\mathbf{a}_2;$$

and the virtual variations of the rotations are, likewise,

$$\delta\boldsymbol{\phi}^S = \delta\boldsymbol{\phi}^M + \delta\phi_1\mathbf{a}_1 + \delta\phi_2\mathbf{a}_2. \tag{6.6.4–1}$$

Thus, the joint imposes three constraints (each component of the angular velocity of the slave node is constrained) but introduces two additional degrees of freedom in the form of the relative rotations ϕ_1 and ϕ_2. This means the joint provides a total of one constraint to the model if ϕ_1 and ϕ_2 are not prescribed or up to three constraints if they are.

The virtual work contribution of the joint is

$$\mathbf{M}^S \cdot \delta\boldsymbol{\phi}^S + \mathbf{M}^M \cdot \delta\boldsymbol{\phi}^M + M_1\delta\phi_1 + M_2\delta\phi_2 = 0,$$

where $\mathbf{M}^S$ is the total moment at node $\boldsymbol{S}$, $\mathbf{M}^M$ is the total moment at node $\boldsymbol{M}$, and M_1 and M_2 are the moments in the joint hinges. Applying the constraints (Equation 6.6.4–1), this is

$$(\mathbf{M}^S + \mathbf{M}^M) \cdot \delta\boldsymbol{\phi}^M + (\mathbf{M}^S \cdot \mathbf{a}_1 + M_1)\delta\phi_1 + (\mathbf{M}^S \cdot \mathbf{a}_2 + M_2)\delta\phi_2 = 0.$$

If there are no further constraints associated with the nodes of the joint, $\delta\boldsymbol{\phi}^M$, $\delta\phi_1$ and $\delta\phi_2$ are independent variations, so that the constrained virtual work equation implies that

$$\mathbf{M}^S = -\mathbf{M}^M,$$

$$M_1 = -\mathbf{M}^S \cdot \mathbf{a}_1 = \mathbf{M}^M \cdot \mathbf{a}_1$$

and

$$M_2 = -\mathbf{M}^S \cdot \mathbf{a}_2 = \mathbf{M}^M \cdot \mathbf{a}_2.$$

Because the universal joint is implemented in this manner, the relative rotations in the joint, ϕ_1 and ϕ_2, appear as degrees of freedom in the model (degree of freedom 6 at the third and fourth nodes of the MPC). Moments M_1 and M_2 can, therefore, be applied in the joint by specifying their values as concentrated loads; ϕ_1 and ϕ_2 can be given prescribed variations in time by specifying boundary conditions; or stiffness and/or damping can be associated with relative rotations of the joint by attaching springs and/or dashpots to ground to these degrees of freedom (springs or dashpots to ground are used because the variables are relative rotations).

Reference

- "General multi-point constraints," Section 35.2.2 of the Abaqus Analysis User's Guide

6.6.5 LOCAL VELOCITY CONSTRAINT

Product: Abaqus/Standard

The V LOCAL MPC in Abaqus/Standard constrains the velocity components at the first MPC node to be equal to the velocity components at the third node along local, rotating, directions. These local directions rotate according to the rotation at the second node. In the initial configuration the first local direction is from the second to the third node of the MPC. The global z-axis is used if these nodes coincide. The velocity of the first node $\dot{\mathbf{u}}^1$, is as follows:

$$\dot{\mathbf{u}}^1 = \sum_{i=1}^{3} \mathbf{e}_i^2 \dot{u}_i^3.$$

The constraint is integrated approximately to define

$$\Delta \mathbf{u}^1 = \sum_{i=1}^{3} \mathbf{e}_i^2\big|_{t+\frac{1}{2}\Delta t} \Delta u_i^3,$$

where

$$\mathbf{e}_i^2\big|_{t+\frac{1}{2}\Delta t} \stackrel{\text{def}}{=} \mathbf{C}\big|_{t+\frac{1}{2}\Delta t} \cdot \mathbf{e}_i^2\big|_t,$$

where

$$\mathbf{C}\big|_{t+\frac{1}{2}\Delta t} = \mathbf{C}(\frac{1}{2}\Delta\boldsymbol{\phi}^2)$$

is the increment of rotation defined by half of the magnitude of the increment of rotation at the second node of the constraint, $\Delta\boldsymbol{\phi}^2$, and $\mathbf{e}_i^2\big|_t$, $i = 1, 2, 3$, are the local directions at the beginning of the increment.

Reference

- "General multi-point constraints," Section 35.2.2 of the Abaqus Analysis User's Guide

6.6.6 KINEMATIC COUPLING

Product: Abaqus/Standard

A kinematic coupling constrains a group of slave nodes to the translation and rotation of a master node in a customized manner; combinations of slave node degrees of freedom are selected to participate in the constraint. Since each slave node has a separate relationship with the master node, the kinematic coupling constraint can be considered as the combination of general master-slave constraints. These general constraints are described below.

Rotations

Each possible combination of selected constraints on a slave node rotation results in rotation relationships that are unique but analogous to existing MPC types: zero constrained rotational degrees of freedom results in a pin constraint; one, results in a universal constraint; two, results in a revolute constraint; and three, results in a beam constraint. Each of these combinations is treated according to the appropriate theory employed in Abaqus/Standard. To implement these constraints, an additional node is created internally for each slave node. For example, for revolute and universal constraints this additional node is used in a similar manner to the nodes required in the specification of the analog MPC types.

Translations

The additional internal node, described above, also reflects the motion of the slave node, relative to that of a fully constrained slave node in the following manner. Let $\mathbf{X}^m$ be the position of the master node and $\mathbf{X}^s$ be the position of the slave node in the reference configuration. The reference configuration position of the slave node with respect to the master node is then

$$\mathbf{N} = \mathbf{X}^s - \mathbf{X}^m.$$

Then, in the current configuration

$$\hat{\mathbf{x}}^s = \mathbf{x}^m + \mathbf{n},$$

where $\hat{\mathbf{x}}^s$ is the fully constrained slave node position and

$$\mathbf{n} = \mathbf{C}(\boldsymbol{\phi}^m) \cdot \mathbf{N},$$

where $\mathbf{C}(\boldsymbol{\phi}^m)$ is the rotation matrix associated with the master node rotation, $\boldsymbol{\phi}^m$. The selectively constrained slave node position can be described as

$$\mathbf{x}^s = \hat{\mathbf{x}}^s + y_i \mathbf{e}_i, \tag{6.6.6–1}$$

where y_i are the translation degrees of freedom at the additional node and $\mathbf{e}_i$ are the current configuration base vectors, which rotate from the reference global Cartesian base vectors $\mathbf{e}_i$ according to

$$\mathbf{e}_i = \mathbf{C}(\boldsymbol{\phi}^m) \cdot \mathbf{e}_i.$$

This base vector rotation is made regardless of the choice of rotation constraint at the slave node. Constraint or release of slave node translation degree of freedom i can now be described as the constraint ($y_i = 0$) or release of translation degrees of freedom on the additional node. With these constraints on y_i, Equation 6.6.6–1 can be used to define the constraint equations.

Linearized form

The linearized form is readily obtained as

$$\begin{aligned} \delta\mathbf{x}^s &= \delta\mathbf{x}^m + \delta\mathbf{n} + \delta y_i \mathbf{e}_i + y_i \delta\mathbf{e}_i \\ &= \delta\mathbf{x}^m + \delta\boldsymbol{\phi}^m \times \mathbf{r} + \delta y_i \mathbf{e}_i, \end{aligned} \tag{6.6.6–2}$$

where $\mathbf{r} = \mathbf{x}^s - \mathbf{x}^m$.

Second-order form

The initial stress stiffness terms can be obtained from

$$\begin{aligned} \mathrm{d}\delta\mathbf{x}^s &= \mathrm{d}\delta\mathbf{n} + y_i \mathrm{d}\delta\mathbf{e}_i + \delta y_i \mathrm{d}\mathbf{e}_i + \mathrm{d}y_i \delta\mathbf{e}_i \\ &= -(\delta\boldsymbol{\phi}^m \cdot \mathrm{d}\boldsymbol{\phi}^m)\mathbf{r} + \tfrac{1}{2}(\delta\boldsymbol{\phi}^m \cdot \mathbf{r})\mathrm{d}\boldsymbol{\phi}^m + \tfrac{1}{2}(\mathrm{d}\boldsymbol{\phi}^m \cdot \mathbf{r})\delta\boldsymbol{\phi}^m + \delta y_i \mathrm{d}\boldsymbol{\phi}^m \times \mathbf{e}_i + \mathrm{d}y_i \delta\boldsymbol{\phi}^m \times \mathbf{e}_i. \end{aligned} \tag{6.6.6–3}$$

Reference

- "Kinematic coupling constraints," Section 35.2.3 of the Abaqus Analysis User's Guide

7. References

References 7.1

7.1 References

- “References,” Section 7.1.1

7.1.1 REFERENCES

Agah-Tehrani, A., E. H. Lee, R. L. Mallett, and E. T. Onat, "The Theory of Elastic-Plastic Deformation at Finite Strain with Induced Anisotropy Modeled as Combined Isotropic-Kinematic Hardening," Metal Forming Report, Rensselaer Polytechnic Institute, Troy, New York, 1986.

Al-Ani, A. M., and J. W. Hancock, "J-Dominance of Short Cracks in Tension and Bending," Journal of the Mechanics and Physics of Solids, vol. 39, pp. 23–43, 1991.

Allik, H., "The Application of Finite and Infinite Elements to Problems in Structural Acoustics," Computational Mechanics '91: Proceedings of the International Conference on Computational Engineering Science, ICES Publications, Atlanta, Georgia, 1991.

Anagnastopoulos, S. A., "Response Spectrum Techniques for Three Component Earthquake Design," Earthquake Engineering and Structural Dynamics, vol. 9, pp. 459–476, 1981.

Aoki, S., K. Kishimoto, and M. Sakjata, "Crack Tip Stress and Strain Singularity in Thermally Loaded Elastic-Plastic Material," Transactions of the ASME, Journal of Applied Mechanics, vol. 48, no. 2, pp. 428–429, 1981.

Aravas, N., "On the Numerical Integration of a Class of Pressure-Dependent Plasticity Models," International Journal for Numerical Methods in Engineering, vol. 24, pp. 1395–1416, 1987.

Aravas, N., and E. C. Aifantis, "On the Geometry of Slip and Spin in Finite Plastic Deformation," International Journal of Plasticity, vol. 7, pp. 141–160, 1991.

Arruda, E. M., and M. C. Boyce, "A Three-Dimensional Constitutive Model for the Large Stretch Behavior of Rubber Elastic Materials," Journal of the Mechanics and Physics of Solids, vol. 41, no. 2, pp. 389–412, 1993.

Ashwell, D. G., and R. H. Gallagher, Editors, *Finite Elements for Thin Shells and Curved Members,* John Wiley and Sons, London, 1976.

Astley, R. J., G. J. Macaulay, and J. P. Coyette, "Mapped Wave Envelope Elements for Acoustical Radiation and Scattering," Journal of Sound and Vibration, vol. 170, pp. 97–118, 1994.

Atomic Energy Commission Regulatory Guide 1.60, "Design Response Spectra for Seismic Design of Nuclear Power Plants."

Atomic Energy Commission Regulatory Guide 1.92, "Combining Modal Responses."

Barlow, J., "Optimal Stress Locations in Finite Element Models," International Journal for Numerical Methods in Engineering, vol. 10, pp. 243–251, 1976.

Barnett, D. M., and R. J. Asaro, "The Fracture Mechanics of Slit-Like Cracks in Anisotropic Elastic Media," Journal of the Mechanics and Physics of Solids, vol. 20, pp. 353–366, 1972.

Bathe, K. J., and C. A. Almeida, "A Simple and Effective Pipe Elbow Element—Linear Analysis," Transactions of the ASME, Journal of Applied Mechanics, vol. 47, no. 1, 1980.

Bathe, K. J., and E. N. Dvorkin, “A Continuum Mechanics-Based Four-Node Shell Element for General Non-Linear Analysis,” International Journal of Computer Aided Engineering Software, vol. 1, 1984.

Bathe, K. J., and E. L. Wilson, “Large Eigenvalue Problems in Dynamic Analysis,” Proceedings of the ASCE, EM6, 98, pp. 1471–1485, 1972.

Batoz, J. L., K. J. Bathe, and L. W. Ho, “A Study of Three-Node Triangular Plate Bending Elements,” International Journal for Numerical Methods in Engineering, vol. 15, pp. 1771–1821, 1980.

Bayliss, A., M. Gunzberger, and E. Turkel, “Boundary Conditions for the Numerical Solution of Elliptic Equations in Exterior Regions,” SIAM Journal of Applied Mathematics., vol. 42, no. 2, pp. 430–451, 1982.

Bear, J., *Dynamics of Fluids in Porous Media,* American Elsevier Publishing Company, Dover, New York, 1972.

Belytschko, T., “Survey of Numerical Methods and Computer Programs for Dynamic Structural Analysis,” Nuclear Engineering and Design, vol. 37, pp. 23–34, 1976.

Belytschko, T., and L. P. Bindeman, “Assumed Strain Stabilization of the Eight Node Hexahedral Element,” Computer Methods in Applied Mechanics and Engineering, vol. 105, pp. 225–260, 1993.

Belytschko, T., J. I. Lin, and C. S. Tsay, “Explicit Algorithms for the Nonlinear Dynamics of Shells,” Computer Methods in Applied Mechanics and Engineering, vol. 43, pp. 251–276, 1984.

Belytschko, T., B. L. Wong, and H. Y. Chiang, “Advances in One-Point Quadrature Shell Elements,” Computer Methods in Applied Mechanics and Engineering, vol. 96, pp. 93–107, 1992.

Bergan, P. B., G. Horrigmoe, B. Krakeland, and T. H. Soreide, “Solution Techniques for Non-Linear Finite Element Problems,” International Journal for Numerical Methods in Engineering, vol. 12, pp. 1677–1696, 1978.

Bergström, J. S., and M. C. Boyce, “Constitutive Modeling of the Large Strain Time-Dependent Behavior of Elastomers,” Journal of the Mechanics and Physics of Solids, vol. 46, pp. 931–954, 1998.

Betegón, C., and J. W. Hancock, “Two-Parameter Characterization of Elastic-Plastic Crack-Tip Fields,” Journal of Applied Mechanics, vol. 58, pp. 104–110, 1991.

Bettess, P., C. Emson, and T. C. Chaim, “A New Mapped Infinite Element for Exterior Wave Problems,” *Numerical Methods in Coupled Systems*, Edited by R. W. Lewis et al, John WIley & Sons, 1984.

Bilby, B. A., G. E. Goldthorpe, and I. C. Howard, “A Finite Element Investigation of the Effect of Specimen Geometry on the Fields of Stress and Strain at the Tip of Stationary Cracks,” *Size Effects in Fracture*, Institution of Mechanical Engineers, London, pp. 37–46, 1986.

Bilkhu, S., Private communication, 1987.

Budiansky, B., and J. L. Sanders, “On the ‘Best’ First-Order Linear Shell Theory,” *Progress in Applied Mechanics,* The Prager Anniversary Volume, Macmillan, London, pp. 129–140, 1963.

Burnett, D. S., “A Three-Dimensional Acoustic Infinite Element Based on a Prolate Spheroidal Multipole Expansion,” Journal of the Acoustical Society of America, vol. 96(5), pp. 2798–2816, 1994.

Calladine, C. R., "A Microstructural View of the Mechanical Properties of Saturated Clay," Geotechnique, vol. 21, no. 4, pp. 391–415.

Chen, W. F., and D. J. Han, *Plasticity for Structural Engineers,* Springer-Verlag, New York, 1988.

Chu, C. C., and A. Needleman, "Void Nucleation Effects in Biaxially Stretched Sheets," Journal of Engineering Materials and Technology, vol. 102, pp. 249–256, 1980.

Clough, R. W., and J. Penzien, *Dynamics of Structures,* McGraw-Hill, New York, 1975.

Coffin, L. F., Jr., "The Flow and Fracture of a Brittle Material," Journal of Applied Mechanics, vol. 72, pp. 233–248, 1950.

Cohen, M., and P. C. Jennings, *Silent Boundary Methods for Transient Analysis* (in *Computational Methods for Transient Analysis*), Ed. T. Belytschko and T. R. J. Hughes, Elsevier, 1983.

Cormeau, I., "Numerical Stability in Quasi-Static Elasto-Visco Plasticity," International Journal for Numerical Methods in Engineering, vol. 9, pp. 109–127, 1975.

Cotterell, B., and J. R. Rice, "Slightly Curved or Kinked Cracks," International Journal of Fracture, vol. 16, pp. 155–169, 1980.

Cowper, G. R., "Gaussian Quadrature for Triangles," International Journal for Numerical Methods in Engineering, vol. 7, pp. 405–408, 1973.

Crank, J., *The Mathematics of Diffusion,* Clarendon Press, Oxford, 1956.

Crisfield, M. A., "A Fast Incremental/Iteration Solution Procedure that Handles 'Snap-Through'," Computers and Structures, vol. 13, pp. 55–62, 1981.

Crisfield, M. A., "Snap-Through and Snap-Back Response in Concrete Structures and the Dangers of Under-Integration," International Journal for Numerical Methods in Engineering, vol. 22, pp. 751–767, 1986.

Czekanski, A., N. El-Abbasi, and S. A. Meguid, "Optimal Time Integration Parameters for Elastodynamic Contact Problems," Communications in Numerical Methods in Engineering, vol. 17, pp. 379–384, 2001.

DeGroot, S. R., and P. Mazur, *Non Equilibrium Thermodynamics,* North Holland Publishing Company, North Holland, Amsterdam, 1962.

Desai, C. S., "Finite Element Methods for Flow in Porous Media" in *Finite Elements in Fluids*, vol. 1, Wiley, London, pp. 157–181, 1975.

Deshpande, V. S., and N. A. Fleck, "Isotropic Constitutive Model for Metallic Foams," Journal of the Mechanics and Physics of Solids, vol. 48, pp. 1253–1276, 2000.

Dodge, W. G., and S. E. Moore, "Stress Indices and Flexibility Factors for Moment Loadings on Elbows and Curved Pipes," Welding Research Council Bulletin, no. 179, December 1972.

Drucker, D. C., and W. Prager, "Soil Mechanics and Plastic Analysis or Limit Design," Quarterly of Applied Mathematics, vol. 10, pp. 157–165, 1952.

Du, Z. -Z., and J. W. Hancock, "The Effect of Non-Singular Stresses on Crack-Tip Constraint," Journal of the Mechanics and Physics of Solids, vol. 39, pp. 555–567, 1991.

Dupuis, G., "Stabilité Elastique des Structures Unidimensionelles," ZAMP, vol. 20, pp. 94–106, 1969.

Eleiche, A. S. M., "A Literature Survey of the Combined Effects of Strain Rate and Elevated Temperature on the Mechanical Properties of Metals," Air Force Materials Laboratories, Report AFML–TR–72–125, 1972.

Engelmann, B. E., and R. G. Whirley, *A New Explicit Shell Element Formulation for Impact Analysis* (in *Computational Aspects of Contact, Impact and Penetration*), Ed. R. F. Kulak and L. E. Schwer, Elmepress International, 1990.

Engquist, B., and A. Majda, "Absorbing Boundary Conditions for the Numerical Simulation of Waves," Mathematics of Computation, vol. 31, pp. 629–651, 1977.

Ericsson, T., and A. Ruhe, "The Spectral Transformation Lanczos Method for the Numerical Solution of Large Sparse Generalized Symmetric Eigenvalue Problems," Mathematics of Computation, vol. 35, pp. 1251–1268, 1980.

Erdogan, F., and G. C. Sih, "On the Crack Extension in Plates under Plane Loading and Transverse Shear," Journal of Basic Engineering, vol. 85, pp. 519–527, 1963.

Everstine, G., "A Symmetric Potential Formulation for Fluid-Structure Interaction," Journal of Sound and Vibration, vol. 79, pp. 157–160, 1981.

Faltinsen, O. M., *Sea Loads on Ships and Offshore Structures,* Cambridge University Press, 1990.

Farin, G., *Curves and Surfaces for Computer Aided Geometric Design,* Academic Press, San Diego, Second Edition edition, 1990.

Farin, G., "Smooth Interpolation to Scattered 3D Data" in *Surfaces in Computer Aided Geometric Design,* Barnhill, R. E. and Boehm, W., eds., North-Holland, pp. 43–63, 1983.

Flanagan, D. P., and T. Belytschko, "A Uniform Strain Hexahedron and Quadrilateral with Orthogonal Hourglass Control," International Journal for Numerical Methods in Engineering, vol. 17, pp. 679–706, 1981.

Flugge, W., *Tensor Analysis and Continuum Mechanics,* Springer-Verlag, New York, 1972.

Fung, Y. C., K. Fronek, and P. Patitucci, "Pseudoelasticity of Arteries and the Choice of its Mathematical Expressions," American Journal of Physiology, vol. 237, pp. H620–H631, 1979.

Gao, H., M. Abbudi, and D. M. Barnett, "Interfacial Crack-Tip Fields in Anisotropic Elastic Solids," Journal of the Mechanics and Physics of Solids, vol. 40, pp. 393–416, 1992.

Gasser, T. C., G. A. Holzapfel, and R. W. Ogden, "Hyperelastic Modelling of Arterial Layers with Distributed Collagen Fibre Orientations," Journal of the Royal Society Interface, vol. 3, pp. 15–35, 2006.

Geers, T. L., and K. S. Hunter, "An Integrated Wave-Effects Model for an Underwater Explosion Bubble," Journal of the Acoustical Society of America, vol. 111(4), pp. 1584–1601, 2002.

Geers, T. L., and C. K. Park, "Optimization of the G & H Bubble Model," Shock and Vibration, vol. 12(1), pp. 3–8, 2005.

Gibson, L. J., and M. F. Ashby, "The Mechanics of Three-Dimensional Cellular Materials," Proceedings of the Royal Society, London, A 382, pp. 43–59, 1982.

Gibson, L. J., M. F. Ashby, G. S. Schajer, and C. I. Robertson, "The Mechanics of Two-Dimensional Cellular Materials," Proceedings of the Royal Society, London, A 382, pp. 25–42, 1982.

Gordon, J. L., "OUTCUR: An Automated Evaluation of Two-Dimensional Finite Element Stresses According to ASME Section III Stress Requirements," Paper No. 76–WA/PVP-16, ASME Winter Annual Meeting, December 1976.

Govindarajan, S. M., J. A. Hurtado, and W. V. Mars, "Simulation of Mullins Effect and Permanent Set in Filled Elastomers Using Multiplicative Decomposition," Proceedings of the 5th European Conference for Constitutive Models for Rubber, Paris, September 2007.

Green, A. E., and P. M. Naghdi, "A General Theory of an Elastic-Plastic Continuum," Archives of Rational Mechanics and Analysis, vol. 17.

Grimes, R. G., J. G. Lewis, and H. D. Simon, "A Shifted Block Lanczos Algorithm for Solving Sparse Symmetric Generalized Eigenproblems," SIAM Journal on Matrix Analysis and Applications, vol. 15, pp. 228–272, 1994.

Grote, M., and J. Keller, "On Nonreflecting Boundary Conditions," Journal of Computational Physics, vol. 122, pp. 231–243, 1995.

Gudehus, G., Editor, *Finite Elements in Geomechanics,* Wiley and Sons, 1977.

Gupta, A. K., *Response Spectrum Method in Seismic Analysis and Design of Structures,* Blackwell Scientific Publications, 1990.

Gurson, A. L., "Continuum Theory of Ductile Rupture by Void Nucleation and Growth: Part I—Yield Criteria and Flow Rules for Porous Ductile Materials," Journal of Engineering Materials and Technology, vol. 99, pp. 2–15, 1977.

Hansen, H. T., *Nonlinear Static and Dynamic Analysis of Slender Structures Subjected to Hydrodynamic Loading,* University of Trondheim, Norway, 1988.

Hayashi, K., and S. Nemat-Nasser, "Energy-Release Rate and Crack Kinking under Combined Loading," Journal of Applied Mechanics, vol. 48, pp. 520–524, 1981.

He, M. -Y., and J. W. Hutchinson, "Kinking of a Crack out of an Interface: Tabulated Solution Coefficients," Harvard University, Cambridge, Massachusetts, Division of Applied Mechanics, 1989.

Hegemier, G. A., "Evaluation of Models for MX Siting, Volume II—Reinforced Concrete Models," Systems, Science and Software, Report SSS–R–80–4155, La Jolla, California, 1979.

Hegemier, G. A., and K. J. Cheverton, "Evaluation of Reinforced Concrete Models for Nuclear Power Plant Application," EPRI, Palo Alto, California, 1980.

Hibbitt, H. D., "Some Follower Forces and Load Stiffness," International Journal for Numerical Methods in Engineering, vol. 14, pp. 937–941, 1979.

Hibbitt, H. D., "Special Structural Elements of Piping Analysis," *Pressure Vessels and Piping; Analysis and Computers*, ASME, New York, 1974.

Hibbitt, H. D., and B. I. Karlsson, "Analysis of Pipe Whip," EPRI, Report NP–1208, 1979.

Hilber, H. M., T. J. R. Hughes, and R. L. Taylor, "Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics," Earthquake Engineering and Structural Dynamics, vol. 5, pp. 283–292, 1977.

Hilber, H. M., and T. J. R. Hughes, "Collocation, Dissipation and 'Overshoot' for Time Integration Schemes in Structural Dynamics," Earthquake Engineering and Structural Dynamics, vol. 6, pp. 99–117, 1978.

Hilderbrand, F. B., *Introduction to Numerical Analysis,* McGraw Hill, 1974.

Hillerborg, A., M. Modeer, and P. E. Petersson, "Analysis of Crack Formation and Crack Growth in Concrete by Means of Fracture Mechanics and Finite Elements," Cement and Concrete Research, vol. 6, pp. 773–782, 1976.

Hjelm, H. E., "Elastoplasticity of Grey Cast Iron, FE-Algorithms and Biaxial Experiments," Ph.D. Thesis, Chalmers University of Technology, Division of Solid Mechanics, Sweden, 1992.

Hjelm, H. E., "Yield Surface for Grey Cast Iron under Biaxial Stress," Journal of Engineering Materials and Technology, vol. 116, pp. 148–154, 1994.

Hoenig, A., "Near-Tip Behavior of a Crack in a Plane Anisotropic Elastic Body," Engineering Fracture Mechanics, vol. 16, pp. 393–403, 1982.

Holman, J. P., *Heat Transfer,* McGraw-Hill, 1990.

Holzapfel, G. A., T. C. Gasser, and R. W. Ogden, "A New Constitutive Framework for Arterial Wall Mechanics and a Comparative Study of Material Models," Journal of Elasticity, vol. 61, pp. 1–48, 2000.

Hottel, H. C., and A. D. Sarofim, *Radiative Transfer,* McGraw-Hill, 1967.

Hughes, T. J. R., and A. Brooks, "A Theoretical Framework for Petrov-Galerkin Methods with Discontinuous Weighting Functions" in *Finite Elements in Fluids*, vol. 4, edited by R. H. Gallagher, D. H. Norrie, J. T. Oden, and O. C. Zienkiewicz, John Wiley & Sons, 1982.

Hughes, T. J. R., and W. K. Liu, "Nonlinear Finite Element Analysis of Shells, Part I: Three-Dimensional Shells," California Institute of Technology, Pasadena, California, 1980.

Hughes, T. J. R., R. L. Taylor, and W. Kanoknukulchai, "A Simple and Efficient Element for Plate Bending," International Journal for Numerical Methods in Engineering, vol. 11, no. 10, pp. 1529–1543, 1977.

Hughes, T. J. R., and T. E. Tezduyar, "Finite Elements based upon Mindlin Plate Theory with Particular Reference to the Four-Node Bilinear Isoparametric Element," Journal of Applied Mechanics, pp. 587–596, 1981.

Hughes, T. J. R., and J. Winget, "Finite Rotation Effects in Numerical Integration of Rate Constitutive Equations Arising in Large Deformation Analysis," International Journal for Numerical Methods in Engineering, vol. 15, pp. 1862–1867, 1980.

Humphrey, J. D., "Mechanics of Arterial Wall: Review and Directions," Critical Reviews in Biomedical Engineering, vol. 23, pp. 1–162, 1995.

Hurty, W. C., and M. F. Rubinstein, *Dynamics of Structures,* Prentice-Hall, Englewood Cliffs, New Jersey, 1964.

Hussain, M. A., S. L. Pu, and J. Underwood, "Strain-Energy-Release Rate for a Crack under Combined Mode I and Mode II," ASTM-STP-560, pp. 2–28, 1974.

Hutchinson, J. W., "Singular Behaviour at the End of a Tensile Crack in a Hardening Material," Journal of the Mechanics and Physics of Solids, vol. 16, pp. 13–31, 1968.

Hutchinson, J. W., and Z. Suo, "Mixed Mode Cracking in Layered Materials," Advances in Applied Mechanics, vol. 29, pp. 63–191, 1992.

Irons, B., and S. Ahmad, *Techniques of Finite Elements,* Ellis Horwood Limited, Halsted Press, John Wiley and Sons, Chichester, England, 1980.

Ikeda, T., *Fundamentals of Piezoelectricity,* Oxford University Press, New York, 1990.

Johnson, D., "Surface to Surface Radiation in the Program TAU, Taking Account of Multiple Reflection," United Kingdom Atomic Energy Authority Report ND-R-1444(R), 1987.

Kaliske, M., and H. Rothert, "On the Finite Element Implementation of Rubber-like Materials at Finite Strains," Engineering Computations, vol. 14, no. 2, pp. 216–232, 1997.

Kawabata, S., Y. Yamashita, H. Ooyama, and S. Yoshida, "Mechanism of Carbon-Black Reinforcement of Rubber Vulcanizate," Rubber Chemistry and Technology, vol. 68, no. 2, pp. 311–329, 1995.

Der Kiureghian, A., "A Response Spectrum Method For Random Vibration Analysis of MDF Systems," Earthquake Engineering and Structural Dynamics, vol. 9, pp. 419–435, 1981.

Kennedy, J. M., T. Belytschko, and J. I. Lin, "Recent Developments in Explicit Finite Element Techniques and Their Applications to Reactor Structures," Nuclear Engineering and Design, vol. 96, pp. 1–24, 1986.

Kfouri, A. P., "Some Evaluations of the Elastic T-Term Using Eshelby's Method," International Journal of Fracture, vol. 30, pp. 301–315, 1986.

Kilian, H.-G., "Equation of State of Real Networks," Polymer, vol. 22, pp. 209–216, 1981.

Kilian, H.-G., H. F. Enderle, and K. Unseld, "The Use of the van der Waals Model to Elucidate Universal Aspects of Structure-Property Relationships in Simply Extended Dry and Swollen Rubbers," Colloid and Polymer Science, vol. 264, pp. 866–876, 1986.

Kleiber, M., H. Antúnez, T. D. Hien, and P. Kowalczyk, *Parameter Sensitivity in Nonlinear Mechanics,* John Wiley & Sons, 1997.

Kroenke, W. C., "Classification of Finite Element Stresses According to ASME Section III Stress Categories," ASME 94th Winter Annual Meeting, November 1973.

Kroenke, W. C., G. W. Addicott, and B. M. Hinton, "Interpretation of Finite Element Stresses According to ASME Section III," Second National Congress on Pressure Vessels and Piping, June 1975.

Kumar, V., M. D. German, and C. F. Shih, "An Engineering Approach for Elastic-Plastic Fracture Analysis," Electric Power Research Institute Report NP-1931, Project 1237-1, Electric Power Research Institute, Palo Alto, CA, 1981.

Kwata, K., S. Asai, and H. Takeda, "A Solution for the IAEA International Piping Benchmark Problem," Century Research Corporation, Tokyo, 1978.

Lamb, H., *Hydrodynamics,* Dover Publications, New York, 1945.

Larsson, S. G., and A. J. Carlsson, "Influence of Non-Singular Stress Terms and Specimen Geometry on Small-Scale Yielding at Crack Tips in Elastic-Plastic Material," Journal of the Mechanics and Physics of Solids, vol. 21, pp. 263–278, 1973.

Lee, J., and G. L. Fenves, "Plastic-Damage Model for Cyclic Loading of Concrete Structures," Journal of Engineering Mechanics, vol. 124, no. 8, pp. 892–900, 1998.

Lemaitre, J., and J.-L. Chaboche, *Mechanics of Solid Materials,* Cambridge University Press, 1990.

Leppington, F., "Inverse Scattering" in *Modern Methods in Analytical Acoustics,* edited by D. G. Crighton, et al., Springer Verlag, pp. 551–564, 1992.

Lianis, G., "Small Deformations Superposed on an Initial Large Deformation in Viscoelastic Bodies," Proceedings of the 4th International Congress on Rheology, part 2, pp. 109–119, Interscience, New York, 1965.

Lin, C. S., and A. C. Scordelis, "Nonlinear Analysis of RC Shells of General Form," ASCE, Journal of Structural Engineering, vol. 101, pp. 523–538, 1975.

Lindholm, J. S., and R. L. Besseny, "A Survey of Rate Dependent Strength Properties of Metals," Air Force Materials Laboratory, Report AFML–TR–69–119, 1969.

Lubliner, J., J. Oliver, S. Oller, and E. Oñate, "A Plastic-Damage Model for Concrete," International Journal of Solids and Structures, vol. 25, no. 3, pp. 229–326, 1989.

Lysmer, J., and R. L. Kuhlemeyer, "Finite Dynamic Model for Infinite Media," Journal of the Engineering Mechanics Division of the ASCE, pp. 859–877, August 1969.

MacNeal, R. H., "A Simple Quadrilateral Shell Element," Computers and Structures, vol. 8, pp. 175–183, 1978.

MacNeal, R. H., "Derivation of Element Stiffness Matrices by Assumed Strain Distributions," Nuclear Engineering and Design, vol. 70, pp. 3–12, 1982.

Maiti, S. K., L. J. Gibson, and M. F. Ashby, "Deformation and Energy Absorption Diagrams for Cellular Solids," Acta Metallurgica, vol. 32, no. 11, pp. 1963–1975, 1984.

Maiti, S. K., and R. A. Smith, "Comparison of the Criteria for Mixed Mode Brittle Fracture Based on the Preinstability Stress-Strain Field, Part I: Slit and Elliptical Cracks under Uniaxial Tensile Loading," International Journal of Fracture, vol. 23, pp. 281–295, 1983.

Mang, H. A., "Symmetricability of Pressure Stiffness Matrices for Shells with Loaded Free Edges," International Journal for Numerical Methods in Engineering, vol. 15, pp. 981–990, 1980.

Matthies, H., and G. Strang, "The Solution of Nonlinear Finite Element Equations," International Journal for Numerical Methods in Engineering, vol. 14, pp. 1613–1626, 1979.

Menétrey, Ph., and K. J. William, "Triaxial Failure Criterion for Concrete and its Generalization," ACI Structural Journal, 92:311–18, May/June, 1995.

Mills-Curran, W. C., "Calculation of Eigenvector Derivatives for Structures with Repeated Eigenvalues," AIAA Journal, vol. 26, no. 7, pp. 867–871, 1988.

Mitalas, G. P., and D. G. Stephenson, "FORTRAN IV Programs to Calculate Radiant Interchange Factors," National Research Council of Canada, Division of Building Research, DBR-25, 1966.

Morman, K. N., "Analytical Prediction of Elastomeric Mount Dynamic Characteristics," Report L2143-2, Ford Engineering Computer Systems, Car Engineering, Dearborn, Michigan, 1979.

Morman, K. N., and J. C. Nagtegaal, "Finite Element Analysis of Sinusoidal Small-Amplitude Vibrations in Deformed Viscoelastic Solids. Part I: Theoretical Development," International Journal of Numerical Methods in Engineering, vol. 19, pp. 1079–1103, 1983.

Morse, P., and K. Ingard, *Theoretical Acoustics,* McGraw-Hill, 1968.

Mullins, L., "Effect of Stretching on the Properties of Rubber," Journal of Rubber Research, vol. 16, pp. 275–289, 1947.

Murff, J. D., Private communication, 1982.

Murthy, D. V., and R. T. Haftka, "Derivatives of Eigenvalues and Eigenvectors of a General Complex Matrix," International Journal of Numerical Methods in Engineering, vol. 26, pp. 293–311, 1988.

Nagtegaal, J. C., D. M. Parks, and J. R. Rice, "On Numerically Accurate Finite Element Solutions in the Fully Plastic Range," Computer Methods in Applied Mechanics and Engineering, vol. 4, pp. 153–177, 1977.

Nakamura, T., and D. M. Parks, "Antisymmetrical 3-D Stress Field near the Crack Front of a Thin Elastic Plate," International Journal of Solids and Structures, vol. 25, pp. 1411–1426, 1989.

Nakamura, T., and D. M. Parks, "Determination of Elastic T-Stress along Three-dimensional Crack Fronts Using an Interaction Integral," International Journal of Solids and Structures, vol. 28, pp. 1597–1611, 1992.

Nayak, G. C., and O. C. Zienkiewicz, "A Convenient Form of Invariants and its Application to Plasticity," Proceedings of the ASCE, Engineering Mechanics Division, vol. 98, no. St4, pp. 949–954, 1972.

Needleman, A., and V. Tvergaard, "Necking of Biaxially Stretched Elastic-Plastic Circular Plates," Journal of the Mechanics and Physics of Solids, vol. 25, pp. 159–183, 1977.

Newman, M., and A. Pipano, "Fast Modal Extraction in NASTRAN via the FEER Computer Program," NASA TM X-2893, pp. 485–506, 1973.

Newmark, N. M., J. A. Blume, and K. K. Kapur, "Seismic Design Spectra for Nuclear Power Plants," Power Division Journal, ASCE, 99, P02, Proceedings Paper 10142, pp. 287–303, 1973.

Nguyen, H. V., and D. F. Durso, "Absorption of Water by Fiber Webs: an Illustration of Diffusion Transport," Tappi Journal, vol. 66, no. 12, 1983.

Nishimura, H., M. Isobe, and K. Horikawa, “Higher Order Solutions of Stokes and Cnoidal Waves,” Journal of Faculty Engineering, University of Tokyo (B), vol. 34, no. 2, 1977.

Nuclear Standard NE F 9-5T, “Guidelines and Procedures for Design of Class 1 Elevated Temperature Nuclear System Components,” United States Department of Energy, Nuclear Energy Programs, March 1981.

Oden, J. T., *Mechanics of Elastic Structures,* McGraw-Hill, New York, 1967.

Oden, J. T., and N. Kikuchi, “Finite Element Methods for Constrained Problems in Elasticity,” TICOM, University of Texas at Austin, Austin, Texas, 1980.

Oden, J. T., and N. Kikuchi, “Finite Element Methods for Constrained Problems in Elasticity,” Fifth Invitational Symposium on the Unification of Finite Elements—Finite Differences and Calculus of Variations, Edited by H. Kardestuncer, University of Connecticut, Storrs, CT.

Oden, J. T., and E. B. Pires, “Nonlocal and Nonlinear Friction Laws and Variational Principles for Contact Problems in Elasticity,” Journal of Applied Mechanics, vol. 50, pp. 67–73, 1983.

Ogden, R. W., and D. G. Roxburgh, “A Pseudo-Elastic Model for the Mullins Effect in Filled Rubber,” Proceedings of the Royal Society of London, Series A, vol. 455, pp. 2861–2877, 1999.

Ohayon, R., “Fluid-Structure Modal Analysis. New Symmetric Continuum-Based Formulations. Finite Element Applications,” *Proceedings of the International Conference on Advances in Numerical Methods in Engineering: Theory and Applications*, Edited by G. N. Pande and J. Middleton, Martinus Nijhoff, 1987.

Ohtsubo, H., and O. Watanabe, “Flexibility and Stress Factors for Pipe Bends—An Analysis by the Finite Ring Method,” ASME paper 76–PVP–40, 1976.

Palaniswamy, K., and W. G. Knauss, “On the Problem of Crack Extension in Brittle Solids under General Loading,” *Mechanics Today*, Edited by S. Nemat-Nasser, Vol. 4, Pergamon Press, 1978.

Papoulis, A., *Probability, Random Variables, and Stochastic Processes,* McGraw-Hill, New York, 1965.

Parks, D. M., “The Virtual Crack Extension Method for Nonlinear Material Behavior,” Computer Methods in Applied Mechanics and Engineering, vol. 12, pp. 353–364, 1977.

Parks, D. M., “Advances in Characterization of Elastic-Plastic Crack-Tip Fields,” *Topics in Fracture and Fatigue*, Edited by A. S. Argon, Springer Verlag, 1992.

Parks, D. M., and C. S. White, “Elastic-Plastic Line Spring Finite Elements for Surface Cracked Plates and Shells,” Journal of Pressure Vessel Technology, vol. 104, pp. 287–292, 1982.

Parlett, B. N., *The Symmetric Eigenvalue Problem,* Prentice-Hall, Englewood Cliffs, New Jersey, 1980.

Parlett, B. N., and B. Nour-Omid, “Toward a Black Box Lanczos Program,” Computer Physics Communications, vol. 53, pp. 169–179, 1989.

Parry, R. H. G., Editor, *Stress-Strain Behaviour of Soils,* G. T. Foulis and Co., Henley, England, 1972.

Penzien, J., and M. Watabe, “Characteristic of 3-Dimensional Earthquake Ground Motions,” Earthquake Engineering and Structural Dynamics, vol. 3, pp. 365–373, 1975.

Powell, G., and J. Simons, “Improved Iterative Strategy for Nonlinear Structures,” International Journal for Numerical Methods in Engineering, vol. 17, pp. 1455–1467, 1981.

Puso, M. A., “A Highly Efficient Enhanced Assumed Strain Physically Stabilized Hexahedral Element,” International Journal for Numerical Methods in Engineering, vol. 49, pp. 1029–1064, 2000.

Ramaswami, S., “Towards Optimal Solution Techniques for Large Eigenproblems in Structural Mechanics,” Ph.D. Thesis, MIT, 1979.

Ramm, E., “Strategies for Tracing the Nonlinear Response Near Limit Points,” *Nonlinear Finite Element Analysis in Structural Mechanics*, Edited by E. Wunderlich, E. Stein, and K. J. Bathe, Springer-Verlag, Berlin, 1981.

Rice, J. R., “The Line Spring Model for Surface Flaws,” *The Surface Crack: Physical Problems and Computational Solutions*, J. L. Sedlow, Editor, ASME, 1972.

Rice, J. R., “Continuum Mechanics and Thermodynamics of Plasticity in Relation to Microscale Deformation Mechanisms,” *Constitutive Equations in Plasticity*, Argon, A. S., Editor, MIT Press, Cambridge, Massachusetts, 1975.

Rice, J. R., and G. F. Rosengren, “Plane Strain Deformation Near a Crack Tip in a Power Law Hardening Material,” Journal of the Mechanics and Physics of Solids, vol. 16, pp. 1–12, 1968.

Rodal, J. J. A., and E. A. Witmer, “Finite-Strain Large-Deflection Elastic-Viscoplastic Finite-Element Transient Response Analysis of Structures,” NASA contractor, Report 159874, NASA Lewis, 1979.

Rots, J. G., and J. Blaauwendraad, “Crack Models for Concrete: Discrete or Smeared? Fixed, Multi-directional or Rotating?,” HERON, vol. 34, no. 1, pp. 1–59, 1989.

Sandler, I. S., “A New Computational Procedure for Wave Propagation Problems and a New Procedure for Non-Reflection Boundaries,” Computer Methods in Applied Mechanics and Engineering, vol. 164, pp. 223–233, 1998.

Scheller, J. D., and R. Mallet, “Numerical Evaluation of an Inelastic Piping Elbow Element,” ASME Paper 79–PVP–41, 1979.

Schofield, A., and C. P. Wroth, *Critical State Soil Mechanics,* McGraw-Hill, New York, 1968.

Schreyer, H. L., R. F. Kulak, and J. M. Kramer, “Accurate Numerical Solutions for Elastic-Plastic Models,” Transactions of the ASME, Journal of Pressure Vessel Technology, vol. 101, no. 3, 1979.

Segalman, D. J., C. W. G. Fulcher, G. M. Reese, and R. V. Field, Jr., “An Efficient Method for Calculating RMS von Mises Stress in a Random Vibration Environment,” Sandia Report, SAND98-0260, 1998.

Shaw, J., and S. Jayasuriya, “Modal Sensitivities for Repeated Eigenvalues and Eigenvalue Derivatives,” AIAA Journal, vol. 30, no. 3, pp. 850–852, 1991.

Shih, C. F., and R. J. Asaro, “Elastic-Plastic Analysis of Cracks on Bimaterial Interfaces: Part I—Small Scale Yielding,” Journal of Applied Mechanics, pp. 299–316, 1988.

Shih, C. F., B. Moran, and T. Nakamura, “Energy Release Rate along a Three-Dimensional Crack Front in a Thermally Stressed Body,” International Journal of Fracture, vol. 30, pp. 79–102, 1986.

Siegel, R., and J. R. Howell, *Thermal Radiation Heat Transfer,* McGraw Hill, New York, 1980.

Sih, G. C., "Some Basic Problems in Fracture Mechanics and New Concepts," Engineering Fracture Mechanics, vol. 5, pp. 365–377, 1973.

Sih, G. C., P. C. Paris, and G. R. Irwin, "On Cracks in Rectilinearly Anisotropic Bodies," International Journal of Fracture Mechanics, vol. 1, pp. 189–202, 1965.

Simo, J. C., "On a Fully Three-Dimensional Finite-Strain Viscoelastic Damage Model: Formulation and Computational Aspects," Computer Methods in Applied Mechanics and Engineering, vol. 60, pp. 153–173, 1987.

Simo, J. C., "(Symmetric) Hessian for Geometrically Nonlinear Models in Solid Mechanics: Intrinsic Definition and Geometric Interpretation," Computer Methods in Applied Mechanics and Engineering, vol. 96, no. 2, pp. 189–200, 1992.

Simo, J. C., "Algorithms for Static and Dynamic Multiplicative Plasticity that Preserve the Classical Return Mapping Schemes of the Infinitesimal Theory," Computer Methods in Applied Mechanics and Engineering, vol. 99, pp. 61–112, 1992.

Simo, J. C., and F. Armero, "Geometrically Nonlinear Enhanced Strain Mixed Methods and the Method of Incompatible Modes," International Journal for Numerical Methods in Engineering, vol. 33, pp. 1413–1449, 1992.

Simo, J. C., D. D. Fox, and M. S. Rifai, "On a Stress Resultant Geometrically Exact Shell Model. Part III: Computational Aspects of the Nonlinear Theory," Computer Methods in Applied Mechanics and Engineering, vol. 79, pp. 21–70, 1989.

Simo, J. C., and M. S. Rifai, "A Class of Assumed Strain Methods and the Method of Incompatible Modes," International Journal for Numerical Methods in Engineering, vol. 29, pp. 1595–1638, 1990.

Simon, H. D., "The Lanczos Algorithm with Partial Reorthogonalization," Mathematics of Computation, vol. 42, pp. 115–142, 1984.

Skjelbreia, L., and J. Hendrickson, "Fifth Order Gravity Wave Theory," Proceedings of the 87th Coastal England Conference, Chapter Ten, den Haag, 1960.

Smeby, W., and A. Der Kiureghian, "Modal Combination Rules for Multicomponent Earthquake Excitation," Earthquake Engineering and Structural Dynamics, vol. 12, pp. 1–12, 1984.

Sobel, L. H., "In-Plane Bending of Elbows," Computers and Structures, vol. 7, pp. 701–715, 1977.

Sofronis, P., and R. M. McMeeking, "Numerical Analysis of Hydrogen Transport Near a Blunting Crack Tip," Journal of the Mechanics and Physics of Solids, vol. 37, no. 3, pp. 317–350, 1989.

Spence, J., "An Upper Bound Analysis for the Deformation of Smooth Pipe Bends in Creep," Second IUTAM Symposium on Creep in Structures, Goteberg, 1970, Springer Verlag, 1972.

Spencer, A. J. M., "Constitutive Theory for Strongly Anisotropic Solids," A. J. M. Spencer (ed.), Continuum Theory of the Mechanics of Fibre-Reinforced Composites, CISM Courses and Lectures No. 282, International Centre for Mechanical Sciences, Springer-Verlag, Wien, pp. 1–32, 1984.

Spring, K. W., "Euler Parameters and the Use of Quaternion Algebra in the Manipulation of Finite Rotations: a Review," Mechanism and Machine Theory, vol. 21, pp. 365–373, 1986.

Storåkers, B., "On Material Representation and Constitutive Branching in Finite Compressible Elasticity," Journal of the Mechanics and Physics of Solids, vol. 34, no. 2, pp. 125–145, 1986.

Strang, G., *Linear Algebra and Its Applications,* Academic Press, New York, 1976.

Strang, G., and G. Fix, *An Analysis of the Finite Element Method,* Prentice-Hall, New York, 1973.

Stroh, A. N., "Dislocation and Cracks in Anisotropic Elasticity," Philosophical Magazine, vol. 7, pp. 625–646, 1958.

Stroud, A. H., *Approximate Calculation of Multiple Integrals,* Prentice-Hall, 1971.

Suo, Z., "Singularities, Interfaces and Cracks in Dissimilar Anisotropic Media," Proceedings of the Royal Society, London, A, vol. 427, pp. 331–358, 1990.

Sutter, T. R., C. J. Camarda, J. L. Walsh, and H. M. Adelman, "Comparison of Several Methods for Calculating Vibration Mode Shape Derivatives," AIAA Journal, vol. 26, no. 12, pp. 1506–1511, 1988.

Symonds, P. S., "Survey of Methods of Analysis for Plastic Deformation of Structures under Dynamic Loading," Brown University, Division of Engineering Report, June 1967.

Symonds, P. S., T. C. T. Ting, and D. N. Robinson, "Survey of Progress in Plastic Wave Propagation in Solid Bodies," Brown University, Division of Engineering Report, June 1967.

Tada, H., C. Paris, and G. R. Irwin, *The Stress Analysis of Cracks Handbook,* Del Research Corporation, Hellertown, Pennsylvania, 1973.

Takeda, H., S. Asai, and K. Kwata, "A New Finite Element for Structural Analysis of Piping Systems," Proceedings of the Fifth SMIRT Conference, Berlin, 1979.

Tanaka, T., and D. J. Fillmore, "Kinetics of Swelling of Gels," Journal of Chemical Physics, vol. 70, pp. 1214–1218, 1979.

Tariq, S. M., "Evaluation of Flow Characteristics of Perforations Including Nonlinear Effects With the Finite Element Method," SPE Production Engineering, pp. 104–112, May 1987.

Taylor, R. L., P. J. Beresford, and E. L. Wilson, "A Nonconforming Element for Stress Analysis," International Journal for Numerical Methods in Engineering, vol. 10, pp. 1211–1219, 1976.

Theocaris, P. S., "A Higher Order Approximation for the T-Criterion of Fracture in Biaxial Fields," Engineering Fracture Mechanics, vol. 19, pp. 975–991, 1984.

Thompson, C. J., *Classical Equilibrium Statistical Mechanics,* Oxford University Press, New York, 1988.

Timoshenko, S. P., *Strength of Materials; Part III, Advanced Theory and Problems,* D. Van Nostrand Co., Princeton, New Jersey, Third Edition edition, 1956.

Treloar, L. R. G., *The Physics of Rubber Elasticity,* Clarendon Press, Oxford, Third Edition edition, 1975.

Tvergaard, V., "Influence of Voids on Shear Band Instabilities under Plane Strain Condition," International Journal of Fracture Mechanics, vol. 17, pp. 389–407, 1981.

Twizell, E. H., and R. W. Ogden, "Non-linear Optimization of the Material Constants in Ogden's Stress-Deformation Function for Incompressible Isotropic Elastomers," Journal of the Australian Mathematical Society, Series B 24, pp. 424–434, 1986.

Wang, B. P., "Improved Approximate Methods for Computing Eigenvector Derivatives in Structural Dynamics," AIAA Journal, vol. 29, no. 6, pp. 1018–1020, 1991.

Wang, Y. -Y., "A Two-Parameter Characterization of Elastic-Plastic Crack Tip Fields and Application to Cleavage Fracture," Ph.D. Thesis, Department of Mechanical Engineering, MIT, Cambridge, MA, 1991.

Weber, G., and L. Anand, "Finite Deformation Constitutive Equations and a Time Integration Procedure for Isotropic, Hyperelastic-Viscoplastic Solids," Computer Methods in Applied Mechanics and Engineering, vol. 79, pp. 173–202, 1990.

Whitham, G. B., *Linear and Nonlinear Waves,* John Wiley & Sons, 1974.

Wilkinson, J. H., *The Algebraic Eigenvalue Problem,* Oxford University Press, Oxford, 1965.

Williams, M. L., "On the Stress Distribution at the Base of a Stationary Crack," Journal of Applied Mechanics, vol. 24, pp. 109–114, 1957.

Wilson, E. L., A. Der Kiureghian, and E. P. Bayo, "A Replacement for the SRSS Method in Seismic Analysis," Earthquake Engineering and Structural Dynamics, vol. 9, pp. 187–192, 1981.

Wilson, E. L., R. L. Taylor, W. P. Doherty, and J. Ghaboussi, "Incompatible Displacement Models" in *Numerical and Computer Models in Structural Mechanics*, Eds. S. F. Fenves, N. Perrone, A. R. Robinson, and W. C. Schnobrich, Academic Press, New York, 1973.

Wu, T. H., *Soil Mechanics,* Allyn and Bacon, Boston, 1976.

Xu, G., F. Bower, and M. Ortiz, "An Analysis of Non-Planar Crack Growth under Mixed Mode Loading," International Journal of Solids and Structures, vol. 31, pp. 2167–2193, 1994.

Yeoh, O. H., "Some Forms of the Strain Energy Function for Rubber," Rubber Chemistry and Technology, vol. 66, pp. 754–771, 1993.

Yu, C. C., and J. C. Heinrich, "Petrov-Galerkin Methods for the Time-Dependent Convective Transport Equation," International Journal for Numerical Methods in Engineering, vol. 23, pp. 883–902, 1986.

Yu, C. C., and J. C. Heinrich, "Petrov-Galerkin Method for Multidimensional, Time-Dependent, Convective-Diffusive Equations," International Journal for Numerical Methods in Engineering, vol. 24, pp. 2201–2215, 1987.

Zhong, Z. H., "Contact Problems with Friction," Proceedings of Numiform 89, pp. 599–606, Balkema, Rotterdam, 1989.

Zienkiewicz, O. C., *The Finite Element Method,* McGraw-Hill, London, Third Edition edition, 1977.

Zienkiewicz, O. C., C. Emson, and P. Bettess, "A Novel Boundary Infinite Element," International Journal for Numerical Methods in Engineering, vol. 19, pp. 393–404, 1983.

Zienkiewicz, O. C., and R. Newton, "Coupled Vibrations of a Structure Submerged in a Compressible Fluid," Proceedings of the International Symposium on Finite Element Techniques, Stuttgart, 1969.

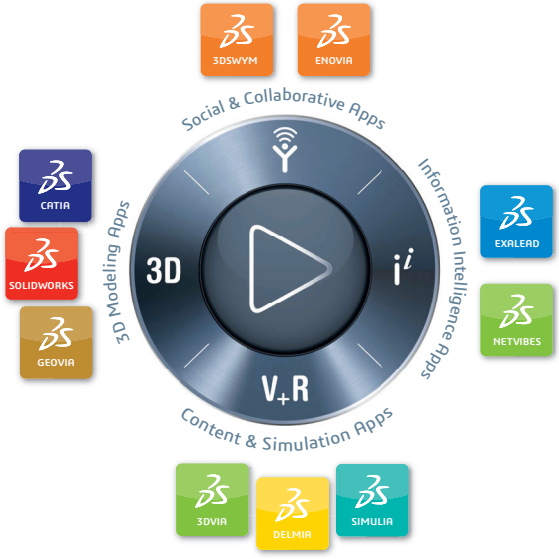
Zienkiewicz, O. C., and G. N. Pande, “Time Dependent Multilaminate Model of Rocks—A Numerical Study of Deformation and Failure of Rock Masses,” International Journal for Numerical and Analytical Methods in Geomechanics, vol. 1, pp. 219–247, 1977.

About SIMULIA

Dassault Systèmes SIMULIA applications, including Abaqus, Isight, Tosca, and Simulation Lifecycle Management, enable users to leverage physics-based simulation and high-performance computing to explore real-world behavior of products, nature, and life. As an integral part of Dassault Systèmes' **3D**EXPERIENCE platform, SIMULIA applications accelerate the process of making highly informed, mission-critical design and engineering decisions before committing to costly and time-consuming physical prototypes. **www.3ds.com/simulia**

Our **3D**EXPERIENCE® Platform powers our brand applications, serving 12 industries, and provides a rich portfolio of industry solution experiences.

Dassault Systèmes, the **3D**EXPERIENCE Company, provides business and people with virtual universes to imagine sustainable innovations. Its world-leading solutions transform the way products are designed, produced, and supported. Dassault Systèmes' collaborative solutions foster social innovation, expanding possibilities for the virtual world to improve the real world. The group brings value to over 170,000 customers of all sizes in all industries in more than 140 countries. For more information, visit **www.3ds.com**.



© Dassault Systèmes 2014, all rights reserved. 3DEXPERIENCE®, CATIA, SolidWorks, ENOVIA, SIMULIA, DELMIA, 3DVIA, 3DSwym, EXALEAD, and Netvibes are registered trademarks of Dassault Systèmes or its subsidiaries in the US and/or other countries.

DASSAULT SYSTEMES | The **3D**EXPERIENCE Company

Europe/Middle East/Africa
Dassault Systèmes
10, rue Marcel Dassault
CS 40501
78946 Vélizy-Villacoublay Cedex
France

Asia-Pacific
Dassault Systèmes
Pier City Shibaura Bldg 10F
3-18-1 Kaigan, Minato-Ku
Tokyo 108-002
Japan

Americas
Dassault Systèmes
175 Wyman Street
Waltham, Massachusetts
02451-1223
USA