DASSAULT SYSTEMES

SIMULIA

ABAQUS 6.14

EXAMPLE PROBLEMS GUIDE

VOLUME I: STATIC AND DYNAMIC ANALYSES



3DEXPERIENCE

Abaqus

Example Problems Guide

Volume I

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

Volume I

1. Static Stress/Displacement Analyses

Static and quasi-static stress analyses

Axisymmetric analysis of bolted pipe flange connections 1.1.1
Elastic-plastic collapse of a thin-walled elbow under in-plane bending and internal pressure 1.1.2
Parametric study of a linear elastic pipeline under in-plane bending 1.1.3
Indentation of an elastomeric foam specimen with a hemispherical punch 1.1.4
Collapse of a concrete slab 1.1.5
Jointed rock slope stability 1.1.6
Notched beam under cyclic loading 1.1.7
Uniaxial ratchetting under tension and compression 1.1.8
Hydrostatic fluid elements: modeling an airspring 1.1.9
Shell-to-solid submodeling and shell-to-solid coupling of a pipe joint 1.1.10
Stress-free element reactivation 1.1.11
Transient loading of a viscoelastic bushing 1.1.12
Indentation of a thick plate 1.1.13
Damage and failure of a laminated composite plate 1.1.14
Analysis of an automotive boot seal 1.1.15
Pressure penetration analysis of an air duct kiss seal 1.1.16
Self-contact in rubber/foam components: jounce bumper 1.1.17
Self-contact in rubber/foam components: rubber gasket 1.1.18
Submodeling of a stacked sheet metal assembly 1.1.19
Axisymmetric analysis of a threaded connection 1.1.20
Direct cyclic analysis of a cylinder head under cyclic thermal-mechanical loadings 1.1.21
Erosion of material (sand production) in an oil wellbore 1.1.22
Submodel stress analysis of pressure vessel closure hardware 1.1.23
Using a composite layup to model a yacht hull 1.1.24
Energy computations in a contact analysis 1.1.25

Buckling and collapse analyses

Snap-through buckling analysis of circular arches 1.2.1
Laminated composite shells: buckling of a cylindrical panel with a circular hole 1.2.2
Buckling of a column with spot welds 1.2.3
Elastic-plastic K-frame structure 1.2.4
Unstable static problem: reinforced plate under compressive loads 1.2.5
Buckling of an imperfection-sensitive cylindrical shell 1.2.6

Forming analyses

Upsetting of a cylindrical billet: quasi-static analysis with mesh-to-mesh solution mapping (Abaqus/Standard) and adaptive meshing (Abaqus/Explicit) 1.3.1
Superplastic forming of a rectangular box 1.3.2
Stretching of a thin sheet with a hemispherical punch 1.3.3
Deep drawing of a cylindrical cup 1.3.4
Extrusion of a cylindrical metal bar with frictional heat generation 1.3.5
Rolling of thick plates 1.3.6
Axisymmetric forming of a circular cup 1.3.7
Cup/trough forming 1.3.8
Forging with sinusoidal dies 1.3.9
Forging with multiple complex dies 1.3.10
Flat rolling: transient and steady-state 1.3.11
Section rolling 1.3.12
Ring rolling 1.3.13
Axisymmetric extrusion: transient and steady-state 1.3.14
Two-step forming simulation 1.3.15
Upsetting of a cylindrical billet: coupled temperature-displacement and adiabatic analysis 1.3.16
Unstable static problem: thermal forming of a metal sheet 1.3.17
Inertia welding simulation using Abaqus/Standard and Abaqus/CAE 1.3.18
Moldflow translation examples 1.3.19

Fracture and damage

A plate with a part-through crack: elastic line spring modeling 1.4.1
Contour integrals for a conical crack in a linear elastic infinite half space 1.4.2
Elastic-plastic line spring modeling of a finite length cylinder with a part-through axial flaw 1.4.3
Crack growth in a three-point bend specimen 1.4.4
Analysis of skin-stiffener debonding under tension 1.4.5
Failure of blunt notched fiber metal laminates 1.4.6
Debonding behavior of a double cantilever beam 1.4.7
Debonding behavior of a single leg bending specimen 1.4.8
Postbuckling and growth of delaminations in composite panels 1.4.9

Import analyses

Springback of two-dimensional draw bending 1.5.1
Deep drawing of a square box 1.5.2

2. Dynamic Stress/Displacement Analyses

Dynamic stress analyses

Nonlinear dynamic analysis of a structure with local inelastic collapse 2.1.1

Detroit Edison pipe whip experiment 2.1.2
Rigid projectile impacting eroding plate 2.1.3
Eroding projectile impacting eroding plate 2.1.4
Tennis racket and ball 2.1.5
Pressurized fuel tank with variable shell thickness 2.1.6
Modeling of an automobile suspension 2.1.7
Explosive pipe closure 2.1.8
Knee bolster impact with general contact 2.1.9
Crimp forming with general contact 2.1.10
Collapse of a stack of blocks with general contact 2.1.11
Cask drop with foam impact limiter 2.1.12
Oblique impact of a copper rod 2.1.13
Water sloshing in a baffled tank 2.1.14
Seismic analysis of a concrete gravity dam 2.1.15
Progressive failure analysis of thin-wall aluminum extrusion under quasi-static and dynamic loads 2.1.16
Impact analysis of a pawl-ratchet device 2.1.17
High-velocity impact of a ceramic target 2.1.18

Mode-based dynamic analyses

Analysis of a rotating fan using substructures and cyclic symmetry 2.2.1
Linear analysis of the Indian Point reactor feedwater line 2.2.2
Response spectra of a three-dimensional frame building 2.2.3
Brake squeal analysis 2.2.4
Dynamic analysis of antenna structure utilizing residual modes 2.2.5
Steady-state dynamic analysis of a vehicle body-in-white model 2.2.6

Eulerian analyses

Rivet forming 2.3.1
Impact of a water-filled bottle 2.3.2

Co-simulation analyses

Dynamic impact of a scooter with a bump 2.4.1

Volume II

3. Tire and Vehicle Analyses

Tire analyses

Symmetric results transfer for a static tire analysis	3.1.1
Steady-state rolling analysis of a tire	3.1.2
Subspace-based steady-state dynamic tire analysis	3.1.3
Steady-state dynamic analysis of a tire substructure	3.1.4
Coupled acoustic-structural analysis of a tire filled with air	3.1.5
Import of a steady-state rolling tire	3.1.6
Analysis of a solid disc with Mullins effect and permanent set	3.1.7
Tread wear simulation using adaptive meshing in Abaqus/Standard	3.1.8
Dynamic analysis of an air-filled tire with rolling transport effects	3.1.9
Acoustics in a circular duct with flow	3.1.10

Vehicle analyses

Inertia relief in a pick-up truck	3.2.1
Substructure analysis of a pick-up truck model	3.2.2
Display body analysis of a pick-up truck model	3.2.3
Continuum modeling of automotive spot welds	3.2.4

Occupant safety analyses

Seat belt analysis of a simplified crash dummy	3.3.1
Side curtain airbag impactor test	3.3.2

4. Mechanism Analyses

Resolving overconstraints in a multi-body mechanism model	4.1.1
Crank mechanism	4.1.2
Snubber-arm mechanism	4.1.3
Flap mechanism	4.1.4
Tail-skid mechanism	4.1.5
Cylinder-cam mechanism	4.1.6
Driveshaft mechanism	4.1.7
Geneva mechanism	4.1.8
Trailing edge flap mechanism	4.1.9
Substructure analysis of a one-piston engine model	4.1.10
Application of bushing connectors in the analysis of a three-point linkage	4.1.11
Gear assemblies	4.1.12

5. Heat Transfer and Thermal-Stress Analyses

Thermal-stress analysis of a disc brake	5.1.1

A sequentially coupled thermal-mechanical analysis of a disc brake with an Eulerian approach 5.1.2
Exhaust manifold assemblage 5.1.3
Coolant manifold cover gasketed joint 5.1.4
Conductive, convective, and radiative heat transfer in an exhaust manifold 5.1.5
Thermal-stress analysis of a reactor pressure vessel bolted closure 5.1.6

6. Fluid Dynamics and Fluid-Structure Interaction

Conjugate heat transfer analysis of a component-mounted electronic circuit board 6.1.1

7. Electromagnetic Analyses

Piezoelectric analyses

Eigenvalue analysis of a piezoelectric transducer 7.1.1
Transient dynamic nonlinear response of a piezoelectric transducer 7.1.2

Joule heating analyses

Thermal-electrical modeling of an automotive fuse 7.2.1

8. Mass Diffusion Analyses

Hydrogen diffusion in a vessel wall section 8.1.1
Diffusion toward an elastic crack tip 8.1.2

9. Acoustic and Shock Analyses

Fully and sequentially coupled acoustic-structural analysis of a muffler 9.1.1
Coupled acoustic-structural analysis of a speaker 9.1.2
Coupled acoustic-structural analysis of a speaker 9.1.3
Response of a submerged cylinder to an underwater explosion shock wave 9.1.4
Convergence studies for shock analyses using shell elements 9.1.5
UNDEX analysis of a detailed submarine model 9.1.6
Coupled acoustic-structural analysis of a pick-up truck 9.1.7
Long-duration response of a submerged cylinder to an underwater explosion 9.1.8
Deformation of a sandwich plate under CONWEP blast loading 9.1.9

10. Soils Analyses

Plane strain consolidation 10.1.1
Calculation of phreatic surface in an earth dam 10.1.2
Axisymmetric simulation of an oil well 10.1.3
Analysis of a pipeline buried in soil 10.1.4
Hydraulically induced fracture in a well bore 10.1.5
Permafrost thawing–pipeline interaction 10.1.6

11. Structural Optimization Analyses

Topology optimization analyses

Topology optimization of an automotive control arm 11.1.1

Shape optimization analyses

Shape optimization of a connecting rod 11.2.1

Sizing optimization analyses

Sizing optimization of a gear shift control holder 11.3.1
Sizing optimization of a car door 11.3.2

12. Abaqus/Aqua Analyses

Jack-up foundation analyses 12.1.1
Riser dynamics 12.1.2

13. Particle Methods Analyses

Discrete element method analyses

Mixing of granular media in a drum mixer 13.1.1

14. Design Sensitivity Analyses

Overview

Design sensitivity analysis: overview 14.1.1

Examples

Design sensitivity analysis of a composite centrifuge 14.2.1
Design sensitivities for tire inflation, footprint, and natural frequency analysis 14.2.2
Design sensitivity analysis of a windshield wiper 14.2.3
Design sensitivity analysis of a rubber bushing 14.2.4

15. Postprocessing of Abaqus Results

User postprocessing of Abaqus results files: overview 15.1.1
Joining data from multiple results files and converting file format: FJOIN 15.1.2
Calculation of principal stresses and strains and their directions: FPRIN 15.1.3
Creation of a perturbed mesh from original coordinate data and eigenvectors: FPERT 15.1.4
Output radiation view factors and facet areas: FRAD 15.1.5
Creation of a data file to facilitate the postprocessing of elbow element results: FELBOW 15.1.6
Translating Abaqus data to modal neutral file format for analysis in MSC.ADAMS 15.1.7

1.0 INTRODUCTION

This is the Example Problems Guide for Abaqus. It contains many solved examples that illustrate the use of the program for common types of problems. Some of the problems are quite difficult and require combinations of the capabilities in the code.

The problems have been chosen to serve two purposes: to verify the capabilities in Abaqus by exercising the code on nontrivial cases and to provide guidance to users who must work on a class of problems with which they are relatively unfamiliar. In each worked example the discussion in the guide states why the example is included and leads the reader through the standard approach to an analysis: element and mesh selection, material model, and a discussion of the results. Many of these problems are worked with different element types, mesh densities, and other variations.

Input data files for all of the analyses are included with the Abaqus release in compressed archive files. The **abaqus fetch** utility is used to extract these input files for use. For example, to fetch input file boltpipeflange_3d_cyclsym.inp, type

```
abaqus fetch job=boltpipeflange_3d_cyclsym.inp
```

Parametric study script (`.psf`) and user subroutine (`.f`) files can be fetched in the same manner. All files for a particular problem can be obtained by leaving off the file extension. The **abaqus fetch** utility is explained in detail in "Fetching sample input files," Section 3.2.16 of the Abaqus Analysis User's Guide.

It is sometimes useful to search the input files. The **findkeyword** utility is used to locate input files that contain user-specified input. This utility is defined in "Querying the keyword/problem database," Section 3.2.15 of the Abaqus Analysis User's Guide.

To reproduce the graphical representation of the solution reported in some of the examples, the output frequency used in the input files may need to be increased. For example, in "Linear analysis of the Indian Point reactor feedwater line," Section 2.2.2, the figures that appear in the guide can be obtained only if the solution is written to the results file every increment; that is, if the input files are changed to read

```
*NODE FILE, ..., FREQUENCY=1
```

instead of `FREQUENCY=100` as appears now.

In addition to the Abaqus Example Problems Guide, there are two other guides that contain worked problems. The Abaqus Benchmarks Guide contains benchmark problems (including the NAFEMS suite of test problems) and standard analyses used to evaluate the performance of Abaqus. The tests in this guide are multiple element tests of simple geometries or simplified versions of real problems. The Abaqus Verification Guide contains a large number of examples that are intended as elementary verification of the basic modeling capabilities.

The qualification process for new Abaqus releases includes running and verifying results for all problems in the Abaqus Example Problems Guide, the Abaqus Benchmarks Guide, and the Abaqus Verification Guide.

1. Static Stress/Displacement Analyses

- "Static and quasi-static stress analyses," Section 1.1
- "Buckling and collapse analyses," Section 1.2
- "Forming analyses," Section 1.3
- "Fracture and damage," Section 1.4
- "Import analyses," Section 1.5

1.1 Static and quasi-static stress analyses

- "Axisymmetric analysis of bolted pipe flange connections," Section 1.1.1
- "Elastic-plastic collapse of a thin-walled elbow under in-plane bending and internal pressure," Section 1.1.2
- "Parametric study of a linear elastic pipeline under in-plane bending," Section 1.1.3
- "Indentation of an elastomeric foam specimen with a hemispherical punch," Section 1.1.4
- "Collapse of a concrete slab," Section 1.1.5
- "Jointed rock slope stability," Section 1.1.6
- "Notched beam under cyclic loading," Section 1.1.7
- "Uniaxial ratchetting under tension and compression," Section 1.1.8
- "Hydrostatic fluid elements: modeling an airspring," Section 1.1.9
- "Shell-to-solid submodeling and shell-to-solid coupling of a pipe joint," Section 1.1.10
- "Stress-free element reactivation," Section 1.1.11
- "Transient loading of a viscoelastic bushing," Section 1.1.12
- "Indentation of a thick plate," Section 1.1.13
- "Damage and failure of a laminated composite plate," Section 1.1.14
- "Analysis of an automotive boot seal," Section 1.1.15
- "Pressure penetration analysis of an air duct kiss seal," Section 1.1.16
- "Self-contact in rubber/foam components: jounce bumper," Section 1.1.17
- "Self-contact in rubber/foam components: rubber gasket," Section 1.1.18
- "Submodeling of a stacked sheet metal assembly," Section 1.1.19
- "Axisymmetric analysis of a threaded connection," Section 1.1.20
- "Direct cyclic analysis of a cylinder head under cyclic thermal-mechanical loadings," Section 1.1.21
- "Erosion of material (sand production) in an oil wellbore," Section 1.1.22
- "Submodel stress analysis of pressure vessel closure hardware," Section 1.1.23
- "Using a composite layup to model a yacht hull," Section 1.1.24
- "Energy computations in a contact analysis," Section 1.1.25

1.1.1 AXISYMMETRIC ANALYSIS OF BOLTED PIPE FLANGE CONNECTIONS

Product: Abaqus/Standard

A bolted pipe flange connection is a common and important part of many piping systems. Such connections are typically composed of hubs of pipes, pipe flanges with bolt holes, sets of bolts and nuts, and a gasket. These components interact with each other in the tightening process and when operation loads such as internal pressure and temperature are applied. Experimental and numerical studies on different types of interaction among these components are frequently reported. The studies include analysis of the bolt-up procedure that yields uniform bolt stress (Bibel and Ezell, 1992), contact analysis of screw threads (Fukuoka, 1992; Chaaban and Muzzo, 1991), and full stress analysis of the entire pipe joint assembly (Sawa et al., 1991). To establish an optimal design, a full stress analysis determines factors such as the contact stresses that govern the sealing performance, the relationship between bolt force and internal pressure, the effective gasket seating width, and the bending moment produced in the bolts. This example shows how to perform such a design analysis by using an economical axisymmetric model and how to assess the accuracy of the axisymmetric solution by comparing the results to those obtained from a simulation using a three-dimensional segment model. In addition, several three-dimensional models that use multiple levels of substructures are analyzed to demonstrate the use of substructures with a large number of retained degrees of freedom. Finally, a three-dimensional model containing stiffness matrices is analyzed to demonstrate the use of the matrix input functionality.

Geometry and model

The bolted joint assembly being analyzed is depicted in Figure 1.1.1–1. The geometry and dimensions of the various parts are taken from Sawa et al. (1991), modified slightly to simplify the modeling. The inner wall radius of both the hub and the gasket is 25 mm. The outer wall radii of the pipe flange and the gasket are 82.5 mm and 52.5 mm, respectively. The thickness of the gasket is 2.5 mm. The pipe flange has eight bolt holes that are equally spaced in the pitch circle of radius 65 mm. The radius of the bolt hole is modified in this analysis to be the same as that of the bolt: 8 mm. The bolt head (bearing surface) is assumed to be circular, and its radius is 12 mm.

The Young's modulus is 206 GPa and the Poisson's ratio is 0.3 for both the bolt and the pipe hub/flange. The gasket is modeled with either solid continuum or gasket elements. When continuum elements are used, the gasket's Young's modulus, $\boldsymbol{E}$, equals 68.7 GPa and its Poisson's ratio, ν, equals 0.3.

When gasket elements are used, a linear gasket pressure/closure relationship is used with the effective "normal stiffness," S_n, equal to the material Young's modulus divided by the thickness so that $S_n = 27.48$ GPa/mm. Similarly a linear shear stress/shear motion relationship is used with an effective shear stiffness, S_t, equal to the material shear modulus divided by the thickness so that $S_t = 10.57$ GPa/mm. The membrane behavior is specified with a Young's modulus of 68.7 GPa and a Poisson's ratio of 0.3. Sticking contact conditions are assumed in all contact areas: between the bearing surface and the flange and between the gasket and the hub. Contact between the bolt shank and the bolt hole is ignored.

The finite element idealizations of the symmetric half of the pipe joint are shown in Figure 1.1.1–2 and Figure 1.1.1–3, corresponding to the axisymmetric and three-dimensional analyses, respectively. The mesh used for the axisymmetric analysis consists of a mesh for the pipe hub/flange and gasket and a separate mesh for the bolts. In Figure 1.1.1–2 the top figure shows the mesh of the pipe hub and flange, with the bolt hole area shown in a lighter shade; and the bottom figure shows the overall mesh with the gasket and the bolt in place.

For the axisymmetric model second-order elements with reduced integration, CAX8R, are used throughout the mesh of the pipe hub/flange. The gasket is modeled with either CAX8R solid continuum elements or GKAX6 gasket elements. Contact between the gasket and the pipe hub/flange is modeled with contact pairs between surfaces defined on the faces of elements in the contact region or between such element-based surfaces and node-based surfaces. In an axisymmetric analysis the bolts and the perforated flange must be modeled properly. The bolts are modeled as plane stress elements since they do not carry hoop stress. Second-order plane stress elements with reduced integration, CPS8R, are employed for this purpose. The contact surface definitions, which are associated with the faces of the elements, account for the plane stress condition automatically. To account for all eight bolts used in the joint, the combined cross-sectional areas of the shank and the head of the bolts must be calculated and redistributed to the bolt mesh appropriately using the area attributes for the solid elements. The contact area is adjusted automatically.

Figure 1.1.1–4 illustrates the cross-sectional views of the bolt head and the shank. Each plane stress element represents a volume that extends out of the ***x*–*y*** plane. For example, element ***A*** represents a volume calculated as $(H_A) \times (Area_A)$. Likewise, element ***B*** represents a volume calculated as $(H_B) \times (Area_B)$. The sectional area in the ***x*–*z*** plane pertaining to a given element can be calculated as

$$Area = 2\int_{X_1}^{X_2} [(R^2 - x^2)^{\frac{1}{2}}]dx = [x(R^2 - x^2)^{\frac{1}{2}} + R^2 \arcsin(\frac{x}{|R|})]\Big|_{X_1}^{X_2},$$

where ***R*** is the bolt head radius, R_{bolthead}, or the shank radius, R_{shank} (depending on the element location), and X_1 and X_2 are ***x***-coordinates of the left and right side of the given element, respectively.

If the sectional areas are divided by the respective element widths, W_A and W_B, we obtain representative element thicknesses. Multiplying each element thickness by eight (the number of bolts in the model) produces the thickness values that are found in the solid section definition.

Sectional areas that are associated with bolt head elements located on the model's contact surfaces are used to calculate the surface areas of the nodes used in defining the node-based surfaces of the model. Referring again to Figure 1.1.1–4, nodal contact areas for a single bolt are calculated as follows:

$$A_1 = \frac{A_C}{4}, \quad A_9 = \frac{A_F}{4},$$

$$A_2 = \frac{A_C}{2}, \quad A_4 = \frac{A_D}{2}, \quad A_6 = \frac{A_E}{2}, \quad A_8 = \frac{A_F}{2},$$

$$A_3 = (A_C + A_D)/4, \quad A_5 = (A_D + A_E)/4, \quad A_7 = (A_E + A_F)/4,$$

where A_1 through A_9 are contact areas that are associated with contact nodes 1–9 and A_c through A_F are sectional areas that are associated with bolt head elements ***C–F***. Multiplying the above areas by eight (the number of bolts in the model) provides the nodal contact areas found in the contact property definitions.

A common way of handling the presence of the bolt holes in the pipe flange in axisymmetric analyses is to smear the material properties used in the bolt hole area of the mesh and to use inhomogeneous material properties that correspond to a weaker material in this region. General guidelines for determining the effective material properties for perforated flat plates are found in ASME Section VIII Div 2 Article 4–9. For the type of structure under study, which is not a flat plate, a common approach to determining the effective material properties is to calculate the elasticity moduli reduction factor, which is the ratio of the ligament area in the pitch circle to the annular area of the pitch circle. In this model the annular area of the pitch circle is given by $AA = 6534.51$ mm^2, and the total area of the bolt holes is given by $AH = 8\pi 8^2 = 1608.5$ mm^2. Hence, the reduction factor is simply $1 - AH/AA = 0.754$. The effective in-plane moduli of elasticity, $E1'$ and $E2'$, are obtained by multiplying the respective moduli, $E1$ and $E2$, by this factor. We assume material isotropy in the ***r–z*** plane; thus, $E1' = E2' = E'$. The modulus in the hoop direction, $E3'$, should be very small and is chosen such that $E'/E3' = 10^6$. The in-plane shear modulus is then calculated based on the effective elasticity modulus: $G'_{12} = E'/2(1 + \nu)$. The shear moduli in the hoop direction are also calculated similarly but with ν set to zero (they are not used in an axisymmetric model). Hence, we have $E1' = E2' = 155292$ MPa, $E3' = 0.155292$ MPa, $G'_{12} = 59728$ MPa, and $G'_{13} = G'_{23} = 0.07765$ MPa. These orthotropic elasticity moduli are specified using engineering constants for the bolt hole part of the mesh.

The mesh for the three-dimensional analysis without substructures, shown in Figure 1.1.1–3, represents a 22.5° segment of the pipe joint and employs second-order brick elements with reduced integration, C3D20R, for the pipe hub/flange and bolts. The gasket is modeled with C3D20R elements or GK3D18 elements. The top figure shows the mesh of the pipe hub and flange, and the bottom figure shows both the gasket and bolt (in the lighter color). Contact is modeled by the interaction of contact surfaces defined by grouping specific faces of the elements in the contacting regions. For three-dimensional contact where both the master and slave surfaces are deformable, the small-sliding contact pair formulation must be used to indicate that small relative sliding occurs between contacting surfaces. No special adjustments need be made for the material properties used in the three-dimensional model because all parts are modeled appropriately.

Four different meshes that use substructures to model the flange are tested. A first-level substructure is created for the entire 22.5° segment of the flange shown in Figure 1.1.1–3, while the gasket and the bolt are meshed as before. The nodes on the flange in contact with the bolt cap form a node-based surface, while the nodes on the flange in contact with the gasket form another node-based surface. These node-based surfaces will form contact pairs with the master surfaces on the bolt cap and on the gasket, which are defined using the surface definition options. The retained degrees of freedom on the substructure include all three degrees of freedom for the nodes in these node-based surfaces as well as for the nodes on the 0° and 22.5° faces of the flange. Appropriate boundary conditions are specified at the substructure usage level.

A second-level substructure of 45° is created by reflecting the first-level substructure with respect to the 22.5° plane. The nodes on the 22.5° face belonging to the reflected substructure are constrained

in all three degrees of freedom to the corresponding nodes on the 22.5° face belonging to the original first-level substructure. The half-bolt and the gasket sector corresponding to the reflected substructure are also constructed by reflection. The retained degrees of freedom include all three degrees of freedom of all contact node sets and of the nodes on the 0° and 45° faces of the flange. MPC-type CYCLSYM is used to impose cyclic symmetric boundary conditions on these two faces.

A third-level substructure of 90° is created by reflecting the original 45° second-level substructure with respect to the 45° plane and by connecting it to the original 45° substructure. The remaining part of the gasket and the bolts corresponding to the 45°–90° sector of the model is created by reflection and appropriate constraints. In this case it is not necessary to retain any degrees of freedom on the 0° and 90° faces of the flange because this 90° substructure will not be connected to other substructures and appropriate boundary conditions can be specified at the substructure creation level.

The final substructure model is set up by mirroring the 90° mesh with respect to the symmetry plane of the gasket perpendicular to the *y*-axis. Thus, an otherwise large analysis ($\approx$ 750,000 unknowns) when no substructures are used can be solved conveniently ($\approx$ 80,000 unknowns) by using the third-level substructure twice. The sparse solver is used because it significantly reduces the run time for this model.

Finally, a three-dimensional matrix-based model is created by replacing elements for the entire 22.5° segment of the flange shown in Figure 1.1.1–3 with stiffness matrices, while the gasket and the bolt are meshed as before. Contact between the flange and gasket and the flange and bolt cap is modeled using node-based slave surfaces just as for the substructure models. Appropriate boundary conditions are applied as in the three-dimensional model without substructures.

Loading and boundary conditions

The only boundary conditions are symmetry boundary conditions. In the axisymmetric model $u_z = 0$ is applied to the symmetry plane of the gasket and to the bottom of the bolts. In the three-dimensional model $u_y = 0$ is applied to the symmetry plane of the gasket as well as to the bottom of the bolt. The θ =0° and θ =22.5° planes are also symmetry planes. On the θ =22.5° plane, symmetry boundary conditions are enforced by invoking suitable nodal transformations and applying boundary conditions to local directions in this symmetry plane. These transformations are implemented using a local coordinate system definition. On both the symmetry planes, the symmetry boundary conditions $u_z = 0$ are imposed everywhere except for the dependent nodes associated with the C BIQUAD MPC and nodes on one side of the contact surface. The second exception is made to avoid overconstraining problems, which arise if there is a boundary condition in the same direction as a Lagrange multiplier constraint associated with the rough friction specification.

In the models where substructures are used, the boundary conditions are specified depending on what substructure is used. For the first-level 22.5° substructure the boundary conditions and constraint equations are the same as for the three-dimensional model shown in Figure 1.1.1–3. For the 45° second-level substructure the symmetry boundary conditions are enforced on the θ =45° plane with the constraint equation $u_z + u_x = 0$. A transform could have been used as well. For the 90° third-level substructure the face θ =90° is constrained with the boundary condition $u_x = 0$.

For the three-dimensional model containing matrices, nodal transformations are applied for symmetric boundary conditions. Entries in the stiffness matrices for these nodes are also in local coordinates.

A clamping force of 15 kN is applied to each bolt by associating the pre-tension node with a pre-tension section. The pre-tension section is identified by means of a surface definition. The pre-tension is then prescribed by applying a concentrated load to the pre-tension node. In the axisymmetric analysis the actual load applied is 120 kN since there are eight bolts. In the three-dimensional model with no substructures the actual load applied is 7.5 kN since only half of a bolt is modeled. In the models using substructures all half-bolts are loaded with a 7.5 kN force. For all of the models the pre-tension section is specified about halfway down the bolt shank.

Sticking contact conditions are assumed in all surface interactions in all analyses and are simulated with rough friction and no-separation contact.

Results and discussion

All analyses are performed as small-displacement analyses.

Figure 1.1.1–5 shows a top view of the normal stress distributions in the gasket at the interface between the gasket and the pipe hub/flange predicted by the axisymmetric (bottom) and three-dimensional (top) analyses when solid continuum elements are used to model the gasket. The figure shows that the compressive normal stress is highest at the outer edge of the gasket, decreases radially inward, and changes from compression to tension at a radius of about 35 mm, which is consistent with findings reported by Sawa et al. (1991). The close agreement in the overall solution between axisymmetric and three-dimensional analyses is quite apparent, indicating that, for such problems, axisymmetric analysis offers a simple yet reasonably accurate alternative to three-dimensional analysis.

Figure 1.1.1–6 shows a top view of the normal stress distributions in the gasket at the interface between the gasket and the pipe hub/flange predicted by the axisymmetric (bottom) and three-dimensional (top) analyses when gasket elements are used to model the gasket. Close agreement in the overall solution between the axisymmetric and three-dimensional analyses is also seen in this case. The gasket starts carrying compressive load at a radius of about 40 mm, a difference of 5 mm with the previous result. This difference is the result of the gasket elements being unable to carry tensile loads in their thickness direction. This solution is physically more realistic since, in most cases, gaskets separate from their neighboring parts when subjected to tensile loading. Removing the no-separation contact from the gasket/flange contact surface definition in the input files that model the gasket with continuum elements yields good agreement with the results obtained in Figure 1.1.1–6 (since, in that case, the solid continuum elements in the gasket cannot carry tensile loading in the gasket thickness direction).

The models in this example can be modified to study other factors, such as the effective seating width of the gasket or the sealing performance of the gasket under operating loads. The gasket elements offer the advantage of allowing very complex behavior to be defined in the gasket thickness direction. Gasket elements can also use any of the small-strain material models provided in Abaqus including user-defined material models. Figure 1.1.1–7 shows a comparison of the normal stress distributions in the gasket at the interface between the gasket and the pipe hub/flange predicted by the axisymmetric (bottom) and three-dimensional (top) analyses when isotropic material properties are prescribed for gasket elements. The results in Figure 1.1.1–7 compare well with the results in Figure 1.1.1–5 from analyses in which solid and axisymmetric elements are used to simulate the gasket.

Figure 1.1.1–8 shows the distribution of the normal stresses in the gasket at the interface in the plane $z = 0$. The results are plotted for the three-dimensional model containing only solid continuum elements

and no substructures, for the three-dimensional model with matrices, and for the four models containing the substructures described above.

An execution procedure is available to combine model and results data from two substructure output databases into a single output database. For more information, see "Combining output from substructures," Section 3.2.22 of the Abaqus Analysis User's Guide.

This example can also be used to demonstrate the effectiveness of the quasi-Newton nonlinear solver. This solver utilizes an inexpensive, approximate stiffness matrix update for several consecutive equilibrium iterations, rather than a complete stiffness matrix factorization each iteration as used in the default full Newton method. The quasi-Newton method results in an increased number of less expensive iterations, and a net savings in computing cost.

Input files

boltpipeflange_axi_solidgask.inp	Axisymmetric analysis containing a gasket modeled with solid continuum elements.
boltpipeflange_axi_node.inp	Node definitions for boltpipeflange_axi_solidgask.inp and boltpipeflange_axi_gkax6.inp.
boltpipeflange_axi_element.inp	Element definitions for boltpipeflange_axi_solidgask.inp.
boltpipeflange_3d_solidgask.inp	Three-dimensional analysis containing a gasket modeled with solid continuum elements.
boltpipeflange_axi_gkax6.inp	Axisymmetric analysis containing a gasket modeled with gasket elements.
boltpipeflange_3d_gk3d18.inp	Three-dimensional analysis containing a gasket modeled with gasket elements.
boltpipeflange_3d_substr1.inp	Three-dimensional analysis using the first-level substructure (22.5° model).
boltpipeflange_3d_substr2.inp	Three-dimensional analysis using the second-level substructure (45° model).
boltpipeflange_3d_substr3_1.inp	Three-dimensional analysis using the third-level substructure once (90° model).
boltpipeflange_3d_substr3_2.inp	Three-dimensional analysis using the third-level substructure twice (90° mirrored model).
boltpipeflange_3d_gen1.inp	First-level substructure generation data referenced by boltpipeflange_3d_substr1.inp and boltpipeflange_3d_gen2.inp.
boltpipeflange_3d_gen2.inp	Second-level substructure generation data referenced by boltpipeflange_3d_substr2.inp and boltpipeflange_3d_gen3.inp.
boltpipeflange_3d_gen3.inp	Third-level substructure generation data referenced by boltpipeflange_3d_substr3_1.inp and boltpipeflange_3d_substr3_2.inp.

boltpipeflange_3d_node.inp	Nodal coordinates used in boltpipeflange_3d_substr1.inp, boltpipeflange_3d_substr2.inp, boltpipeflange_3d_substr3_1.inp, boltpipeflange_3d_substr3_2.inp, boltpipeflange_3d_cyclsym.inp, boltpipeflange_3d_gen1.inp, boltpipeflange_3d_gen2.inp, and boltpipeflange_3d_gen3.inp.
boltpipeflange_3d_cyclsym.inp	Same as file boltpipeflange_3d_substr2.inp except that CYCLSYM type MPCs are used.
boltpipeflange_3d_missnode.inp	Same as file boltpipeflange_3d_gk3d18.inp except that the option to generate missing nodes is used for gasket elements.
boltpipeflange_3d_isomat.inp	Same as file boltpipeflange_3d_gk3d18.inp except that gasket elements are modeled as isotropic using the *MATERIAL option.
boltpipeflange_3d_ortho.inp	Same as file boltpipeflange_3d_gk3d18.inp except that gasket elements are modeled as orthotropic and the *ORIENTATION option is used.
boltpipeflange_axi_isomat.inp	Same as file boltpipeflange_axi_gkax6.inp except that gasket elements are modeled as isotropic using the *MATERIAL option.
boltpipeflange_3d_usr_umat.inp	Same as file boltpipeflange_3d_gk3d18.inp except that gasket elements are modeled as isotropic with user subroutine **UMAT**.
boltpipeflange_3d_usr_umat.f	User subroutine **UMAT** used in boltpipeflange_3d_usr_umat.inp.
boltpipeflange_3d_solidnum.inp	Same as file boltpipeflange_3d_gk3d18.inp except that solid element numbering is used for gasket elements.
boltpipeflange_3d_matrix.inp	Three-dimensional analysis containing matrices and a gasket modeled with solid continuum elements.
boltpipeflange_3d_stiffPID4.inp	Matrix representing stiffness of a part of the flange segment for three-dimensional analysis containing matrices.
boltpipeflange_3d_stiffPID5.inp	Matrix representing stiffness of the remaining part of the flange segment for three-dimensional analysis containing matrices.
boltpipeflange_3d_qn.inp	Same as file boltpipeflange_3d_gk3d18.inp except that the quasi-Newton nonlinear solver is used.

References

- Bibel, G. D., and R. M. Ezell, "An Improved Flange Bolt-Up Procedure Using Experimentally Determined Elastic Interaction Coefficients," Journal of Pressure Vessel Technology, vol. 114, pp. 439–443, 1992.
- Chaaban, A., and U. Muzzo, "Finite Element Analysis of Residual Stresses in Threaded End Closures," Transactions of ASME, vol. 113, pp. 398–401, 1991.
- Fukuoka, T., "Finite Element Simulation of Tightening Process of Bolted Joint with a Tensioner," Journal of Pressure Vessel Technology, vol. 114, pp. 433–438, 1992.
- Sawa, T., N. Higurashi, and H. Akagawa, "A Stress Analysis of Pipe Flange Connections," Journal of Pressure Vessel Technology, vol. 113, pp. 497–503, 1991.

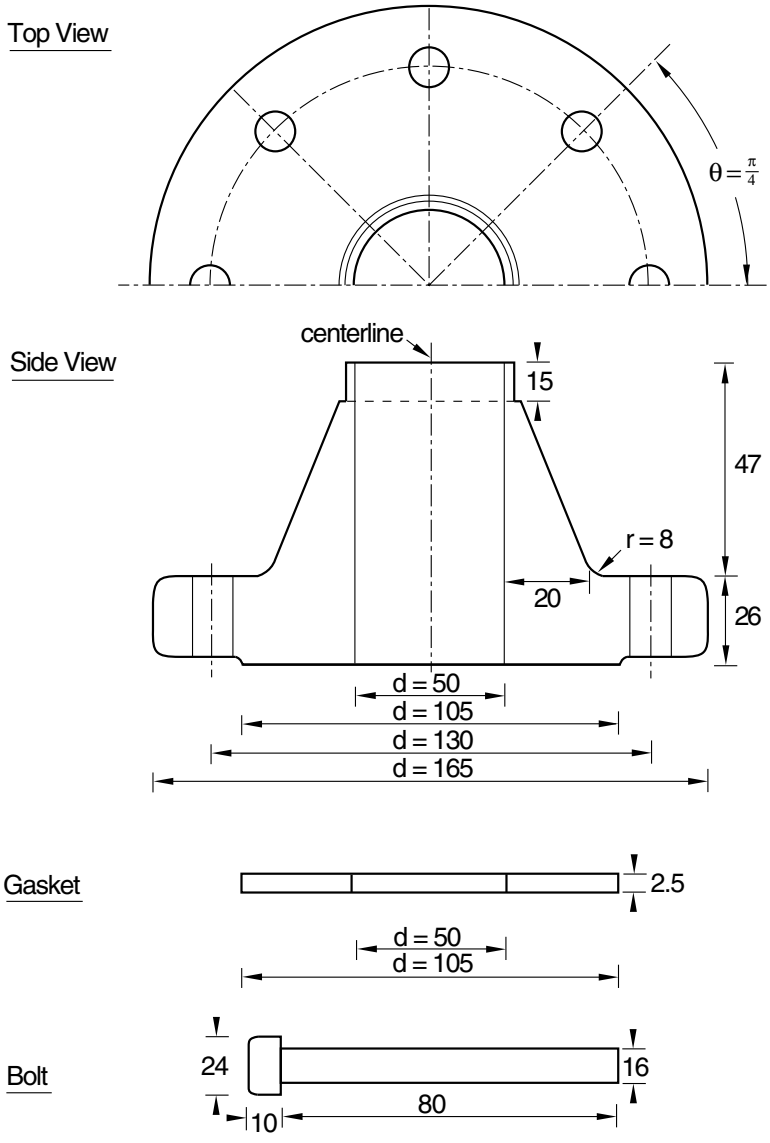


Figure 1.1.1–1 Schematic of the bolted joint. All dimensions in mm.

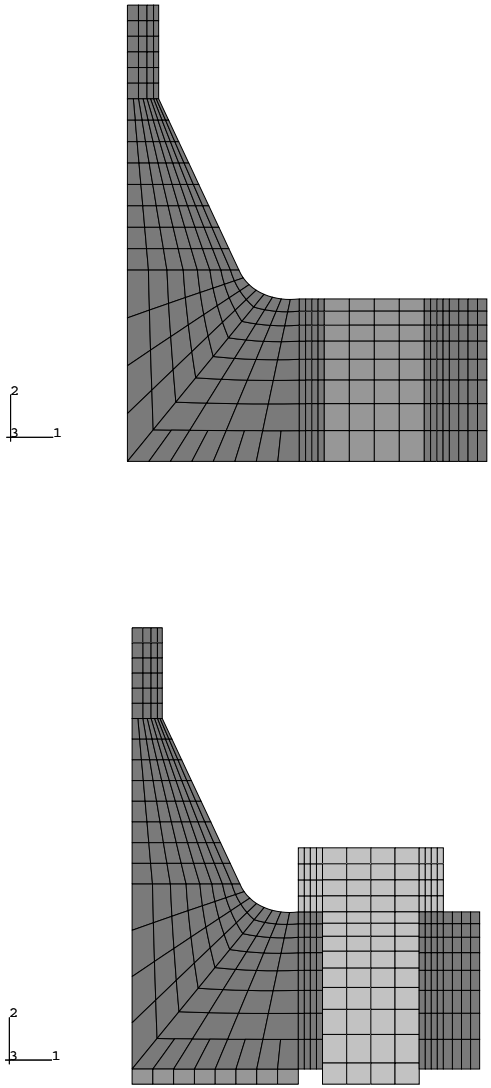


Figure 1.1.1–2 Axisymmetric model of the bolted joint.

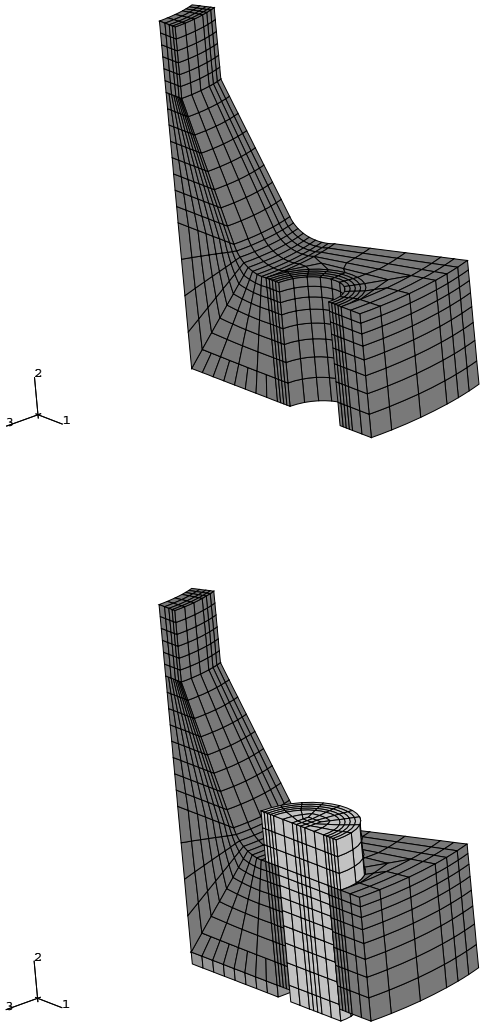


Figure 1.1.1–3 22.5° segment three-dimensional model of the bolted joint.

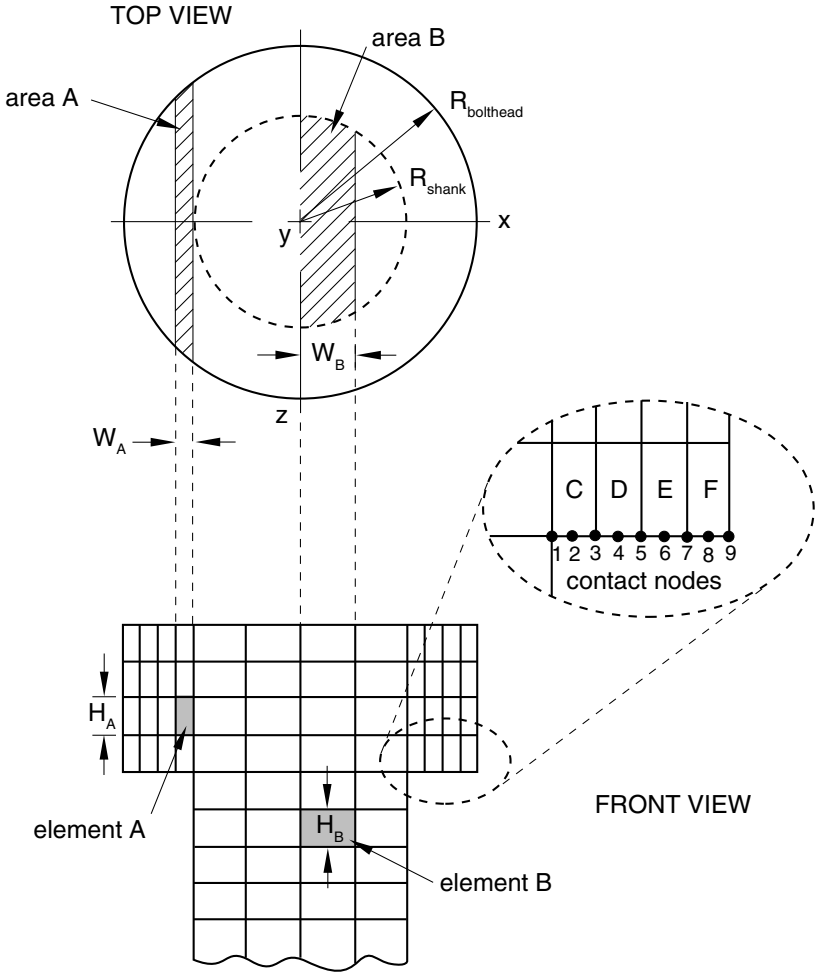


Figure 1.1.1–4 Cross-sectional views of the bolt head and the shank.

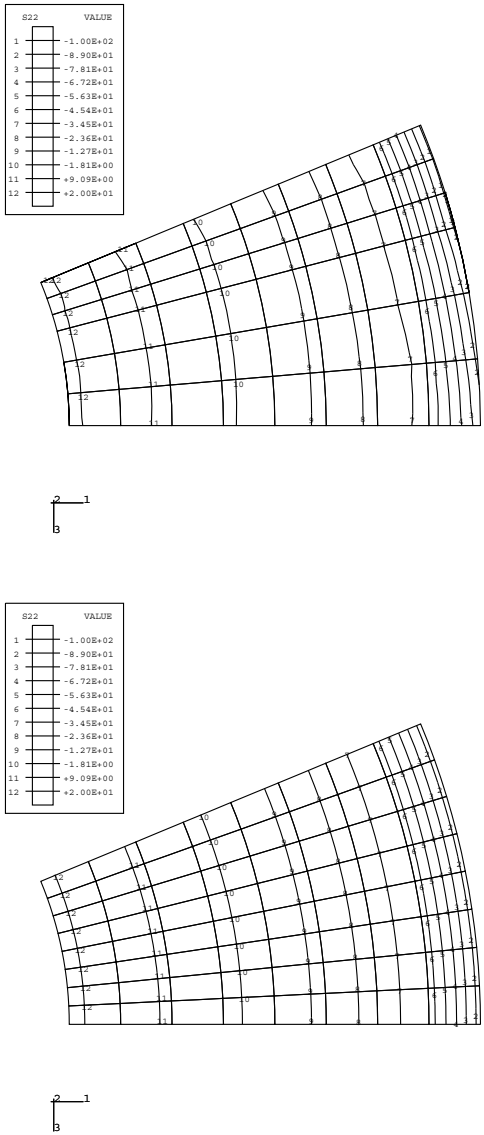


Figure 1.1.1–5 Normal stress distribution in the gasket contact surface when solid elements are used to model the gasket: three-dimensional versus axisymmetric results.

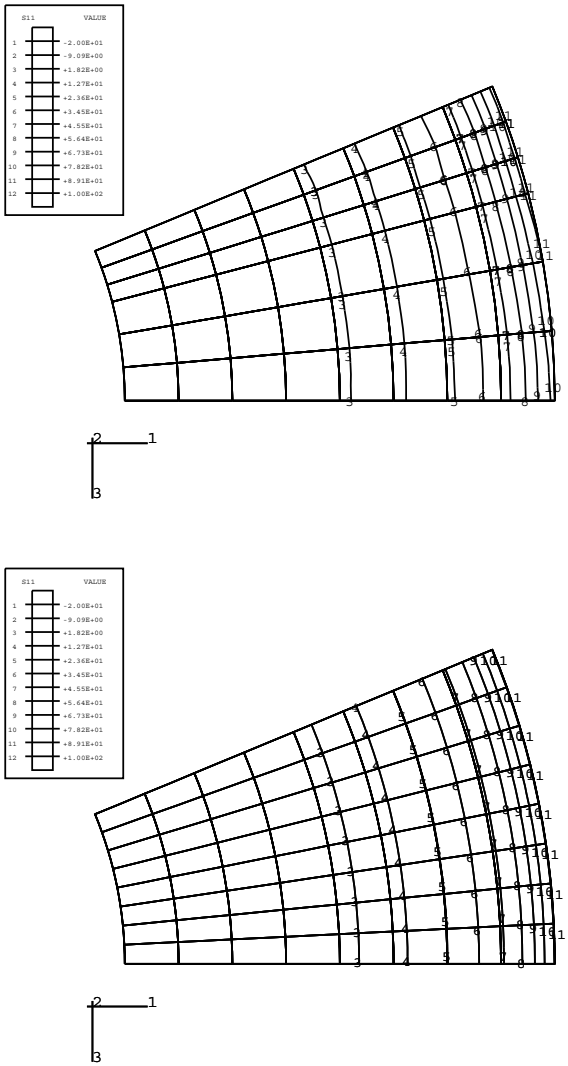


Figure 1.1.1–6 Normal stress distribution in the gasket contact surface when gasket elements are used with direct specification of the gasket behavior: three-dimensional versus axisymmetric results.

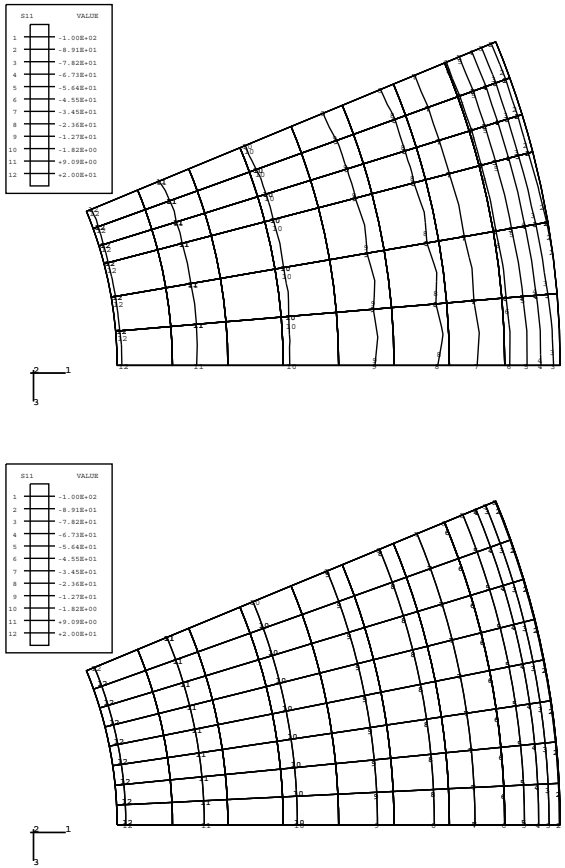


Figure 1.1.1–7 Normal stress distribution in the gasket contact surface when gasket elements are used with isotropic material properties: three-dimensional versus axisymmetric results.

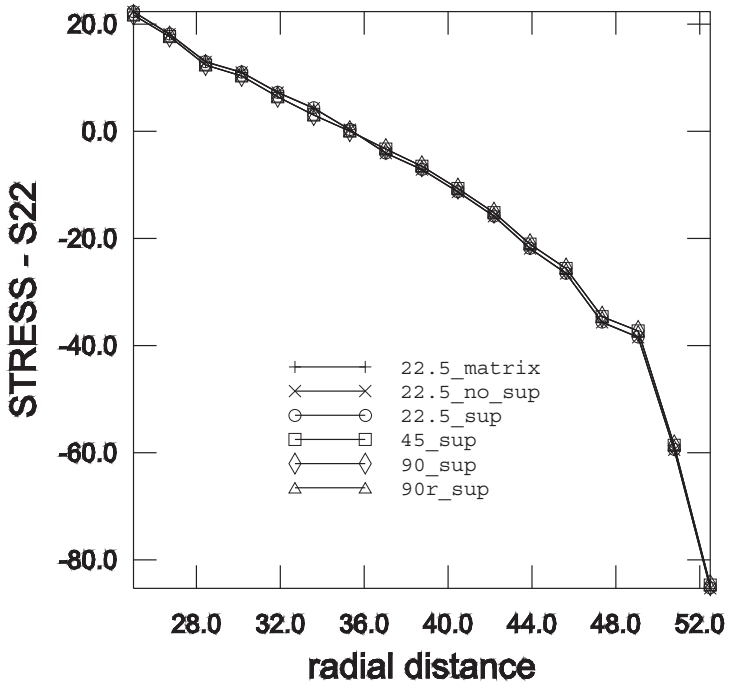


Figure 1.1.1–8 Normal stress distribution in the gasket contact surface along the line $z = 0$ for the models with and without substructures.

1.1.2 ELASTIC-PLASTIC COLLAPSE OF A THIN-WALLED ELBOW UNDER IN-PLANE BENDING AND INTERNAL PRESSURE

Product: Abaqus/Standard

Elbows are used in piping systems because they ovalize more readily than straight pipes and, thus, provide flexibility in response to thermal expansion and other loadings that impose significant displacements on the system. Ovalization is the bending of the pipe wall into an oval—i.e., noncircular—configuration. The elbow is, thus, behaving as a shell rather than as a beam. Straight pipe runs do not ovalize easily, so they behave essentially as beams. Thus, even under pure bending, complex interaction occurs between an elbow and the adjacent straight pipe segments; the elbow causes some ovalization in the straight pipe runs, which in turn tend to stiffen the elbow. This interaction can create significant axial gradients of bending strain in the elbow, especially in cases where the elbow is very flexible. This example provides verification of shell and elbow element modeling of such effects, through an analysis of a test elbow for which experimental results have been reported by Sobel and Newman (1979). An analysis is also included with elements of type ELBOW31B (which includes ovalization but neglects axial gradients of strain) for the elbow itself and beam elements for the straight pipe segments. This provides a comparative solution in which the interaction between the elbow and the adjacent straight pipes is neglected. The analyses predict the response up to quite large rotations across the elbow, so as to investigate possible collapse of the pipe and, particularly, the effect of internal pressure on that collapse.

Geometry and model

The elbow configuration used in the study is shown in Figure 1.1.2–1. It is a thin-walled elbow with elbow factor

$$\lambda = \frac{Rt}{r^2\sqrt{1-\nu^2}} = 0.167$$

and radius ratio $R/r = 3.07$, so the flexibility factor from Dodge and Moore (1972) is 10.3. (The flexibility factor for an elbow is the ratio of the bending flexibility of an elbow segment to that of a straight pipe of the same dimensions, for small displacements and elastic response.) This is an extremely flexible case because the pipe wall is so thin.

To demonstrate convergence of the overall moment-rotation behavior with respect to meshing, the two shell element meshes shown in Figure 1.1.2–2 are analyzed. Since the loading concerns in-plane bending only, it is assumed that the response is symmetric about the midplane of the system so that in the shell element model only one-half of the system need be modeled. Element type S8R5 is used, since tests have shown this to be the most cost-effective shell element in Abaqus (input files using element types S9R5, STRI65, and S8R for this example are included with the Abaqus release). The elbow element meshes replace each axial division in the coarser shell element model with one ELBOW32 or two ELBOW31 elements and use 4 or 6 Fourier modes to model the deformation around the pipe. Seven integration points are used through the pipe wall in all the analyses. This is usually adequate to

provide accurate modeling of the progress of yielding through the section in such cases as these, where essentially monotonic straining is expected.

The ends of the system are rigidly attached to stiff plates in the experiments. These boundary conditions are easily modeled for the ELBOW elements and for the fixed end in the shell element model. For the rotating end of the shell element model the shell nodes must be constrained to a beam node that represents the motion of the end plate using a kinematic coupling constraint as described below.

The material is assumed to be isotropic and elastic-plastic, following the measured response of type 304 stainless steel at room temperature, as reported by Sobel and Newman (1979). Since all the analyses give results that are stiffer than the experimentally measured response, and the mesh convergence tests (results are discussed below) demonstrate that the meshes are convergent with respect to the overall response of the system, it seems that this stress-strain model may overestimate the material's actual strength.

Loading

The load on the pipe has two components: a "dead" load, consisting of internal pressure (with a closed end condition), and a "live" in-plane bending moment applied to the end of the system. The pressure is applied to the model in an initial step and then held constant in the second analysis step while the bending moment is increased. The pressure values range from 0.0 to 3.45 MPa (500 lb/in^2), which is the range of interest for design purposes. The equivalent end force associated with the closed-end condition is applied as a follower force because it rotates with the motion of the end plane.

Kinematic boundary conditions

The fixed end of the system is assumed to be fully built-in. The loaded end is fixed into a very stiff plate. For the ELBOW element models this condition is represented by the **NODEFORM** boundary condition applied at this node. In the shell element model this rigid plate is represented by a single node, and the shell nodes at the end of the pipe are attached to it by using a kinematic coupling constraint and specifying that all degrees of freedom at the shell nodes are constrained to the motion of the single node.

Results and discussion

The moment-rotation responses predicted by the various analysis models and measured in the experiment, all taken at zero internal pressure, are compared in Figure 1.1.2–3. The figure shows that the two shell models give very similar results, overestimating the experimentally measured collapse moment by about 15%. The 6-mode ELBOW element models are somewhat stiffer than the shell models, and those with 4 Fourier modes are much too stiff. This clearly shows that, for this very flexible system, the ovalization of the elbow is too localized for even the 6-mode ELBOW representation to provide accurate results.

Since we know that the shell models are convergent with respect to discretization, the most likely explanation for the excessive stiffness in comparison to the experimentally measured response is that the material model used in the analyses is too strong. Sobel and Newman (1979) point out that the stress-strain curve measured and used in this analysis, shown in Figure 1.1.2–1, has a 0.2% offset yield that is 20% higher than the Nuclear Systems Materials Handbook value for type 304 stainless steel at room temperature, which suggests the possibility that the billets used for the stress-strain curve

measurement may have been taken from stronger parts of the fabrication. If this is the case, it points out the likelihood that the elbow tested is rather nonuniform in strength properties in spite of the care taken in its manufacture. We are left with the conclusion that discrepancies of this magnitude cannot be eliminated in practical cases, and the design use of such analysis results must allow for them.

Figure 1.1.2–4 compares the moment-rotation response for opening and closing moments under 0 and 3.45 MPa (500 lb/in^2) internal pressure and shows the strong influence of large-displacement effects. If large-displacement effects were not important, the opening and closing moments would produce the same response. However, even with a 1° relative rotation across the elbow assembly, the opening and closing moments differ by about 12%; with a 2° relative rotation, the difference is about 17%. Such magnitudes of relative rotation would not normally be considered large; in this case it is the coupling into ovalization that makes geometric nonlinearity significant. As the rotation increases, the cases with closing moment loading show collapse, while the opening moment curves do not. In both cases internal pressure shows a strong effect on the results, which is to be expected in such a thin-walled pipeline. The level of interaction between the straight pipe and the elbows is well illustrated by the strain distribution on the outside wall, shown in Figure 1.1.2–5. The strain contours are slightly discontinuous at the ends of the curved elbow section because the shell thickness changes at those sections.

Figure 1.1.2–6 shows a summary of the results from this example and "Uniform collapse of straight and curved pipe segments," Section 1.1.5 of the Abaqus Benchmarks Guide. The plot shows the collapse value of the closing moment under in-plane bending as a function of internal pressure. The strong influence of pressure on collapse is apparent. In addition, the effect of analyzing the elbow by neglecting interaction between the straight and curved segments is shown: the "uniform bending" results are obtained by using elements of type ELBOW31B in the bend and beams (element type B31) for the straight segments. The importance of the straight/elbow interaction is apparent. In this case the simpler analysis neglecting the interaction is conservative (in that it gives consistently lower values for the collapse moment), but this conservatism cannot be taken for granted. The analysis of Sobel and Newman (1979) also neglects interaction and agrees quite well with the results obtained here.

For comparison the small-displacement limit analysis results of Goodall (1978), as well as his large-displacement, elastic-plastic lower bound (Goodall, 1978a), are also shown in this figure. Again, the importance of large-displacement effects is apparent from that comparison.

Detailed results obtained with the model that uses ELBOW31 elements are shown in Figure 1.1.2–7 through Figure 1.1.2–9. Figure 1.1.2–7 shows the variation of the Mises stress along the length of the piping system. The length is measured along the centerline of the pipe starting at the loaded end. The figure compares the stress distribution at the intrados (integration point 1) on the inner and outer surfaces of the elements (section points 1 and 7, respectively). Figure 1.1.2–8 shows the variation of the Mises stress around the circumference of two elements (451 and 751) that are located in the bend section of the model; the results are for the inner surface of the elements (section point 1). Figure 1.1.2–9 shows the ovalization of elements 451 and 751. A nonovalized, circular cross-section is included in the figure for comparison. From the figure it is seen that element 751, located at the center of the bend section, experiences the most severe ovalization. These three figures were produced with the aid of the elbow element postprocessing program **`felbow.f`** ("Creation of a data file to facilitate the postprocessing of elbow element results: FELBOW," Section 15.1.6), written in FORTRAN. The postprocessing programs **`felbow.C`** ("A C++ version of FELBOW," Section 10.15.6 of the Abaqus Scripting User's Guide) and

felbow.py ("An Abaqus Scripting Interface version of FELBOW," Section 9.10.12 of the Abaqus Scripting User's Guide), written in C++ and Python, respectively, are also available for generating the data for figures such as Figure 1.1.2–8 and Figure 1.1.2–9. The user must ensure that the output variables are written to the output database to use these two programs.

Shell-to-solid submodeling

One particular case is analyzed using the shell-to-solid submodeling technique. This problem verifies the interpolation scheme in the case of double curved surfaces. A solid submodel using C3D27R elements is created around the elbow part of the pipe, spanning an angle of 40°. The finer submodel mesh has three elements through the thickness, 10 elements around half of the circumference of the cylinder, and 10 elements along the length of the elbow. Both ends are driven from the global shell model made of S8R elements. The time scale of the static submodel analysis corresponds to the arc length in the global Riks analysis. The submodel results agree closely with the shell model. The total force and the total moment in a cross-section through the submodel are written to the results (**.fil**) file.

Shell-to-solid coupling

A model using the shell-to-solid coupling capability in Abaqus is included. Such a model can be used for a careful study of the stress and strain fields in the elbow. The entire elbow is meshed with C3D20R elements, and the straight pipe sections are meshed with S8R elements (see Figure 1.1.2–10). At each shell-to-solid interface illustrated in Figure 1.1.2–10, an element-based surface is defined on the edge of the solid mesh and an edge-based surface is defined on the edge of the shell mesh. A shell-to-solid coupling constraint is used in conjunction with these surfaces to couple the shell and solid meshes.

Edge-based surfaces are defined at the end of each pipe segment. These surfaces are coupled to reference nodes that are defined at the center of the pipes using a distributing coupling constraint. The loading and fixed boundary conditions are applied to the reference points. The advantage of using this method is that the pipe cross-sectional areas are free to deform; thus, ovalization at the ends is not constrained. The moment-rotation response of the shell-to-solid coupling model agrees very well with the results shown in Figure 1.1.2–4.

Input files

In all the following input files (with the exception of elbowcollapse_elbow31b_b31.inp, elbowcollapse_s8r5_fine.inp, and elbowcolpse_shl2sld_s8r_c3d20r.inp) the step concerning the application of the pressure load is commented out. To include the effects of the internal pressure in any given analysis, uncomment the step definition in the appropriate input file.

elbowcollapse_elbow31b_b31.inp	ELBOW31B and B31 element model.
elbowcollapse_elbow31_6four.inp	ELBOW31 model with 6 Fourier modes.
elbowcollapse_elbow32_6four.inp	ELBOW32 model with 6 Fourier modes.
elbowcollapse_s8r.inp	S8R element model.
elbowcollapse_s8r5.inp	S8R5 element model.
elbowcollapse_s8r5_fine.inp	Finer S8R5 element model.

elbowcollapse_s9r5.inp	S9R5 element model.
elbowcollapse_stri65.inp	STRI65 element model.
elbowcollapse_submod.inp	Submodel using C3D27R elements.
elbowcolpse_shl2sld_s8r_c3d20r.inp	Shell-to-solid coupling model using S8R and C3D20R elements.

References

- 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, 1972.
- Goodall, I. W., "Lower Bound Limit Analysis of Curved Tubes Loaded by Combined Internal Pressure and In-Plane Bending Moment," Research Division Report RD/B/N4360, Central Electricity Generating Board, England, 1978.
- Goodall, I. W., "Large Deformations in Plastically Deforming Curved Tubes Subjected to In-Plane Bending," Research Division Report RD/B/N4312, Central Electricity Generating Board, England, 1978a.
- Sobel, L. H., and S. Z. Newman, "Elastic-Plastic In-Plane Bending and Buckling of an Elbow: Comparison of Experimental and Simplified Analysis Results," Westinghouse Advanced Reactors Division, Report WARD–HT–94000–2, 1979.

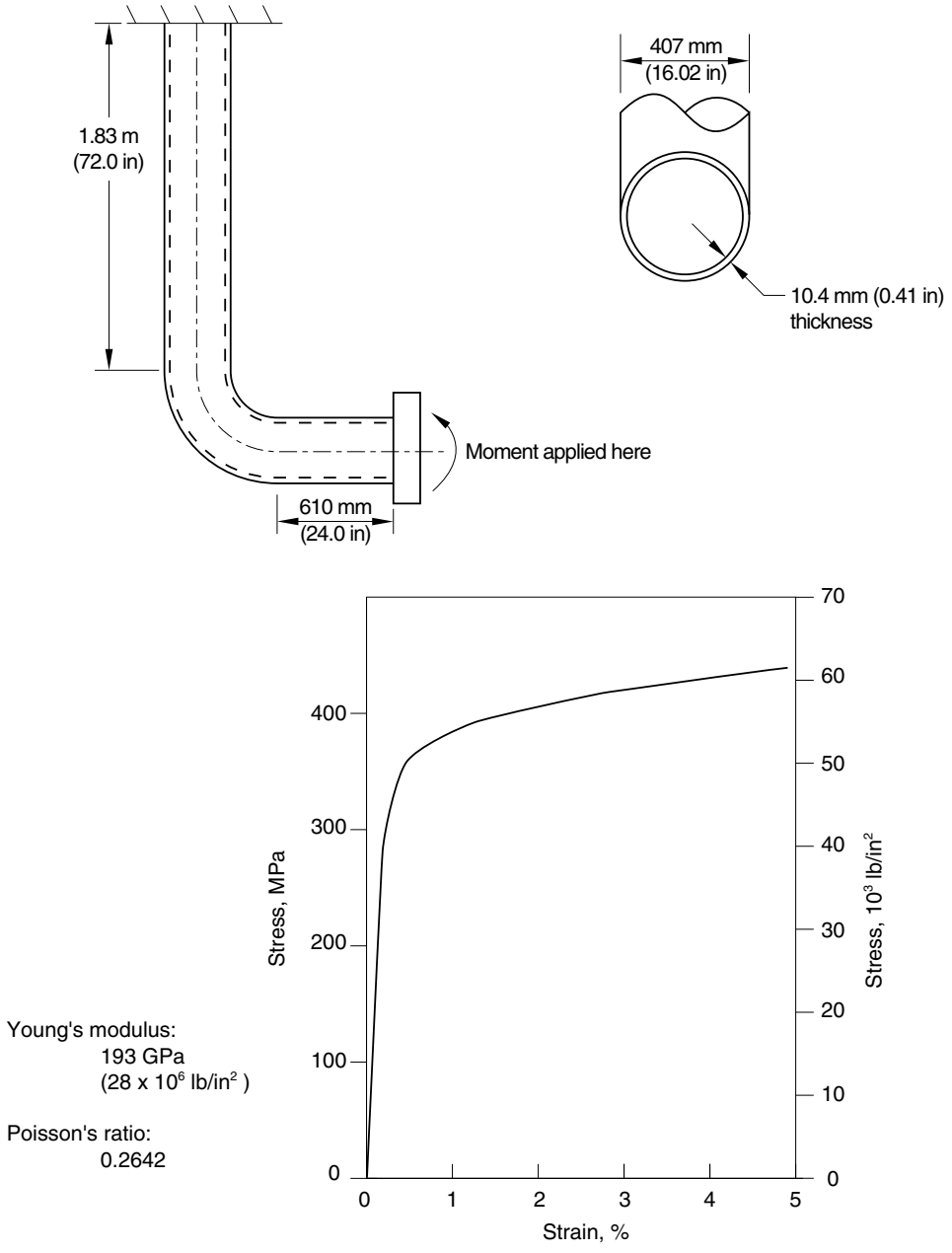


Figure 1.1.2–1 MLTF elbow: geometry and measured material response.

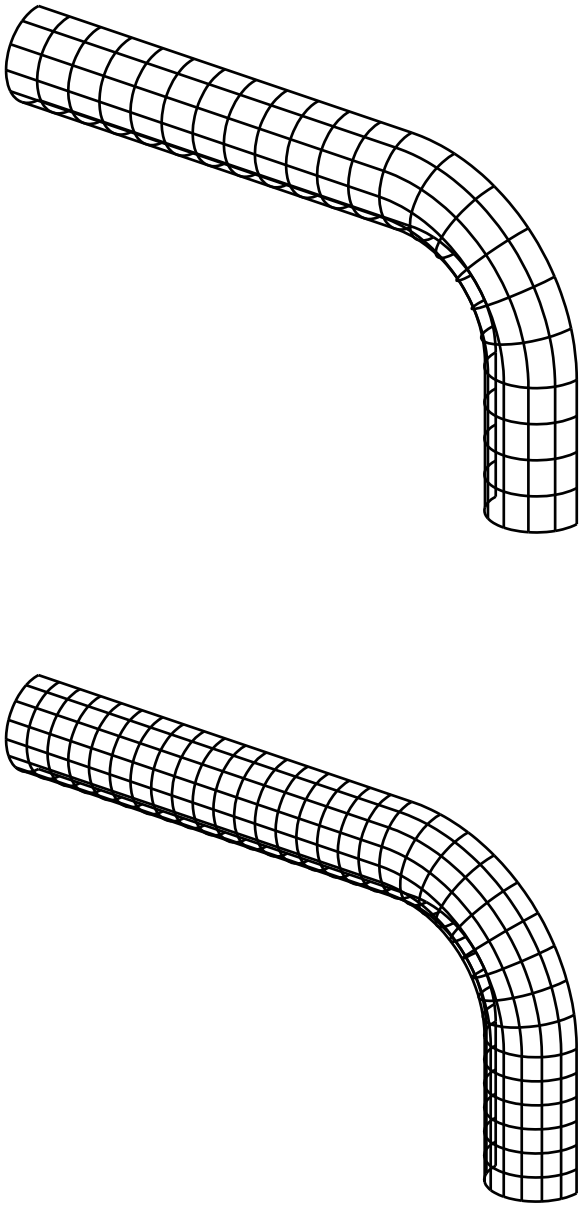


Figure 1.1.2–2 Models for elbow/pipe interaction study.

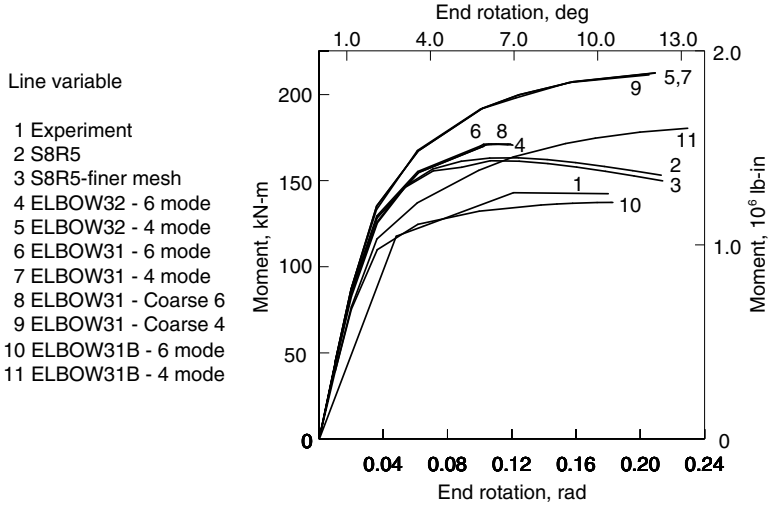


Figure 1.1.2–3 Moment-rotation response: mesh convergence studies.

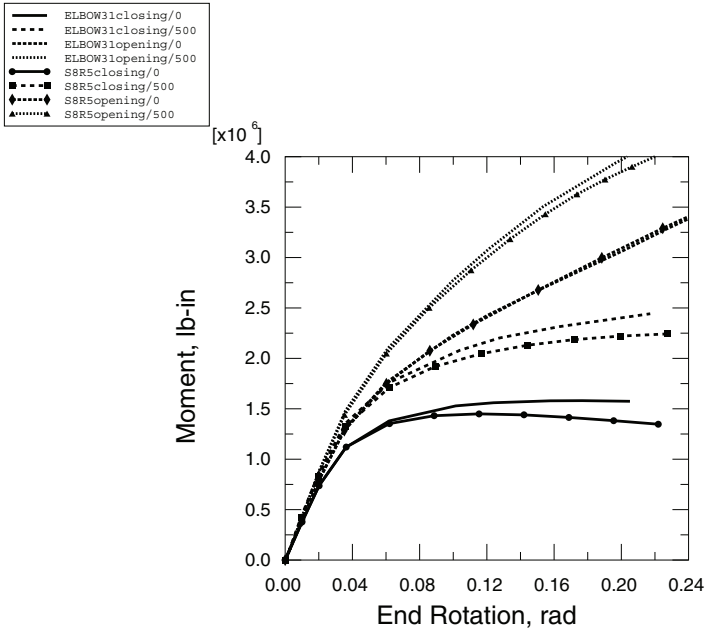
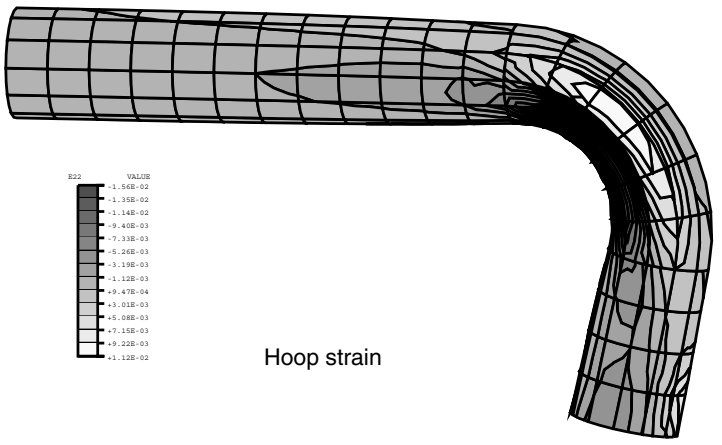


Figure 1.1.2–4 Moment-rotation response: pressure dependence.



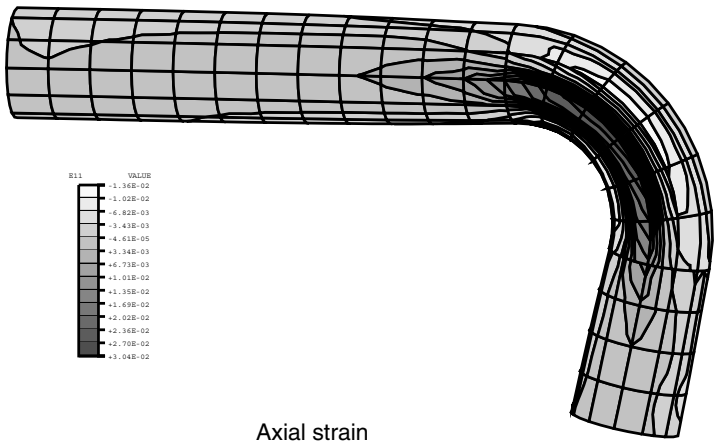


Figure 1.1.2–5 Strain distribution on the outside surface: closing moment case.

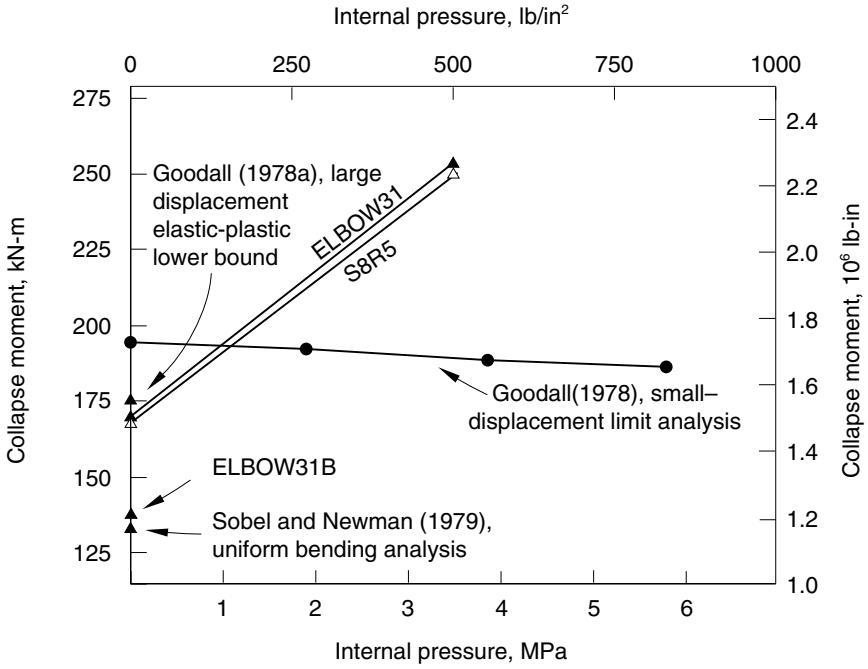


Figure 1.1.2–6 In-plane bending of an elbow, elastic-plastic collapse moment results.

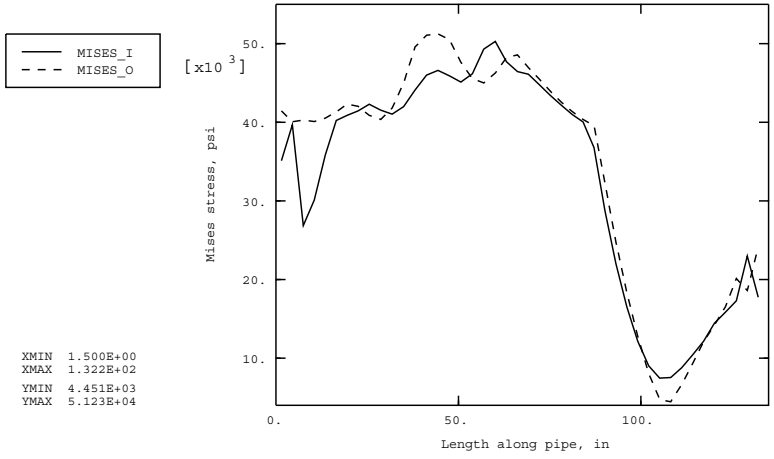


Figure 1.1.2–7 Mises stress distribution along the length of the piping system.

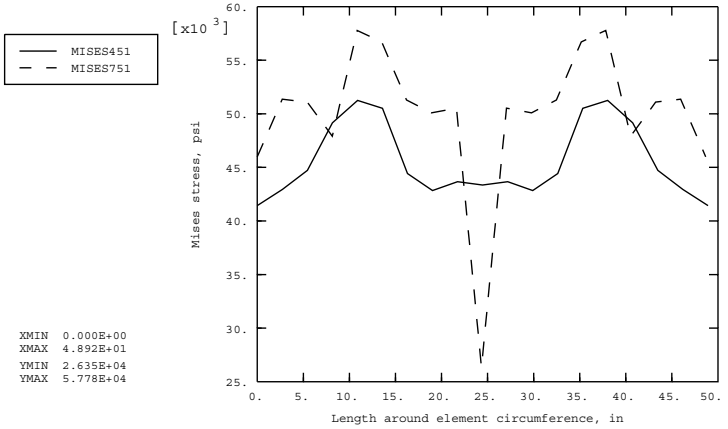


Figure 1.1.2–8 Mises stress distribution around the circumference of elements 451 and 751.

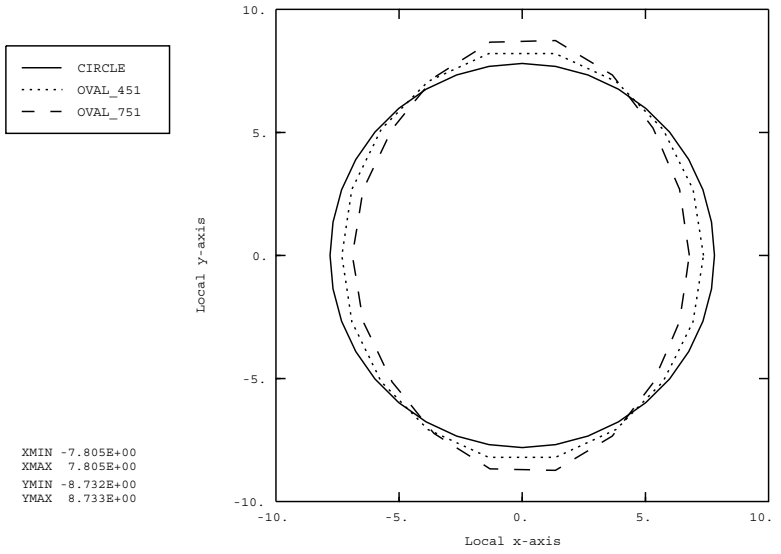


Figure 1.1.2–9 Ovalization of elements 451 and 751.

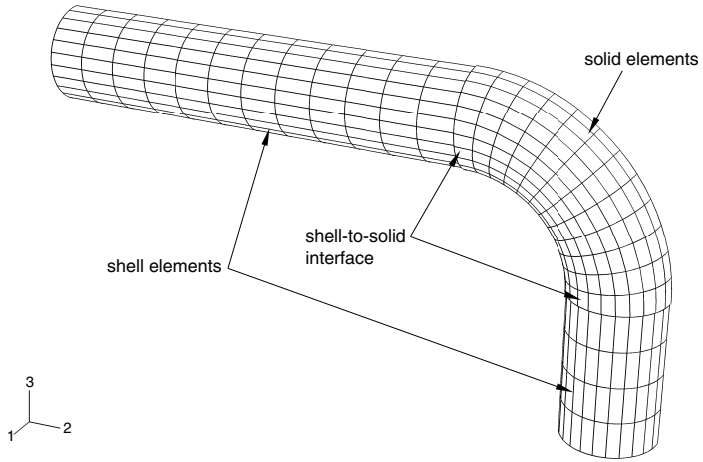


Figure 1.1.2–10 Shell-to-solid coupling model study.

1.1.3 PARAMETRIC STUDY OF A LINEAR ELASTIC PIPELINE UNDER IN-PLANE BENDING

Products: Abaqus/Standard Abaqus/Explicit

Elbows are used in piping systems because they ovalize more readily than straight pipes and, thus, provide flexibility in response to thermal expansion and other loadings that impose significant displacements on the system. Ovalization is the bending of the pipe wall into an oval—i.e., noncircular—configuration. The elbow is, thus, behaving as a shell rather than as a beam. This example demonstrates the ability of elbow elements ("Pipes and pipebends with deforming cross-sections: elbow elements," Section 29.5.1 of the Abaqus Analysis User's Guide) to model the nonlinear response of initially circular pipes and pipebends accurately when the distortion of the cross-section by ovalization is significant. It also provides some guidelines on the importance of including a sufficient number of Fourier modes in the elbow elements to capture the ovalization accurately. In addition, this example illustrates the shortcomings of using "flexibility knockdown factors" with simple beam elements in an attempt to capture the effects of ovalization in an ad hoc manner for large-displacement analyses. Similar analyses involving pipe elements in Abaqus/Explicit are included.

Geometry and model

The pipeline configuration used in the study is shown in Figure 1.1.3–1. It is a simple model with two straight pipe sections connected by a 90° elbow. The straight pipes are 25.4 cm (10.0 inches) in length, the radius of the curved section is 10.16 cm (4.0 inches), and the outer radius of the pipe section is 1.27 cm (0.5 inches). The wall thickness of the pipe is varied from 0.03175 cm to 0.2032 cm (0.0125 inches to 0.08 inches) in a parametric study, as discussed below. The pipe material is assumed to be isotropic linear elastic with a Young's modulus of 194 GPa (28.1×10^6 psi) and a Poisson's ratio of 0.0. The straight portions of the pipeline are assumed to be long enough so that warping at the ends of the structure is negligible.

Two loading conditions are analyzed. The first case is shown in Figure 1.1.3–1 with unit inward displacements imposed on both ends of the structure. This loading condition has the effect of closing the pipeline in on itself. In the second case the sense of the applied unit displacements is outward, opening the pipeline. Both cases are considered to be large-displacement/small-strain analyses.

A parametric study comparing the results obtained with different element types (shells, elbows, and pipes) over a range of flexibility factors, k, is performed. As defined in Dodge and Moore (1972), the flexibility factor for an elbow is the ratio of the bending flexibility of the elbow segment to that of a straight pipe of the same dimensions, assuming small displacements and an elastic response. When the internal (gauge) pressure is zero, as is assumed in this study, k can be approximated as

$$k = \frac{1.66}{\lambda},$$

where

$$\lambda = \frac{Rt}{r^2\sqrt{1-\nu^2}},$$

R is the bend radius of the curved section, r is the mean radius of the pipe, t is the wall thickness of the pipe, and ν is Poisson's ratio. Changes in the flexibility factor are introduced by varying the wall thickness of the pipe.

The pipeline is modeled with three different element types: S4 shell elements, ELBOW31 elbow elements, and PIPE31 pipe elements. The S4 shell element model consists of a relatively fine mesh of 40 elements about the circumference and 75 elements along the length. This mesh is deemed fine enough to capture the true response of the pipeline accurately, although no mesh convergence studies are performed. Two analyses are conducted with the shell mesh: one with automatic stabilization using a constant damping factor (see "Automatic stabilization of static problems with a constant damping factor" in "Solving nonlinear problems," Section 7.1.1 of the Abaqus Analysis User's Guide), and one with adaptive automatic stabilization (see "Adaptive automatic stabilization scheme" in "Solving nonlinear problems," Section 7.1.1 of the Abaqus Analysis User's Guide). The pipe and elbow element meshes consist of 75 elements along the length; the analyses with these element types do not use automatic stabilization.

The results of the shell element model with automatic stabilization using a constant damping factor are taken as the reference solution. The reaction force at the tip of the pipeline is used to evaluate the effectiveness of the pipe and elbow elements. In addition, the ovalization values of the pipeline cross-section predicted by the elbow element models are compared.

The elbow elements are tested with 0, 3, and 6 Fourier modes, respectively. In general, elbow element accuracy improves as more modes are used, although the computational cost increases accordingly. In addition to standard pipe elements, tests are performed on pipe elements with a special flexibility knockdown factor. Flexibility knockdown factors (Dodge and Moore, 1972) are corrections to the bending stiffness based upon linear semianalytical results. They are applied to simple beam elements in an attempt to capture the global effects of ovalization. The knockdown factor is implemented in the PIPE31 elements by scaling the true thickness by the flexibility factor; this is equivalent to scaling the moment of inertia of the pipe element by $1/k$.

Results and discussion

The results obtained with the shell element model with automatic stabilization using a constant damping factor are taken as the reference solution. Very similar results are obtained with the same mesh using the adaptive automatic stabilization scheme.

The tip reaction forces due to the inward prescribed displacements for the various analysis models are shown in Figure 1.1.3–2. The results are normalized with respect to those obtained with the shell model. The results obtained with the ELBOW31 element model with 6 Fourier modes show excellent agreement with the reference solution over the entire range of flexibility factors considered in this study. The remaining four models generally exhibit excessively stiff response for all values of k. The PIPE31 element model, which uses the flexibility knockdown factor, shows a relatively constant error of about

20% over the entire range of flexibility factors. The 0-mode ELBOW31 element model and the PIPE31 element model without the knockdown factor produce very similar results for all values of k.

The normalized tip reaction forces due to the outward unit displacement for the various analysis models are shown in Figure 1.1.3–3. Again, the results obtained with the 6-mode ELBOW31 element model compare well with the reference shell solution. The 0-mode and 3-mode ELBOW31 and the PIPE31 (without the flexibility knockdown factor) element models exhibit overly stiff response. The PIPE31 element model with the knockdown factor has a transition region near $k = 1.5$, where the response changes from being too stiff to being too soft. The results in Abaqus/Explicit for pipe elements are consistent with those obtained in Abaqus/Standard.

Figure 1.1.3–4 and Figure 1.1.3–5 illustrate the effect of the number of included Fourier modes (0, 3, and 6) on the ability of the elbow elements to model the ovalization in the pipebend accurately in both load cases considered in this study. By definition, the 0-mode model cannot ovalize, which accounts for its stiff response. The 3-mode and the 6-mode models show significant ovalization in both loading cases. Figure 1.1.3–6 compares the ovalization of the 6-mode model in the opened and closed deformation states. It clearly illustrates that when the ends of the pipe are displaced inward (closing mode), the height of the pipe's cross-section gets smaller, thereby reducing the overall stiffness of the pipe; the reverse is true when the pipe ends are displaced outward: the height of the pipe's cross-section gets larger, thereby increasing the pipe stiffness. These three figures were produced with the aid of the elbow element postprocessing program **`felbow.f`** ("Creation of a data file to facilitate the postprocessing of elbow element results: FELBOW," Section 15.1.6), written in FORTRAN. The postprocessing programs **`felbow.C`** ("A C++ version of FELBOW," Section 10.15.6 of the Abaqus Scripting User's Guide) and **`felbow.py`** ("An Abaqus Scripting Interface version of FELBOW," Section 9.10.12 of the Abaqus Scripting User's Guide), written in C++ and Python, respectively, are also available for generating the data for these figures. The user must ensure that the output variables are written to the output database to use these two programs.

Parametric study

The performance of the pipe and elbow elements investigated in this example is analyzed conveniently in a parametric study using the Python scripting capabilities of Abaqus ("Scripting parametric studies," Section 20.1.1 of the Abaqus Analysis User's Guide). We perform a parametric study in which eight analyses are executed automatically for each of the three element types (S4, ELBOW31, and PIPE31) discussed above; these parametric studies correspond to wall thickness values ranging from 0.03175 cm to 0.2032 cm (0.0125 inches to 0.08 inches).

The Python script file elbowtest.psf is used to perform the parametric study. The function **`customTable`** (shown below) is an example of advanced Python scripting (Lutz and Ascher, 1999), which is used in elbowtest.psf. Such advanced scripting is not routinely needed, but in this case a dependent variable such as k cannot be included as a column of data in an XYPLOT file. **`customTable`** is designed to overcome this limitation by taking an XYPLOT file from the parametric study and converting it into a new file of reaction forces versus flexibility factors (k).

```
##############################################################
#
def customTable(file1, file2):
```

```
        for line in file1.readlines():
            print line
            nl = string.split(line,',')

            disp = float(nl[0])
            bend_radius = float(nl[1])
            wall_thick  = float(nl[2])
            outer_pipe_radius = float(nl[3])
            poisson = float(nl[4])
            rf = float(nl[6])

            mean_rad = outer_pipe_radius - wall_thick/2.0
            k = bend_radius*wall_thick/mean_rad**2
            k = k/sqrt(1.e0 - poisson**2)
            k = 1.66e0/k

            outputstring = str(k) + ', ' + str(rf) + '\n'
            file2.write(outputstring)
    #
    ##################################################################
```

Input files

elbowtest_shell.inp	S4 model.
elbowtest_shell_stabil_adap.inp	S4 model with adaptive stabilization.
elbowtest_elbow0.inp	ELBOW31 model with 0 Fourier modes.
elbowtest_elbow3.inp	ELBOW31 model with 3 Fourier modes.
elbowtest_elbow6.inp	ELBOW31 model with 6 Fourier modes.
elbowtest_pipek.inp	PIPE31 model with the flexibility knockdown factor.
elbowtest_pipek_xpl.inp	PIPE31 model with the flexibility knockdown factor in Abaqus/Explicit.
elbowtest_pipe.inp	PIPE31 model without the flexibility knockdown factor.
elbowtest_pipe_xpl.inp	PIPE31 model without the flexibility knockdown factor in Abaqus/Explicit.
elbowtest.psf	Python script file for the parametric study.

References

- 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, 1972.
- Lutz, M., and D. Ascher, *Learning Python,* O’Reilly, 1999.

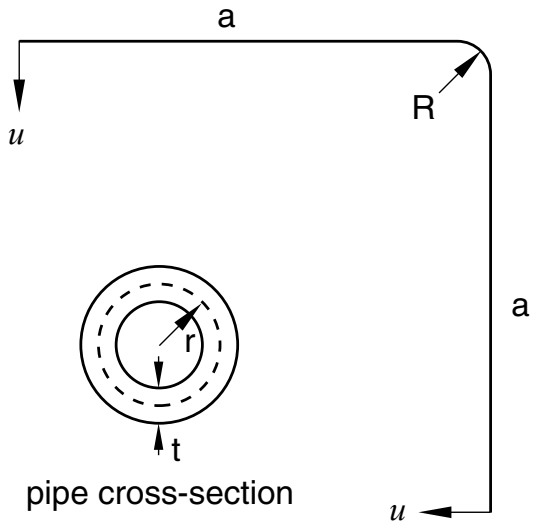


Figure 1.1.3–1 Pipeline geometry with inward prescribed tip displacements.

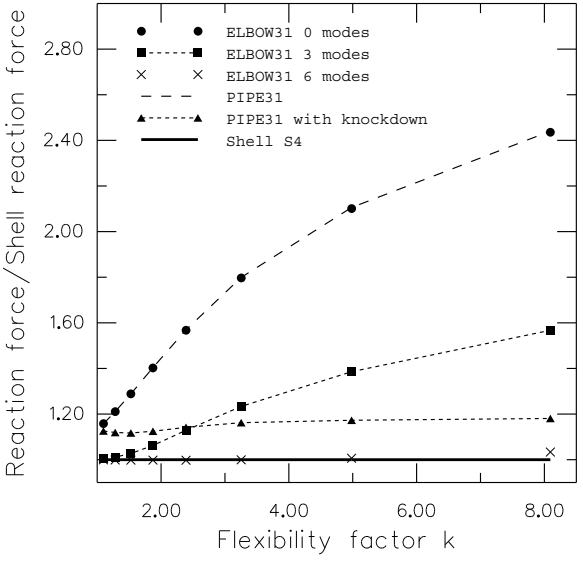


Figure 1.1.3–2 Normalized tip reaction force: closing displacement case.

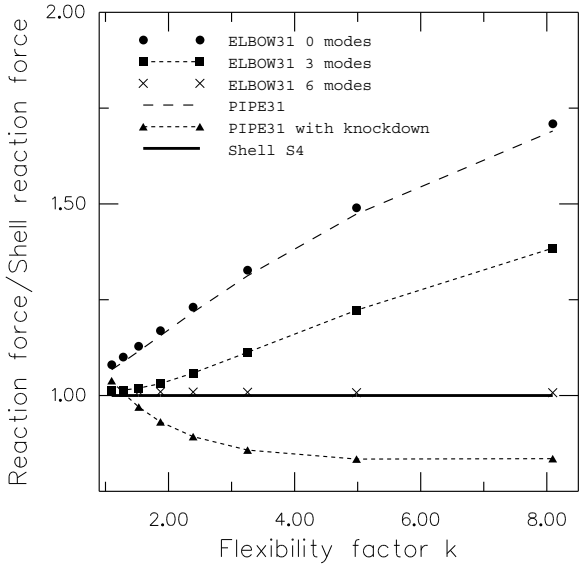


Figure 1.1.3–3 Normalized tip reaction force: opening displacement case.

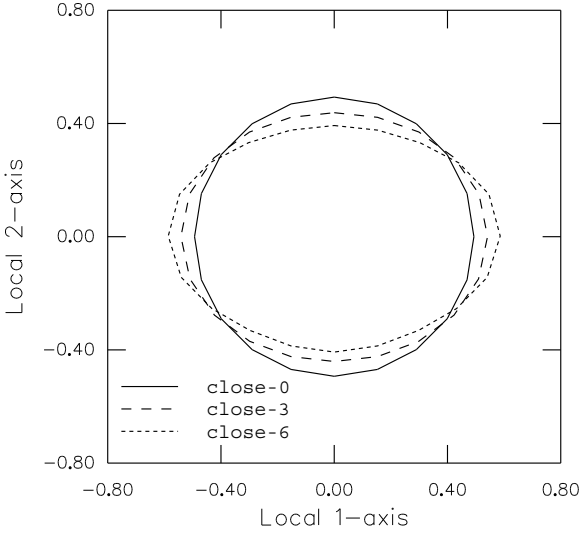


Figure 1.1.3–4 Ovalization of the ELBOW31 cross-sections for 0, 3, and 6 Fourier modes: closing displacement case.

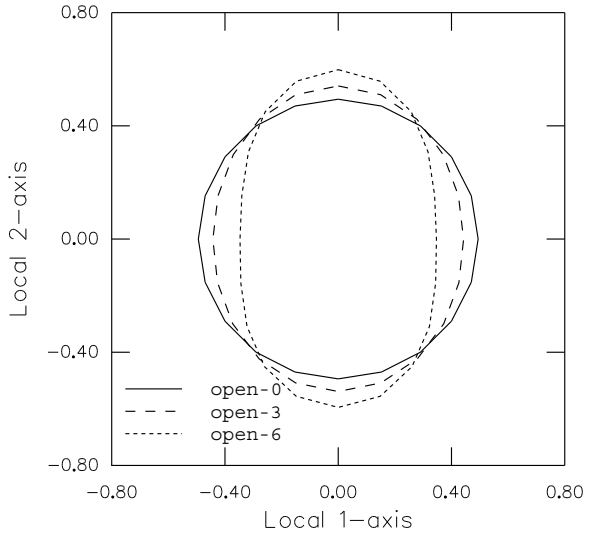


Figure 1.1.3–5 Ovalization of the ELBOW31 cross-sections for 0, 3, and 6 Fourier modes: opening displacement case.

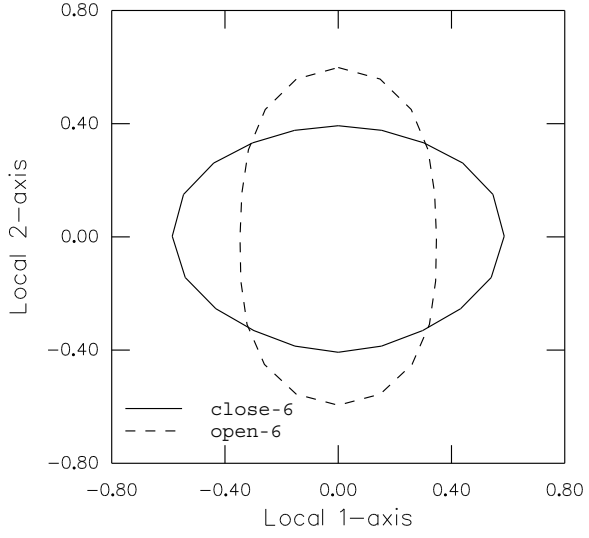


Figure 1.1.3–6 Ovalization of the ELBOW31 cross-sections for 6 Fourier modes: opening and closing displacement cases.

1.1.4 INDENTATION OF AN ELASTOMERIC FOAM SPECIMEN WITH A HEMISPHERICAL PUNCH

Products: Abaqus/Standard Abaqus/Explicit Abaqus/Design

In this example we consider a cylindrical specimen of an elastomeric foam, indented by a rough, rigid, hemispherical punch. Examples of elastomeric foam materials are cellular polymers such as cushions, padding, and packaging materials. This problem illustrates a typical application of elastomeric foam materials when used in energy absorption devices. The same geometry as the crushable foam model of "Simple tests on a crushable foam specimen," Section 3.2.7 of the Abaqus Benchmarks Guide, is used but with a slightly different mesh. Design sensitivity analysis is carried out for a shape design parameter and a material design parameter to illustrate the usage of design sensitivity analysis for a problem involving contact.

Geometry and model

The axisymmetric model (135 linear 4-node elements) analyzed is shown in Figure 1.1.4–1. The mesh refinement is biased toward the center of the foam specimen where the largest deformation is expected. The foam specimen has a radius of 600 mm and a thickness of 300 mm. The punch has a radius of 200 mm. The bottom nodes of the mesh are fixed, while the outer boundary is free to move.

A contact pair is defined between the punch, which is modeled by a rough spherical rigid surface, and a slave surface composed of the faces of the axisymmetric elements in the contact region. The friction coefficient between the punch and the foam is 0.8. A point mass of 200 kg representing the weight of the punch is attached to the rigid body reference node. The model is analyzed in both Abaqus/Standard and Abaqus/Explicit.

Material

The elastomeric foam material is defined using experimental test data. The uniaxial compression and simple shear data stress-strain curves are shown in Figure 1.1.4–2. Other available test data options are biaxial test data, planar test data, and volumetric test data. The test data are defined in terms of nominal stress and nominal strain values. Abaqus performs a nonlinear least-squares fit of the test data to determine the hyperfoam coefficients μ_i, α_i, and β_i.

Details of the formulation and usage of the hyperfoam model are given in "Hyperelastic behavior in elastomeric foams," Section 22.5.2 of the Abaqus Analysis User's Guide; "Hyperelastic material behavior," Section 4.6.1 of the Abaqus Theory Guide; and "Fitting of hyperelastic and hyperfoam constants," Section 4.6.2 of the Abaqus Theory Guide. "Fitting of elastomeric foam test data," Section 3.1.5 of the Abaqus Benchmarks Guide, illustrates the fitting of elastomeric foam test data to derive the hyperfoam coefficients.

For the material used in this example, β_i is zero, since the effective Poisson's ratio, ν, is zero as specified by the POISSON parameter. The order of the series expansion is chosen to be $N = 2$ since this fits the test data with sufficient accuracy. It also provides a more stable model than the $N = 3$ case.

The viscoelastic properties in Abaqus are specified in terms of a relaxation curve (shown in Figure 1.1.4–3) of the normalized modulus $M(t)/M_0$, where $M(t)$ is the shear or bulk modulus as a function of time and M_0 is the instantaneous modulus as determined from the hyperfoam model. This requires Abaqus to calculate the Prony series parameters from data taken from shear and volumetric relaxation tests. The relaxation data are specified as part of the definition of shear test data but actually apply to both shear and bulk moduli when used in conjunction with the hyperfoam model. Abaqus performs a nonlinear least-squares fit of the relaxation data to a Prony series to determine the coefficients, $\overline{g}_i^P$, and the relaxation periods, τ_i. A maximum order of 2 is used for fitting the Prony series. If creep data are available, you can specify normalized creep compliance data to compute the Prony series parameters.

A rectangular material orientation is defined for the foam specimen, so stress and strain are reported in material axes that rotate with the element deformation. This is especially useful when looking at the stress and strain values in the region of the foam in contact with the punch in the direction normal to the punch (direction "22").

The rough surface of the punch is modeled by specifying a friction coefficient of 0.8 for the contact surface interaction.

Procedure and loading definitions

Two cases are analyzed. In the first case the punch is displaced statically downward to indent the foam, and the reaction force-displacement relation is measured for both the purely elastic and viscoelastic cases. In the second case the punch statically indents the foam through gravity loading and is then subjected to impulsive loading. The dynamic response of the punch is sought as it interacts with the viscoelastic foam.

Case 1

In Abaqus/Standard the punch is displaced downward by a prescribed displacement boundary condition in the first step, indenting the foam specimen by a distance of 250 mm. Geometric nonlinearity should be accounted for in this step, since the response involves large deformation. In the second step the punch is displaced back to its original position. Two analyses are performed—one using the static procedure for both steps and the other using the quasi-static procedure for both steps. During a static step the material behaves purely elastically, using the properties specified with the hyperfoam model. The quasi-static, direct-integration implicit dynamic, or fully coupled thermal-stress procedure must be used to activate the viscoelastic behavior. In this case the punch is pushed down in a period of one second and then moved back up again in one second. The accuracy of the creep integration in the quasi-static procedure can be controlled and is typically calculated by dividing an acceptable stress error tolerance by a typical elastic modulus. In this problem we estimate a stress error tolerance of about 0.005 MPa and use the initial elastic modulus, $\mathrm{E}_0 = 2\sum \mu_i = 0.34$, to determine an accuracy tolerance of 0.01.

In Abaqus/Explicit the punch is also displaced downward by a prescribed displacement boundary condition, indenting the foam by a depth of 250 mm. The punch is then lifted back to its original position. In this case the punch is modeled as either an analytical rigid surface or a discrete rigid surface defined with RAX2 elements. The entire analysis runs for 2 seconds. The actual time period of the analysis

is large by explicit dynamic standards. Hence, to reduce the computational time, the mass density of the elements is increased artificially to increase the stable time increment without losing the accuracy of the solution. The mass scaling factor is set to 10, which corresponds to a speedup factor of $\sqrt{10}$. The reaction force-displacement relation is measured for both the elastic and viscoelastic cases.

Case 2

The Abaqus/Standard analysis is composed of three steps. The first step is a quasi-static step, where gravity loading is applied to the point mass of the punch. The gravity loading is ramped up in two seconds, and the step is run for a total of five seconds to allow the foam to relax fully. In the second step, which is a direct-integration implicit dynamic step, an impulsive load in the form of a half sine wave amplitude with a peak magnitude of 5000 N is applied to the punch over a period of one second. In the third step, also a direct-integration implicit dynamic step, the punch is allowed to move freely until the vibration is damped out by the viscoelastic foam. For a dynamic analysis with automatic time incrementation, the value of the half-increment residual tolerance for the direct-integration implicit dynamic procedure controls the accuracy of the time integration. For systems that have significant energy dissipation, such as this heavily damped model, a relatively high value of this tolerance can be chosen. We choose the tolerance to be 100 times a typical average force that we estimate (and later confirm from the analysis results) to be on the order of 50 N. Thus, the half-increment residual tolerance is 5000 N. For the second direct-integration implicit dynamic step we bypass calculation of initial accelerations at the beginning of the step, since there is no sudden change in load to create a discontinuity in the accelerations.

In the Abaqus/Explicit analysis the punch indents the foam quasi-statically through gravity loading and is then subjected to an impulsive loading. In the first step gravity loading is applied to the point mass of the punch, and the foam is allowed to relax fully. The mass scaling factor in this step is set to 10. In the second step a force in the form of a half sine wave is applied to the punch, and the dynamic response of the punch is obtained as it interacts with the viscoelastic foam. In the third step the load is removed, and the punch is allowed to move freely. Mass scaling is not used in Steps 2 and 3 since the true dynamic response is sought.

Design sensitivity analysis

For the design sensitivity analysis (DSA) carried out with static steps in Abaqus/Standard, the hyperfoam material properties are given using direct input of coefficients based on the test data given above. For $N = 2$, the coefficients are $\mu_1 = 0.16245$, $\mu_2 = 3.59734\text{E}{-}05$, $\alpha_1 = 8.89239$, $\alpha_2 = -4.52156$, and $\beta_i = 0.0$. Since the quasi-static procedure is not supported for DSA, it is replaced with the static procedure and the viscoelastic material behavior is removed. In addition, since a more accurate tangent stiffness leads to improved sensitivity results, the solution controls are used to tighten the residual tolerance.

The material parameter μ_1 is chosen as one of the design parameters. The other (shape) design parameter used for design sensitivity analysis, L, represents the thickness of the foam at the free end (see Figure 1.1.4–1). The z-coordinates of the nodes on the top surface are assumed to depend on L via the equation $z = 300 + (L - 300)r/600$. The r-coordinates are considered to be independent of L. To define this dependency in Abaqus, the gradients of the coordinates with respect to L

$$\frac{dr}{dL} = 0$$

$$\frac{dz}{dL} = \frac{r}{600}$$

are given as part of the specification of parameter shape variation.

Results and discussion

This problem tests the hyperfoam material model in Abaqus but does not provide independent verification of the model. The results for all analyses are discussed in the following paragraphs.

Case 1

Deformation and contour plots for oriented S22 stress and LE22 strain are shown for the viscoelastic foam in Figure 1.1.4–4 through Figure 1.1.4–6 for the Abaqus/Standard analysis and Figure 1.1.4–7 through Figure 1.1.4–9 for the Abaqus/Explicit analysis. Even though the foam has been subjected to large strains, only moderate distortions occur because of the zero Poisson's ratio. The maximum logarithmic strain is on the order of -1.85, which is equivalent to a stretch of $\lambda = e^{-1.85} = 0.16$ or a nominal compressive strain of 84%, indicating severe compression of the foam.

Figure 1.1.4–10 shows a comparison of the punch reaction force histories obtained with Abaqus/Standard and Abaqus/Explicit. In the viscoelastic case the stresses relax during loading and, consequently, lead to a softer response than in the purely elastic case. A comparison of the force-displacement responses obtained with Abaqus/Standard and Abaqus/Explicit is shown in Figure 1.1.4–11. The purely elastic material is reversible, while the viscoelastic material shows hysteresis.

Case 2

Figure 1.1.4–12 shows various displaced configurations during the Case 2 analysis for Abaqus/Standard and Abaqus/Explicit. Displacement, velocity, and acceleration histories for the punch are shown in Figure 1.1.4–13, Figure 1.1.4–14, and Figure 1.1.4–15, respectively. The displacement is shown to reach a steady value at the stress relaxation stage, followed by a severe drop due to the impulsive dynamic load. This is followed by a rebound and then finally by a rapid decay of the subsequent oscillations due to the strong damping provided by the viscoelasticity of the foam.

Abaqus/Design

Figure 1.1.4–16 and Figure 1.1.4–17 show the contours of sensitivity of the displacement in the z-direction to the design parameters L and μ_1, respectively. Figure 1.1.4–18 and Figure 1.1.4–19 show the contours of sensitivity of S22 to the design parameters L and μ_1, respectively. To provide an independent assessment of the results provided by Abaqus, sensitivities were computed using the overall finite difference (OFD) technique. The central difference method with a perturbation size of

0.1% of the value of the design parameter was used to obtain the OFD results. Table 1.1.4–1 shows that the sensitivities computed using Abaqus compare well with the overall finite difference results.

Input files

indentfoam_std_elast_1.inp	Case 1 of the Abaqus/Standard example using elastic properties of the foam, which is statically deformed in two *STATIC steps.
indentfoam_std_elast_1_st.inp	Case 1 of the Abaqus/Standard example (CAX4R elements with hourglass control based on total stiffness) using elastic properties of the foam, which is statically deformed in two *STATIC steps.
indentfoam_std_elast_1_eh.inp	Case 1 of the Abaqus/Standard example (CAX4R elements with enhanced hourglass control) using elastic properties of the foam, which is statically deformed in two *STATIC steps.
indentfoam_std_visco_1.inp	Case 1 of the Abaqus/Standard example using viscoelastic properties of the foam, which is statically deformed in two *VISCO steps.
indentfoam_std_visco_1_st.inp	Case 1 of the Abaqus/Standard example (CAX4R elements with hourglass control based on total stiffness) using viscoelastic properties of the foam, which is statically deformed in two *VISCO steps.
indentfoam_std_visco_1_eh.inp	Case 1 of the Abaqus/Standard example (CAX4R elements with enhanced hourglass control) using viscoelastic properties of the foam, which is statically deformed in two *VISCO steps.
indentfoam_std_visco_2.inp	Case 2 of the Abaqus/Standard example using viscoelastic properties of the foam.
indentfoam_std_visco_2_surf.inp	Case 2 of the Abaqus/Standard example using viscoelastic properties of the foam. Surface-to-surface contact is utilized.
indentfoam_xpl_elast_1.inp	Case 1 of the Abaqus/Explicit example using elastic properties of the foam with the punch modeled as an analytical rigid surface.
indentfoam_xpl_elast_1_subcyc.inp	Case 1 of the Abaqus/Explicit example using elastic properties of the foam with the punch modeled as an analytical rigid surface using the subcycling feature.
indentfoam_xpl_elast_fac_1.inp	Case 1 of the Abaqus/Explicit example using elastic properties of the foam with the punch modeled as a faceted rigid surface.

indentfoam_xpl_visco_1.inp	Case 1 of the Abaqus/Explicit example using viscoelastic properties of the foam with the punch modeled as an analytical rigid surface.
indentfoam_xpl_visco_2.inp	Case 2 of the Abaqus/Explicit example using viscoelastic properties of the foam with the punch modeled as an analytical rigid surface.
indentfoamhemipunch_dsa.inp	Design sensitivity analysis.

Table 1.1.4–1 Comparison of normalized sensitivities at the end of the analysis computed using Abaqus and the overall finite difference (OFD) method.

Normalized sensitivity	Abaqus	OFD
$\frac{\mu_1}{\lvert \mathrm{S22} \rvert^{\max}} \left\lvert \frac{d\mathrm{S22}}{d\mu_1} \right\rvert_{\max}$	0.4921	0.4922
$\frac{L}{\lvert \mathrm{S22} \rvert^{\max}} \left\lvert \frac{d\mathrm{S22}}{dL} \right\rvert_{\max}$	1.085	1.104
$\frac{\mu_1}{\lvert u_2 \rvert^{\max}} \left\lvert \frac{du_2}{d\mu_1} \right\rvert_{\max}$	0.006925	0.006927
$\frac{L}{\lvert u_2 \rvert^{\max}} \left\lvert \frac{du_2}{dL} \right\rvert_{\max}$	0.4059	0.4120
$\frac{\mu_1}{\mathrm{CPRESS}^{\max}} \left\lvert \frac{d\mathrm{CPRESS}}{d\mu_1} \right\rvert_{\max}$	0.5084	0.5085
$\frac{L}{\mathrm{CPRESS}^{\max}} \left\lvert \frac{d\mathrm{CPRESS}}{dL} \right\rvert_{\max}$	0.3252	0.3207

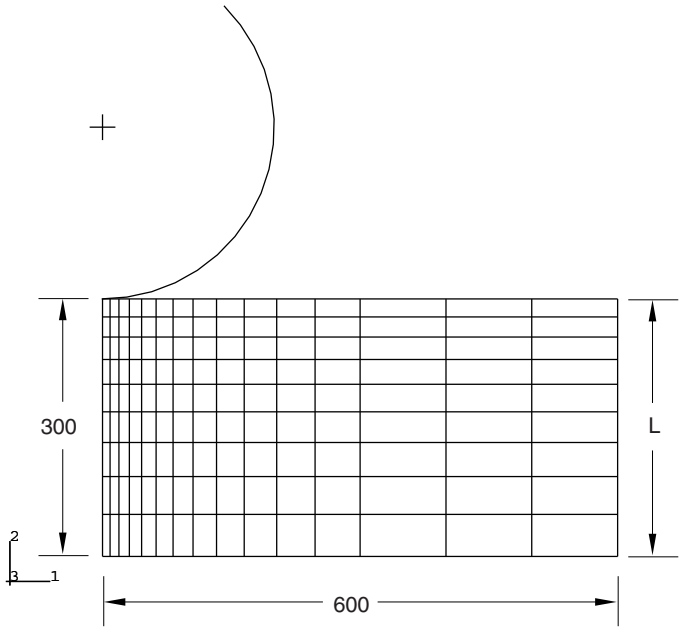


Figure 1.1.4–1 Model for foam indentation by a spherical punch.

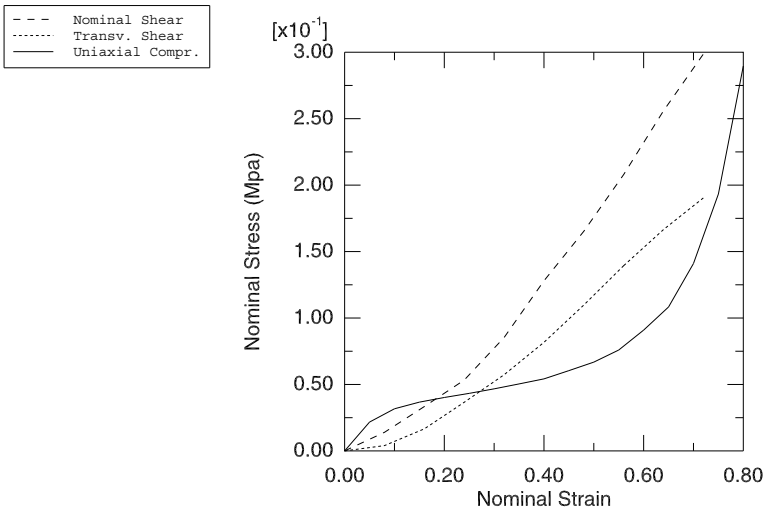


Figure 1.1.4–2 Elastomeric foam stress-strain curves.

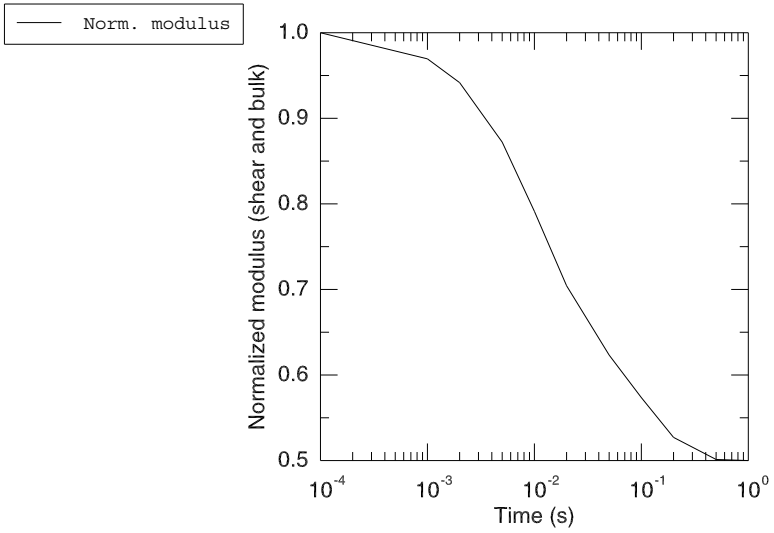


Figure 1.1.4–3 Elastic modulus relaxation curve.

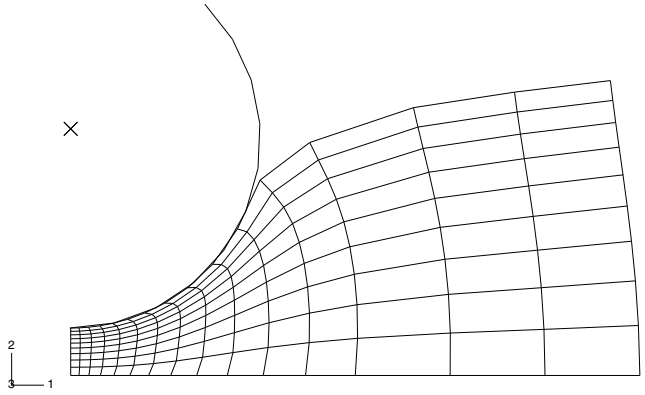


Figure 1.1.4–4 Maximum deformation of viscoelastic foam: Case 1, Abaqus/Standard.

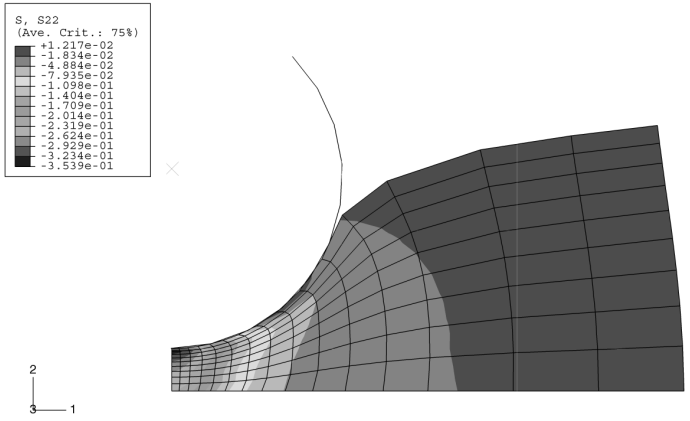


Figure 1.1.4–5 S22 contour plot of viscoelastic foam: Case 1, Abaqus/Standard.

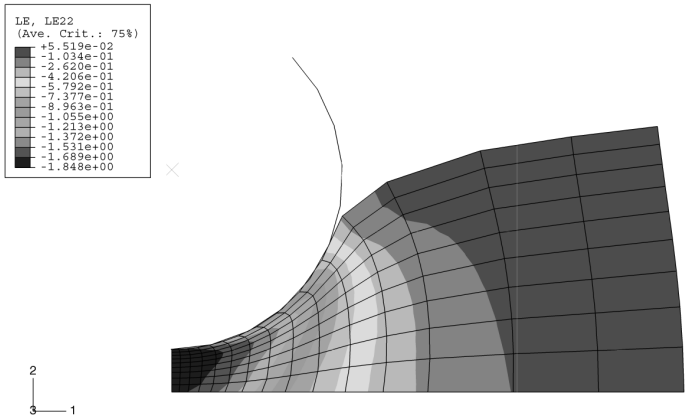


Figure 1.1.4–6 LE22 contour plot of viscoelastic foam: Case 1, Abaqus/Standard.

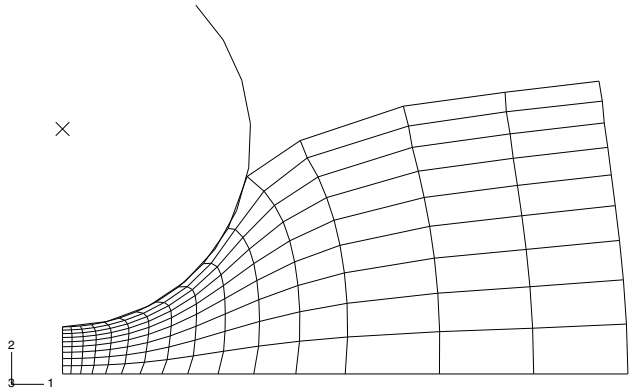


Figure 1.1.4–7 Deformed plot at 1.0 s: Case 1, Abaqus/Explicit.

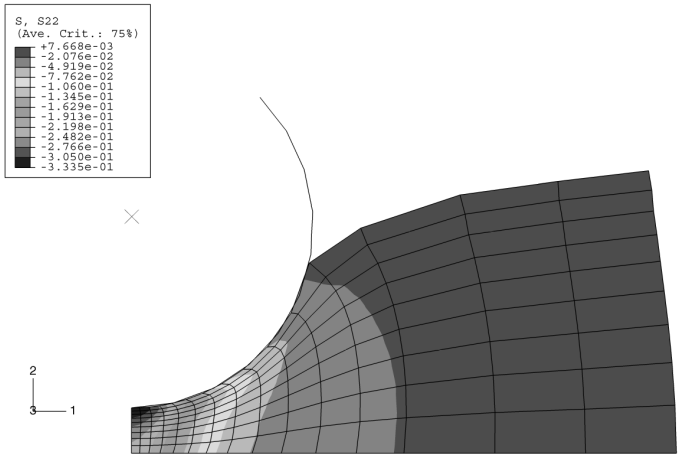


Figure 1.1.4–8 S22 contour plot of viscoelastic foam at 1.0 s: Case 1, Abaqus/Explicit.

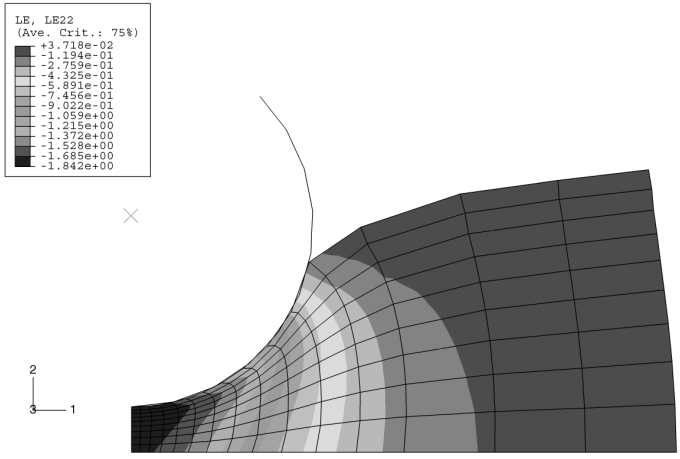


Figure 1.1.4–9 LE22 contour plot of viscoelastic foam at 1.0 s: Case 1, Abaqus/Explicit.

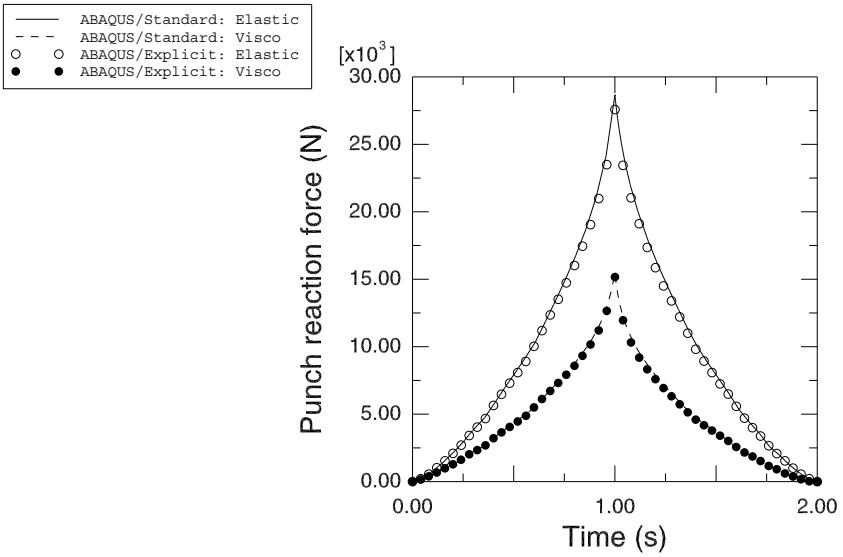


Figure 1.1.4–10 Punch reaction force history: Case 1.

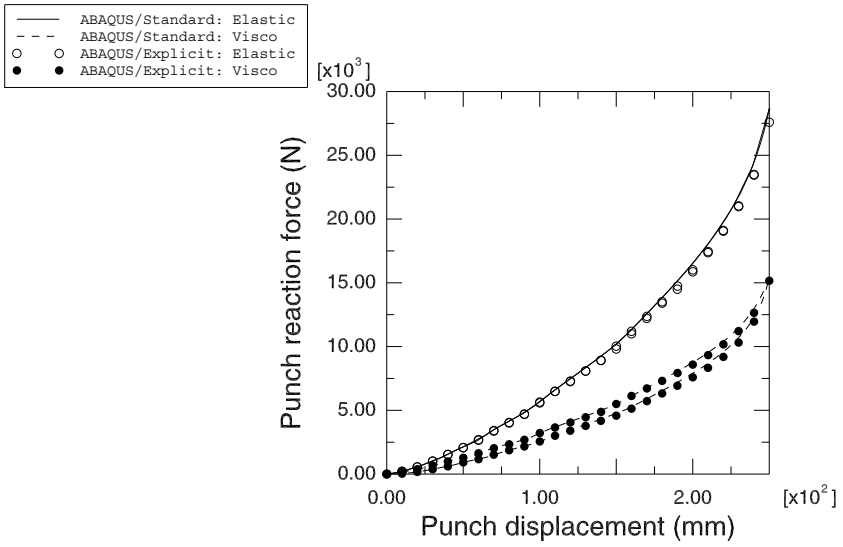


Figure 1.1.4–11 Punch reaction force versus displacement response (loading-unloading curves): Case 1.

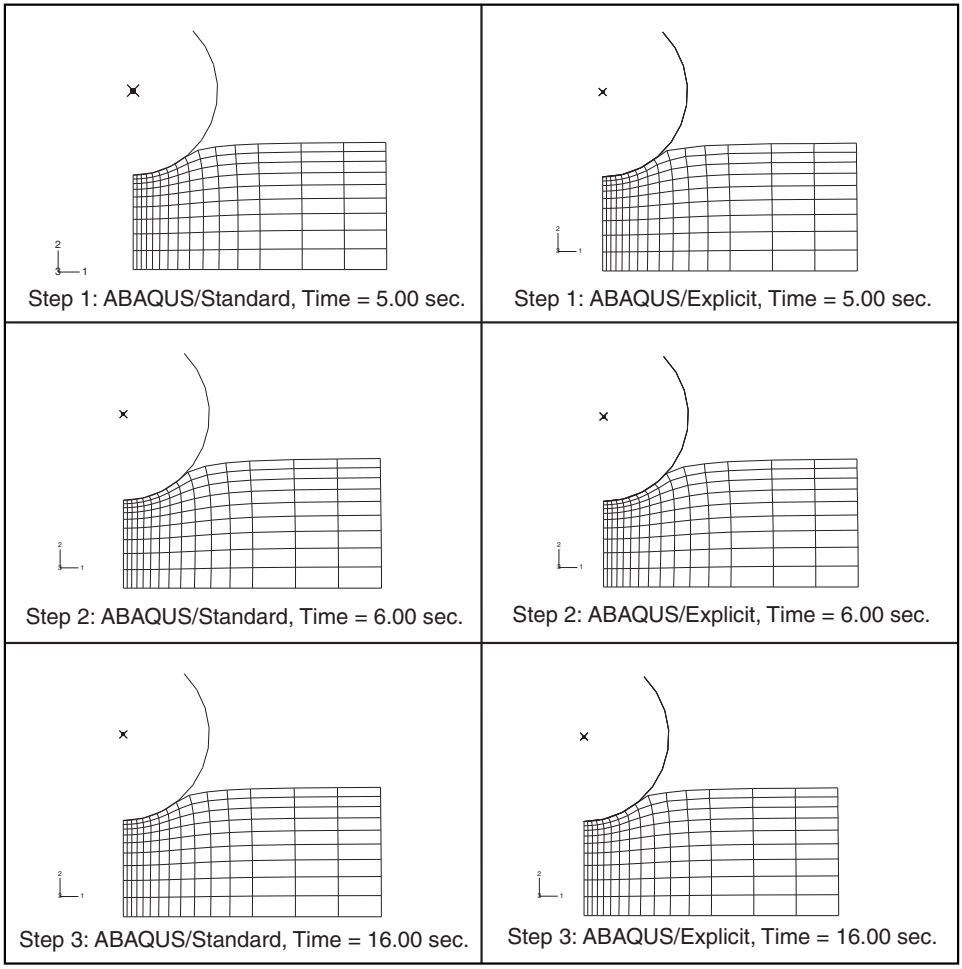


Figure 1.1.4–12 Deformed shape plots at the end of visco and dynamic steps: Case 2, Abaqus/Standard (left) and Abaqus/Explicit (right).

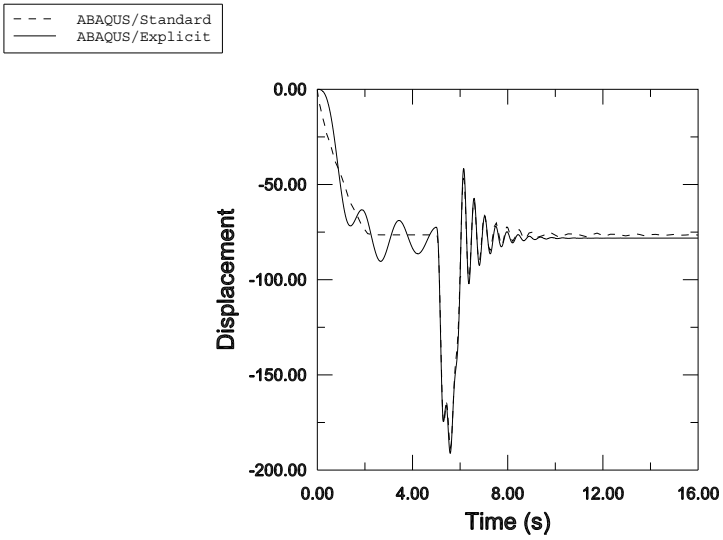


Figure 1.1.4–13 Displacement histories of the punch: Case 2, Abaqus/Standard and Abaqus/Explicit.

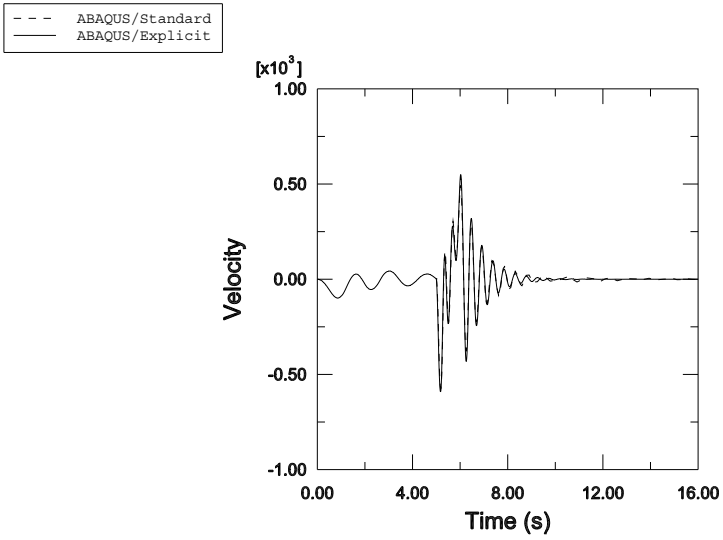


Figure 1.1.4–14 Velocity histories of the punch: Case 2, Abaqus/Standard and Abaqus/Explicit.

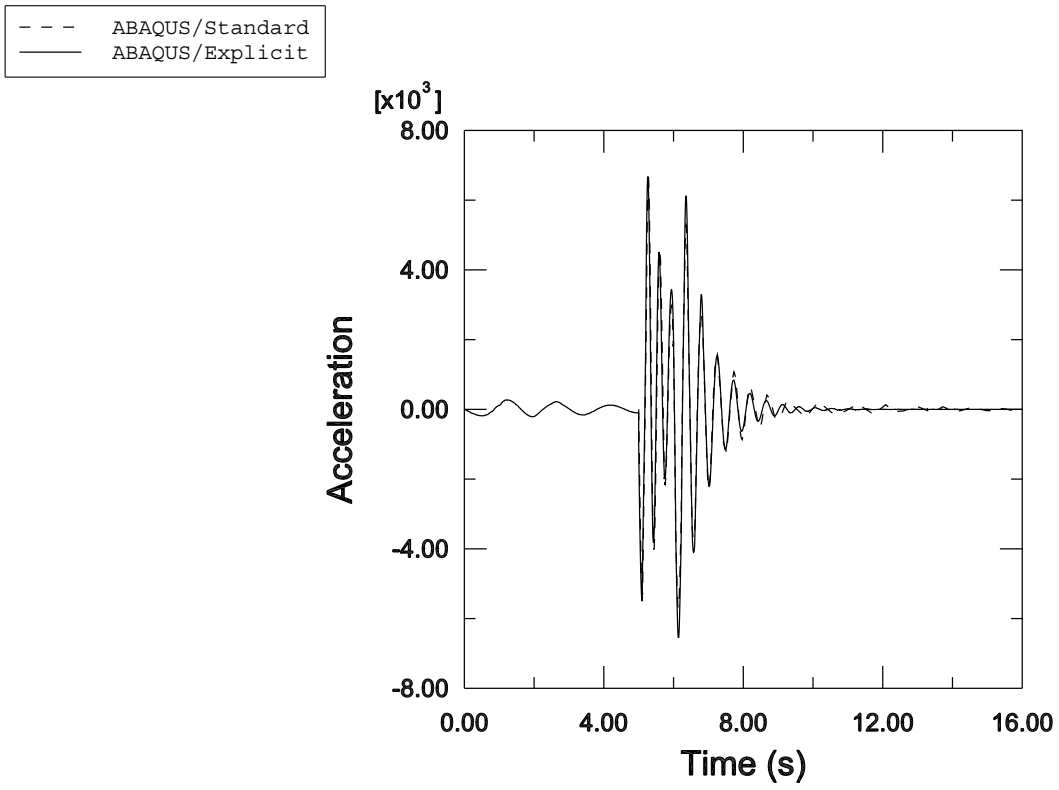


Figure 1.1.4–15 Acceleration histories of the punch: Case 2, Abaqus/Standard and Abaqus/Explicit.

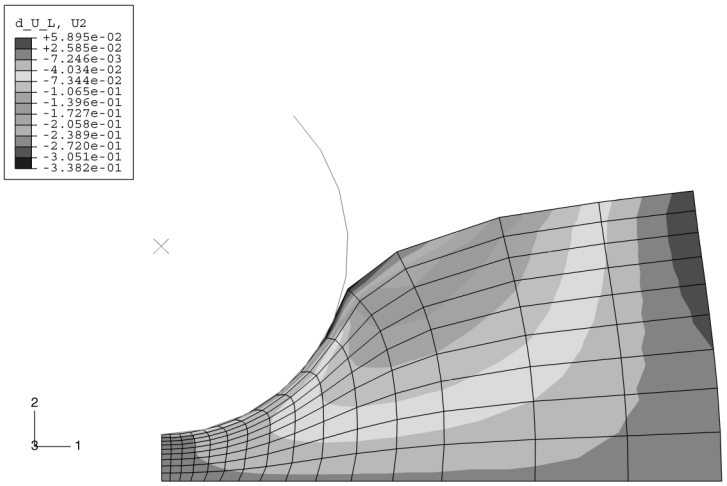


Figure 1.1.4–16 Sensitivities at the end of the analysis for displacement in the z-direction with respect to L.

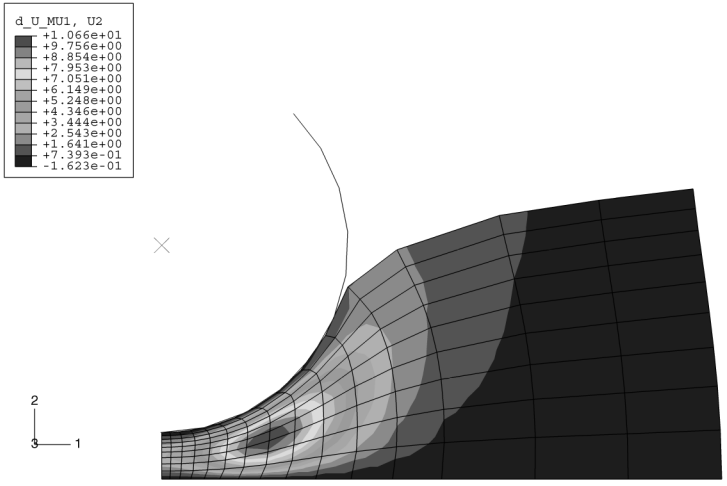


Figure 1.1.4–17 Sensitivities at the end of the analysis for displacement in the z-direction with respect to μ_1.

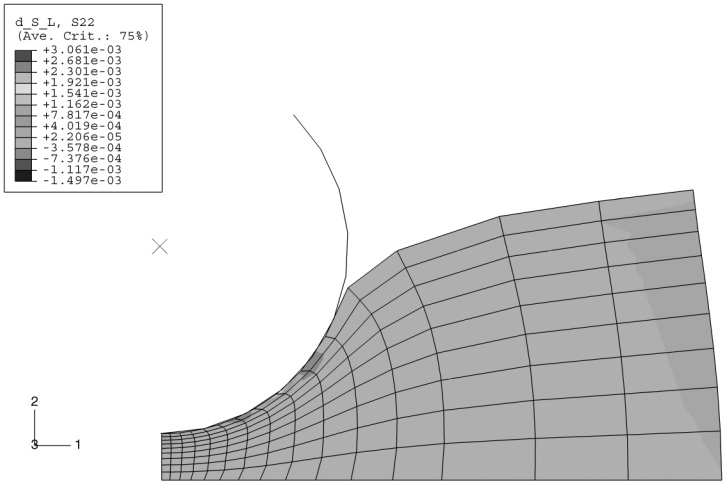


Figure 1.1.4–18 Sensitivities at the end of the analysis for stress S22 with respect to L.

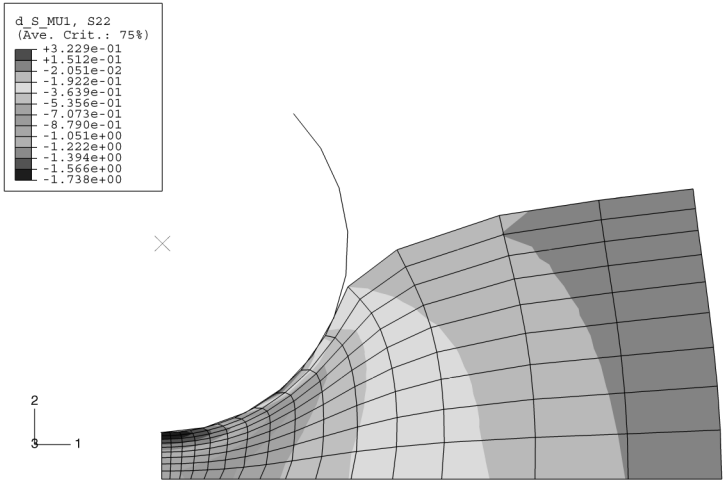


Figure 1.1.4–19 Sensitivities at the end of the analysis for stress S22 with respect to μ_1.

1.1.5 COLLAPSE OF A CONCRETE SLAB

Products: Abaqus/Standard Abaqus/Explicit

This problem examines the use of the smeared crack model ("Concrete smeared cracking," Section 23.6.1 of the Abaqus Analysis User's Guide) and the brittle cracking model ("Cracking model for concrete," Section 23.6.2 of the Abaqus Analysis User's Guide) for the analysis of reinforced concrete structures. The geometry of the problem is defined in Figure 1.1.5–1. A square slab is supported in the transverse direction at its four corners and loaded by a point load at its center. The slab is reinforced in two directions at 75% of its depth. The reinforcement ratio (volume of steel/volume of concrete) is 8.5×10^{-3} in each direction. The slab was tested experimentally by McNeice (1967) and has been analyzed by a number of workers, including Hand et al. (1973), Lin and Scordelis (1975), Gilbert and Warner (1978), Hinton et al. (1981), and Crisfield (1982).

Geometric modeling

Symmetry conditions allow us to model one-quarter of the slab. A 3×3 mesh of 8-node shell elements is used for the Abaqus/Standard analysis. No mesh convergence studies have been performed, but the reasonable agreement between the analysis results and the experimental data suggests that the mesh is adequate to predict overall response parameters with usable accuracy. Three different meshes are used in Abaqus/Explicit to assess the sensitivity of the results to mesh refinement: a coarse 6×6 mesh, a medium 12×12 mesh, and a fine 24×24 mesh of S4R elements. Nine integration points are used through the thickness of the concrete to ensure that the development of plasticity and failure is modeled adequately. The two-way reinforcement is modeled using layers of uniaxial reinforcement (rebars). Symmetry boundary conditions are applied on the two edges of the mesh, and the corner point is restrained in the transverse direction.

Material properties

The material data are given in Table 1.1.5–1. The material properties of concrete are taken from Gilbert and Warner (1978). Some of these data are assumed values, because they are not available for the concrete used in the experiment. The assumed values are taken from typical concrete data. The compressive behavior of concrete in the cracking model in Abaqus/Explicit is assumed to be linear elastic. This is a reasonable assumption for a case such as this problem, where the behavior of the structure is dominated by cracking resulting from tension in the slab under bending.

The modeling of the concrete-reinforcement interaction and the energy release at cracking is of critical importance to the response of a structure such as this once the concrete starts to crack. These effects are modeled in an indirect way by adding "tension stiffening" to the plain concrete model. This approach is described in "A cracking model for concrete and other brittle materials," Section 4.5.3 of the Abaqus Theory Guide; "Concrete smeared cracking," Section 23.6.1 of the Abaqus Analysis User's Guide; and "Cracking model for concrete," Section 23.6.2 of the Abaqus Analysis User's Guide. The simplest tension stiffening model defines a linear loss of strength beyond the cracking failure of the

concrete. In this example three different values for the strain beyond failure at which all strength is lost (5×10^{-4}, 1×10^{-3}, and 2×10^{-3}) are used to illustrate the effect of the tension stiffening parameters on the response.

Since the response is dominated by bending, it is controlled by the material behavior normal to the crack planes. The material's shear behavior in the plane of the cracks is not important. Consequently, the choice of shear retention has no significant influence on the results. In Abaqus/Explicit the shear retention chosen is exhausted at the same value of the crack opening at which tension stiffening is exhausted. In Abaqus/Standard full shear retention is used because it provides a more efficient numerical solution.

Solution control

Since considerable nonlinearity is expected in the response, including the possibility of unstable regimes as the concrete cracks, the modified Riks method is used with automatic incrementation in the Abaqus/Standard analysis. With the Riks method the load data and solution parameters serve only to give an estimate of the initial increment of load. In this case it seems reasonable to apply an initial load of 1112 N (250 lb) to the quarter-model for a total initial load on the structure of 4448 N (1000 lb). This can be accomplished by specifying a load of 22241 N (5000 lb) and an initial time increment of 0.05 out of a total time period of 1.0. The analysis is terminated when the central displacement reaches 25.4 mm (1 in).

Since Abaqus/Explicit is a dynamic analysis program and in this case we are interested in static solutions, the slab must be loaded slowly enough to eliminate any significant inertia effects. The slab is loaded in its center by applying a velocity that increases linearly from 0 to 2.0 in/second such that the center displaces a total of 1 inches in 1 second. This very slow loading rate ensures quasi-static solutions; however, it is computationally expensive. The CPU time required for this analysis can be reduced in one of two ways: the loading rate can be increased incrementally until it is judged that any further increase in loading rate would no longer result in a quasi-static solution, or mass scaling can be used (see "Explicit dynamic analysis," Section 6.3.3 of the Abaqus Analysis User's Guide). These two approaches are equivalent. Mass scaling is used here to demonstrate the validity of such an approach when it is used in conjunction with the brittle cracking model. Mass scaling is done by increasing the density of the concrete and the reinforcement by a factor of 100, thereby increasing the stable time increment for the analysis by a factor of 10 and reducing the computation time by the same amount while using the original slow loading rate. Figure 1.1.5–4 shows the load-deflection response of the slab for analyses using the 12×12 mesh with and without mass scaling. The mass scaling used does not affect the results significantly; therefore, all subsequent analyses are performed using mass scaling.

Results and discussion

Results for each analysis are discussed in the following sections.

Abaqus/Standard results

The numerical and experimental results are compared in Figure 1.1.5–2 on the basis of load versus deflection at the center of the slab. The strong effect of the tension stiffening assumption is very clear in that plot. The analysis with tension stiffening, such that the tensile strength is lost at a strain of 10^{-3}

beyond failure, shows the best agreement with the experiment. This analysis provides useful information from a design viewpoint. The failure pattern in the concrete is illustrated in Figure 1.1.5–3, which shows the predicted crack pattern on the lower surface of the slab at a central deflection of 7.6 mm (0.3 in).

Abaqus/Explicit results

Figure 1.1.5–5 shows the load-deflection response of the slab for the three different mesh densities using a tension stiffening value of 2×10^{-3}. Since the coarse mesh predicts a slightly higher limit load than the medium and fine meshes do and the limit loads for the medium and fine mesh analyses are very close, the tension stiffening study is performed using the medium mesh only.

The numerical (12×12 mesh) results are compared to the experimental results in Figure 1.1.5–6 for the three different values of tension stiffening. It is clear that the less tension stiffening used, the softer the load-deflection response is. A value of tension stiffening somewhere between the highest and middle values appears to match the experimental results best. The lowest tension stiffening value causes more sudden cracking in the concrete and, as a result, the response tends to be more dynamic than that obtained with the higher tension stiffening values.

Figure 1.1.5–7 shows the numerically predicted crack pattern on the lower surface of the slab for the medium mesh.

Input files

Abaqus/Standard input files

collapseconcslab_s8r.inp	S8R elements.
collapseconcslab_s9r5.inp	S9R5 elements.
collapseconcslab_postoutput.inp	*POST OUTPUT analysis.

Abaqus/Explicit input files

mcneice_1.inp	Coarse (6×6) mesh; tension stiffening = 2×10^{-3}.
mcneice_2.inp	Medium (12×12) mesh; tension stiffening = 2×10^{-3}.
mcneice_3.inp	Fine (24×24) mesh; tension stiffening = 2×10^{-3}.
mcneice_4.inp	Medium (12×12) mesh; tension stiffening = 1×10^{-3}.
mcneice_5.inp	Medium (12×12) mesh; tension stiffening = 5×10^{-4}.
mcneice_6.inp	Medium (12×12) mesh; tension stiffening = 2×10^{-3}; no mass scaling.

References

- Crisfield, M. A., "Variable Step-Length for Nonlinear Structural Analysis," Report 1049, Transport and Road Research Lab., Crowthorne, England, 1982.
- Gilbert, R. I., and R. F. Warner, "Tension Stiffening in Reinforced Concrete Slabs," Journal of the Structural Division, American Society of Civil Engineers, vol. 104, ST12, pp. 1885–1900, 1978.

- Hand, F. D., D. A. Pecknold, and W. C. Schnobrich, "Nonlinear Analysis of Reinforced Concrete Plates and Shells," Journal of the Structural Division, American Society of Civil Engineers, vol. 99, ST7, pp. 1491–1505, 1973.
- Hinton, E., H. H. Abdel Rahman, and O. C. Zienkiewicz, "Computational Strategies for Reinforced Concrete Slab Systems," International Association of Bridge and Structural Engineering Colloquium on Advanced Mechanics of Reinforced Concrete, pp. 303–313, 1981.
- Lin, C. S., and A. C. Scordelis, "Nonlinear Analysis of Reinforced Concrete Shells of General Form," Journal of the Structural Division, American Society of Civil Engineers, vol. 101, pp. 523–238, 1975.
- McNeice, A. M., "Elastic-Plastic Bending of Plates and Slabs by the Finite Element Method," Ph. D. Thesis, London University, 1967.

Table 1.1.5–1 Material properties for the McNeice slab.

Concrete properties:

Properties are taken from Gilbert and Warner (1978) if available in that paper.
Properties marked with a * are not available and are assumed values.

Young's modulus	28.6 GPa (4.15×10^6 lb/in^2)
Poisson's ratio	0.15

Uniaxial compression values:

Yield stress	20.68 MPa (3000 lb/in^2)*
Failure stress	37.92 MPa (5500 lb/in^2)
Plastic strain at failure	1.5×10^{-3} *
Ratio of uniaxial tension to compression failure stress	8.36×10^{-2}
Ratio of biaxial to uniaxial compression failure stress	1.16*
Cracking failure stress	459.8 lb/in^2 (3.17 MPa)
Density (before mass scaling)	2.246×10^{-4} lb s^2/in^4 (2400 kg/m^3)

"Tension stiffening" is assumed as a linear decrease of the stress to zero stress, at a strain of 5×10^{-4}, at a strain of 10×10^{-4}, or at a strain of 20×10^{-4}.

Steel (rebar) properties:

Young's modulus	200 GPa (29×10^6 lb/in^2)
Yield stress	345 MPa (50×10^3 lb/in^2)
Density (before mass scaling)	7.3×10^{-4} lb s^2/in^4 (7800 kg/m^3)

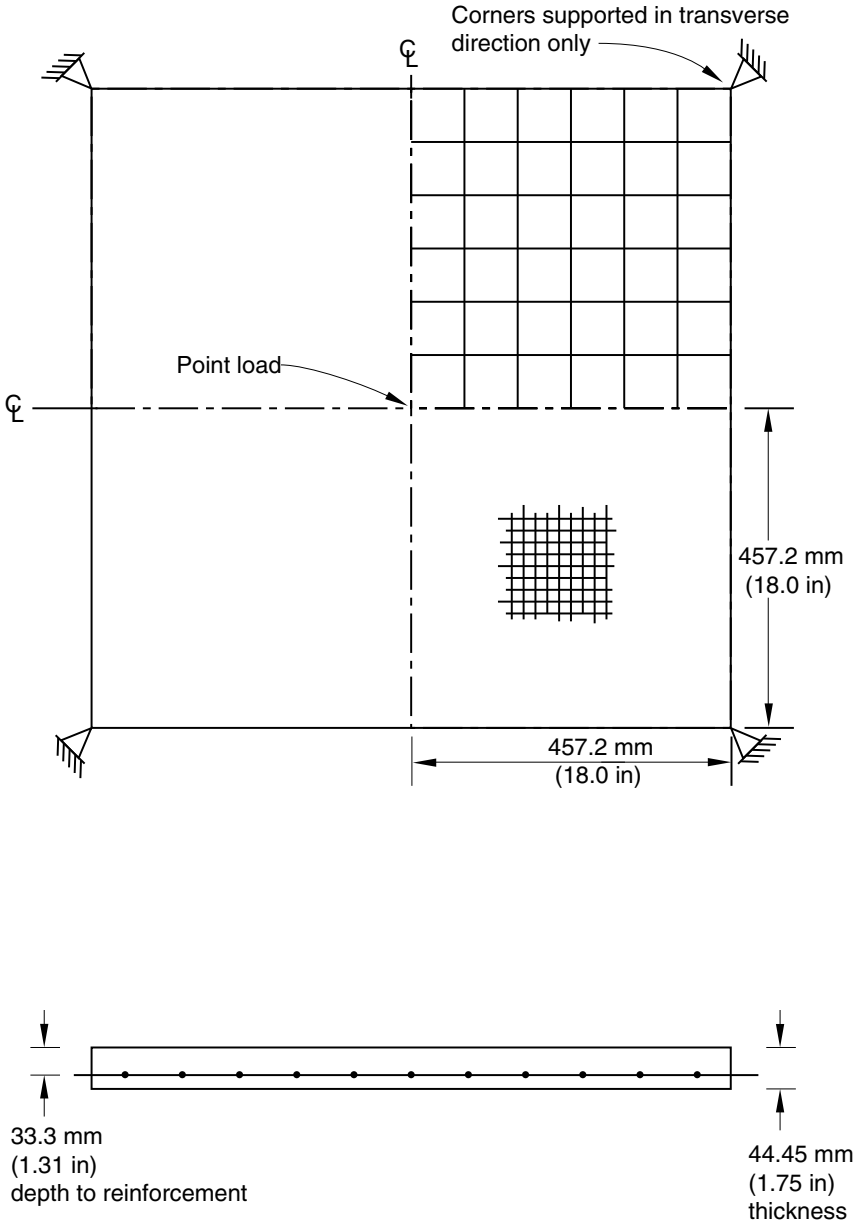


Figure 1.1.5–1 McNeice slab.

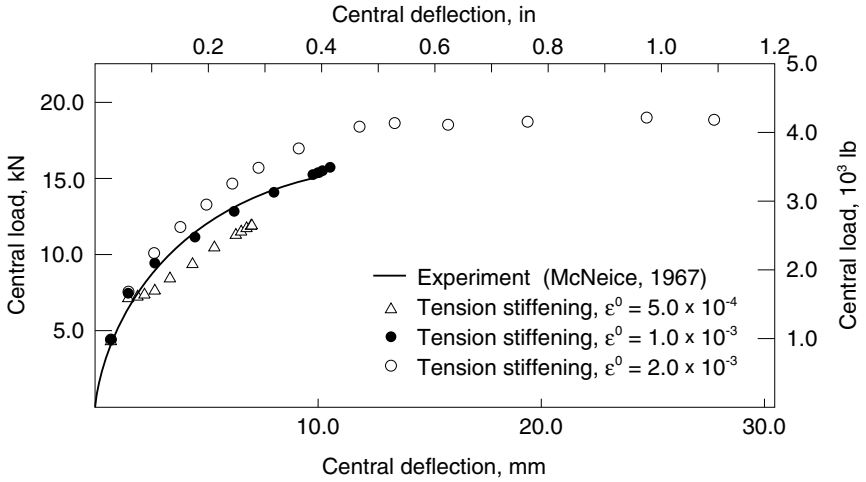


Figure 1.1.5–2 Load-deflection response of McNeice slab, Abaqus/Standard.

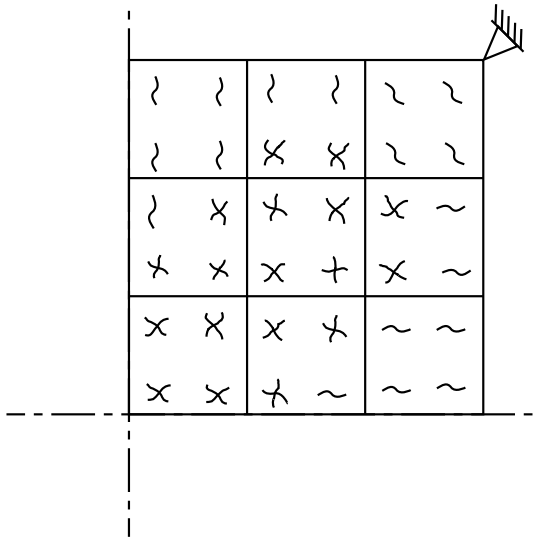


Figure 1.1.5–3 Crack pattern on lower surface of slab, Abaqus/Standard.

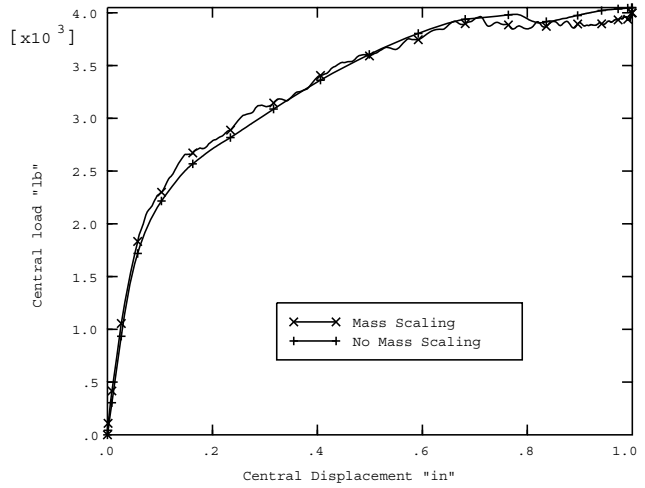


Figure 1.1.5–4 Load-deflection response of McNeice slab, Abaqus/Explicit; influence of mass scaling.

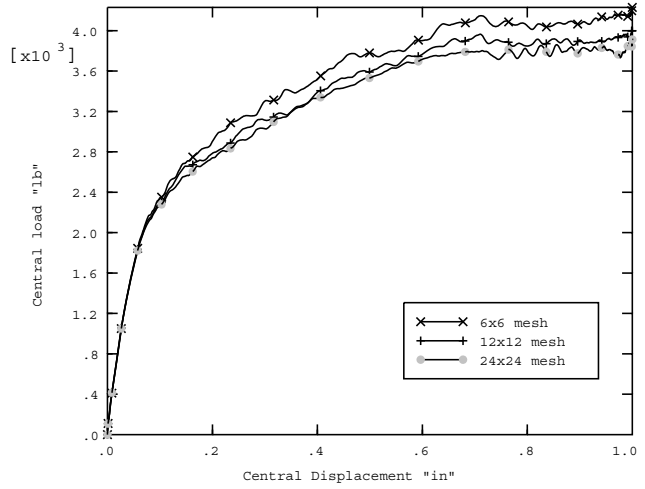


Figure 1.1.5–5 Load-deflection response of McNeice slab, Abaqus/Explicit; influence of mesh refinement.

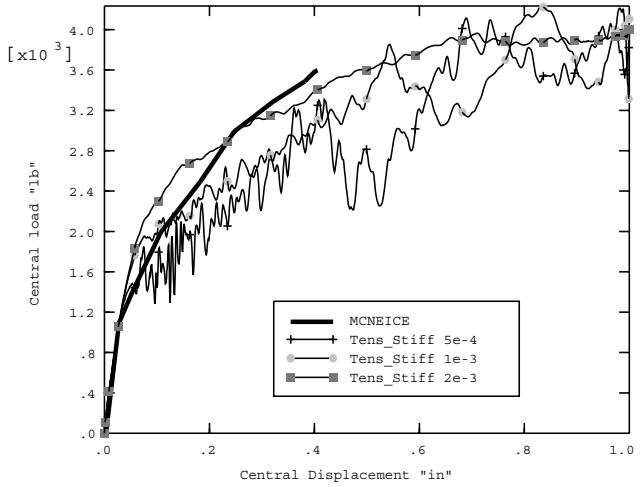


Figure 1.1.5–6 Load-deflection response of McNeice slab, Abaqus/Explicit; influence of tension stiffening.

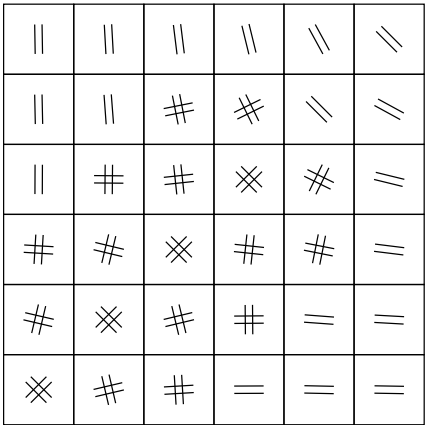


Figure 1.1.5–7 Crack pattern on lower surface of slab, Abaqus/Explicit.

1.1.6 JOINTED ROCK SLOPE STABILITY

Product: Abaqus/Standard

This example illustrates the use of the jointed material model in the context of geotechnical applications. We examine the stability of the excavation of part of a jointed rock mass, leaving a sloped embankment. This problem is chosen mainly as a verification case because it has been studied previously by Barton (1971) and Hoek (1970), who used limit equilibrium methods, and by Zienkiewicz and Pande (1977), who used a finite element model. This example also has been extended to study the slope stability of excavated soil medium with the same geometry, by using the Mohr-Coulomb plasticity model with and without the tension cutoff feature.

Geometry and model

The plane strain model analyzed is shown in Figure 1.1.6–1 together with the excavation geometry and material properties. The rock mass contains two sets of planes of weakness: one vertical set of joints and one set of inclined joints. We begin from a nonzero state of stress. In this problem this consists of a vertical stress that increases linearly with depth to equilibrate the weight of the rock and horizontal stresses caused by tectonic effects: such stress is quite commonly encountered in geotechnical engineering. The active "loading" consists of removal of material to represent the excavation. It is clear that, with a different initial stress state, the response of the system would be different. This illustrates the need of nonlinear analysis in geotechnical applications—the response of a system to external "loading" depends on the state of the system when that loading sequence begins (and, by extension, to the sequence of loading). We can no longer think of superposing load cases, as is done in a linear analysis.

Practical geotechnical excavations involve a sequence of steps, in each of which some part of the material mass is removed. Liners or retaining walls can be inserted during this process. Thus, geotechnical problems require generality in creating and using a finite element model: the model itself, and not just its response, changes with time—parts of the original model disappear, while other components that were not originally present are added. This example is somewhat academic, in that we do not encounter this level of complexity. Instead, following the previous authors' use of the example, we assume that the entire excavation occurs simultaneously.

Solution controls

The jointed material model includes a joint opening/closing capability. When a joint opens, the material is assumed to have no elastic stiffness with respect to direct strain across the joint system. Because of this, and also as a result of the fact that different combinations of joints may be yielding at any one time, the overall convergence of the solution is expected to be nonmonotonic. In such cases setting the time incrementation parameters automatically is generally recommended to prevent premature termination of the equilibrium iteration process because the solution may appear to be diverging.

As the end of the excavation process is approached, the automatic incrementation algorithm reduces the load increment significantly, indicating the onset of failure of the slope. In such analyses it is useful to specify a minimum time step to avoid unproductive iteration.

For the nonassociated flow case the unsymmetric equation solver should be used. This is essential for obtaining an acceptable rate of convergence since nonassociated flow plasticity has a nonsymmetric stiffness matrix.

Results and discussion

In this problem we examine the effect of joint cohesion on slope collapse through a sequence of solutions with different values of joint cohesion, with all other parameters kept fixed. Figure 1.1.6–2 shows the variation of horizontal displacements as cohesion is reduced at the crest of the slope (point ***A*** in Figure 1.1.6–1) and at a point one-third of the way up the slope (point ***B*** in Figure 1.1.6–1). This plot suggests that the slope collapses if the cohesion is less than 24 kPa for the case of associated flow or less than 26 kPa for the case of nonassociated flow. These compare well with the value calculated by Barton (26 kPa) using a planar failure assumption in his limit equilibrium calculations. Barton's calculations also include "tension cracking" (akin to joint opening with no tension strength) as we do in our calculation. Hoek calculates a cohesion value of 24 kPa for collapse of the slope. Although he also makes the planar failure assumption, he does not include tension cracking. This is, presumably, the reason why his calculated value is lower than Barton's. Zienkiewicz and Pande assume the joints have a tension strength of one-tenth of the cohesion and calculate the cohesion value necessary for collapse as 23 kPa for associated flow and 25 kPa for nondilatant flow.

Figure 1.1.6–3 shows the deformed configuration after excavation for the nonassociated flow case and clearly illustrates the manner in which the collapse is expected to occur. Figure 1.1.6–4 shows the magnitude of the frictional slip on each joint system for the nonassociated flow case. A few joints open near the crest of the slope.

The study of soil slope stability using the Mohr-Coulomb plasticity model is performed for two cases: one without tension cutoff and one including the tension cutoff feature. The tension cutoff feature limits the stress carrying capacity of soil in tension. It can be seen that the maximum principal stress without tension cutoff (see contour plot in Figure 1.1.6–5) is higher than the limiting maximum principal stress (see Figure 1.1.6–6) with tension cutoff as expected. With tension cutoff, one also observes the appearance of the equivalent plastic strain in tension, PEEQT in the region of maximum principal stress. In this case it is also seen that the equivalent plastic strain, PEEQ on the cohesion failure surface is higher compared to the case without tension cutoff. The contour plots for the equivalent plastic strains are not shown.

Input files

jointrockstabil_nonassoc_30pka.inp	Nonassociated flow case problem; cohesion = 30 kPa.
jointrockstabil_assoc_25kpa.inp	Associated flow case; cohesion = 25 kPa.
mc_slopestabil.inp	Slope stability analysis, Mohr-Coulomb plasticity without tension cutoff

mctc_slopestabil.inp	Slope stability analysis, Mohr-Coulomb plasticity with tension cutoff

References

- Barton, N., "Progressive Failure of Excavated Rock Slopes," *Stability of Rock Slopes*, Proceedings of the 13th Symposium on Rock Mechanics, Illinois, pp. 139–170, 1971.
- Hoek, E., "Estimating the Stability of Excavated Slopes in Open Cast Mines," Trans. Inst. Min. and Metal., vol. 79, pp. 109–132, 1970.
- 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.

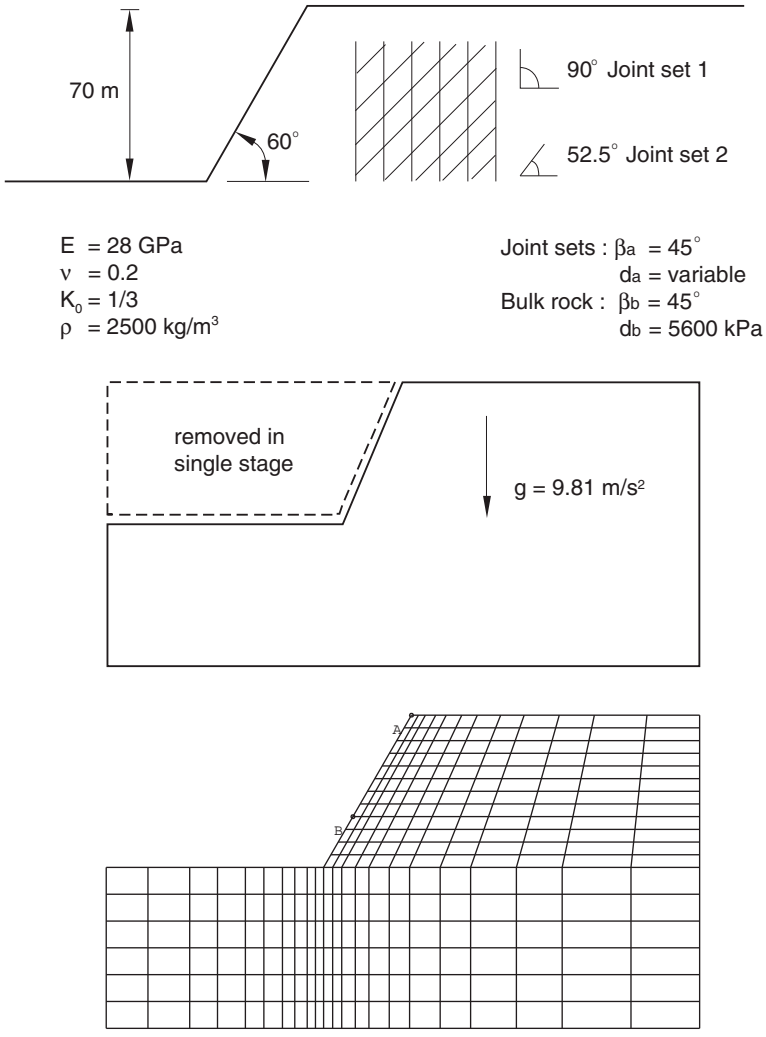


Figure 1.1.6–1 Jointed rock slope problem.

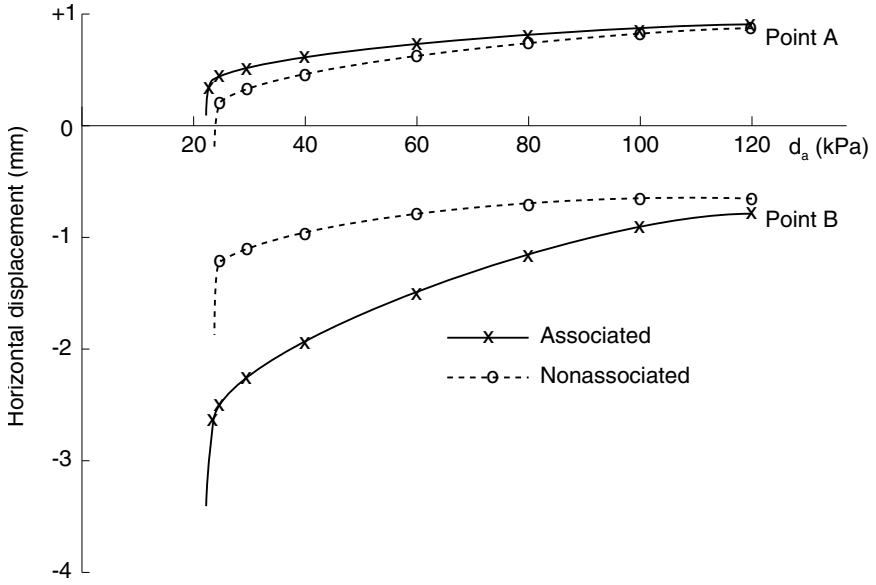


Figure 1.1.6–2 Horizontal displacements with varying cohesion.

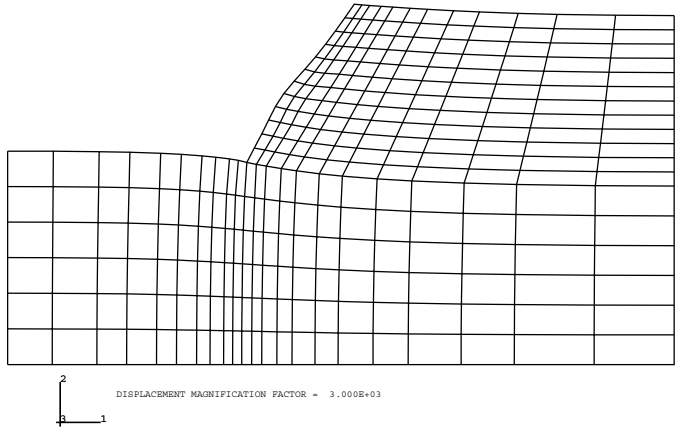
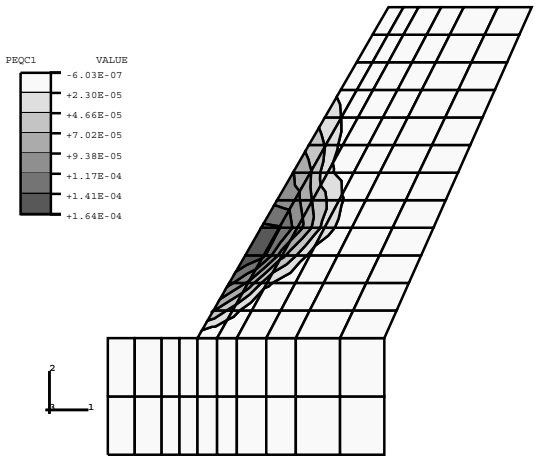
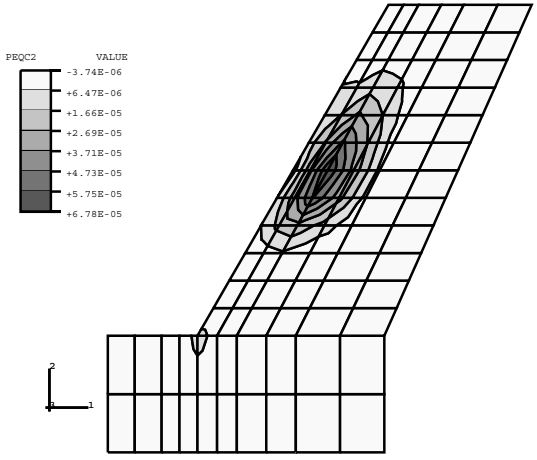


Figure 1.1.6–3 Deformed configuration (nonassociated flow).



Joint set 1 (vertical joints).



Joint set 2 (inclined joints).

Figure 1.1.6–4 Contours of frictional slip magnitudes (nonassociated flow).

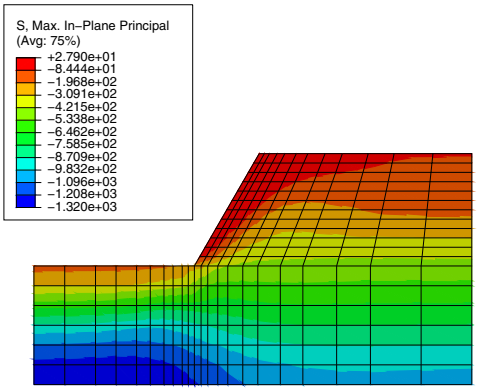


Figure 1.1.6–5 Maximum principal stress without tension cutoff.

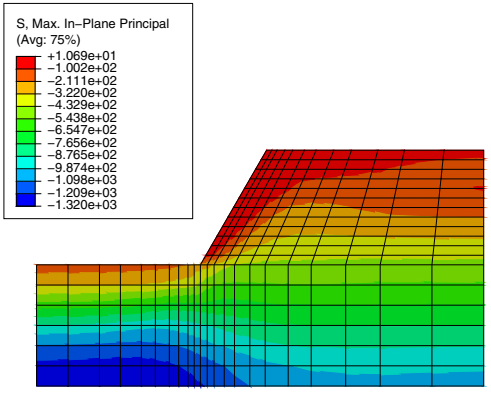


Figure 1.1.6–6 Maximum principal stress with tension cutoff.

1.1.7 NOTCHED BEAM UNDER CYCLIC LOADING

Product: Abaqus/Standard

This example illustrates the use of the nonlinear isotropic/kinematic hardening material model to simulate the response of a notched beam under cyclic loading. The model has two features to simulate plastic hardening in cyclic loading conditions: the center of the yield surface moves in stress space (kinematic hardening behavior), and the size of the yield surface evolves with inelastic deformation (isotropic hardening behavior). This combination of kinematic and isotropic hardening components is introduced to model the Bauschinger effect and other phenomena such as plastic shakedown, ratchetting, and relaxation of the mean stress.

The component investigated in this example is a notched beam subjected to a cyclic 4-point bending load. The results are compared with the finite element results published by Benallal et al. (1988) and Doghri (1993). No experimental data are available.

Geometry and model

The geometry and mesh are shown in Figure 1.1.7–1. Figure 1.1.7–2 shows the discretization in the vicinity of the notch, which is the region of interest in this analysis. Only one-half of the beam is modeled since the geometry and loading are symmetric with respect to the $x = 0$ plane. All dimensions are given in millimeters. The beam is 1 mm thick and is modeled with plane strain, second-order, reduced-integration elements (type CPE8R). The mesh is chosen to be similar to the mesh used by Doghri (1993). No mesh convergence studies have been performed.

Material

The material properties reported by Doghri (1993) for a low-carbon (AISI 1010), rolled steel are used in this example.

A Young's modulus of $E = 210$ GPa and a Poisson's ratio of $\nu = 0.3$ define the elastic response of the material. The initial yield stress is $\sigma|_0 = 200$ MPa.

The nonlinear evolution of the center of the yield surface is defined by the equation

$$\dot{\boldsymbol{\alpha}} = C\frac{1}{\sigma^0}(\boldsymbol{\sigma} - \boldsymbol{\alpha})\dot{\bar{\varepsilon}}^{pl} - \gamma\,\boldsymbol{\alpha}\,\dot{\bar{\varepsilon}}^{pl},$$

where $\boldsymbol{\alpha}$ is the backstress, σ^0 is the size of the yield surface (size of the elastic range), $\bar{\varepsilon}^{pl}$ is the equivalent plastic strain, and $C = 25.5$ GPa and $\gamma = 81$ are the material parameters that define the initial hardening modulus and the rate at which the hardening modulus decreases with increasing plastic strain, respectively. The quantity $\sqrt{2/3}\,C/\gamma = 257$ MPa defines the limiting value of the equivalent backstress $\bar{\alpha} = \sqrt{\boldsymbol{\alpha}^{dev} : \boldsymbol{\alpha}^{dev}}$; further hardening is possible only through the change in the size of the yield surface (isotropic hardening).

The isotropic hardening behavior of this material is modeled with the exponential law

$$\sigma^0 = \sigma|_0 + Q_\infty(1 - e^{-b\,\bar{\varepsilon}^{pl}}),$$

where σ^0 is the size of the yield surface (size of the elastic range), Q_∞ = 2000 MPa is the maximum increase in the elastic range, and $b = 0.26$ defines the rate at which the maximum size is reached as plastic straining develops.

The material used for this simulation is cold rolled. This work hardened state is represented by specifying an initial equivalent plastic strain $\bar{\varepsilon}^{pl}|_0 = 0.43$ (so that $\sigma^0 = 411$ MPa) and an initial backstress tensor

$$\boldsymbol{\alpha}|_0 = \begin{bmatrix} 128 & 0 & 0 \\ 0 & -181 & 0 \\ 0 & 0 & 53 \end{bmatrix} \text{MPa}.$$

Loading and boundary conditions

The beam is subjected to a 4-point bending load. Since only half of the beam is modeled, the model contains one concentrated load at a distance of 26 mm from the symmetry plane (see Figure 1.1.7–1). The pivot point is 42 mm from the symmetry plane. The simulation runs 3 1/2 cycles over 7 time units. In each cycle the load is ramped from zero to 675 N and back to zero. An amplitude curve is used to describe the loading and unloading. The increment size is restricted to a maximum of 0.125 to force Abaqus to follow the prescribed loading/unloading pattern closely.

Results and discussion

Figure 1.1.7–3 shows the final deformed shape of the beam after the 3 1/2 cycles of load; the final load on the beam is 675 N.

The deformation is most severe near the root of the notch. The results reported in Figure 1.1.7–4 and Figure 1.1.7–5 are measured in this area (element 166, integration point 3). Figure 1.1.7–4 shows the time evolution of stress versus strain. Several important effects are predicted using this material model. First, the onset of yield occurs at a lower absolute stress level during the first unloading than during the first loading, which is the Bauschinger effect. Second, the stress-strain cycles tend to shift and stabilize so that the mean stress decreases from cycle to cycle, tending toward zero. This behavior is referred to as the relaxation of the mean stress and is most pronounced in uniaxial cyclic tests in which the strain is prescribed between unsymmetric strain values. Third, the yield surface shifts along the strain axis with cycling, whereas the shape of the stress-strain curve tends to remain similar from one cycle to the next. This behavior is known as ratchetting and is most pronounced in uniaxial cyclic tests in which the stress is prescribed between unsymmetric stress values. Finally, the hardening behavior during the first half-cycle is very flat relative to the hardening curves of the other cycles, which is typical of work hardened metals whose initial hardened state is a result of a large monotonic plastic deformation caused by a forming process such as rolling. The low hardening modulus is the result of the initial conditions on backstress, which places the center of the yield surface at a distance of $\sqrt{\boldsymbol{\alpha}|_0^{dev} : \boldsymbol{\alpha}|_0^{dev}} = 228$ MPa away from the origin of stress space. Since this distance is close to the maximum possible distance (257 MPa), most of the hardening during the first cycle is isotropic.

These phenomena are modeled in this example primarily by the nonlinear evolution of the backstress, since the rate of isotropic hardening is very small. This behavior can be verified by conducting an analysis in which the elastic domain remains fixed throughout the analysis.

Figure 1.1.7–5 shows the evolution of the direct components of the deviatoric part of the backstress tensor. The backstress components evolve most during the first cycle as the Bauschinger effect overcomes the initial hardening configuration. Only the deviatoric components of the backstress are shown so that the results obtained using Abaqus can be compared to those reported by Doghri (1993). Since Abaqus uses an extension of the Ziegler evolution law, a backstress tensor with nonzero pressure is produced, whereas the backstress tensor produced with the law used by Doghri (which is an extension of the linear Prager law) is deviatoric. Since the plasticity model considers only the deviatoric part of the backstress, this difference in law does not affect the other solution variables.

The results shown in Figure 1.1.7–4 and Figure 1.1.7–5 agree well with the results reported by Doghri (1993).

Input files

cyclicnotchedbeam.inp	Input data.
cyclicnotchedbeam_mesh.inp	Element and node data.

References

- Benallal, A., R. Billardon, and I. Doghri, "An Integration Algorithm and the Corresponding Consistent Tangent Operator for Fully Coupled Elastoplastic and Damage Equations," Communications in Applied Numerical Methods, vol. 4, pp. 731–740, 1988.
- Doghri, I., "Fully Implicit Integration and Consistent Tangent Modulus in Elasto-Plasticity," International Journal for Numerical Methods in Engineering, vol. 36, pp. 3915–3932, 1993.

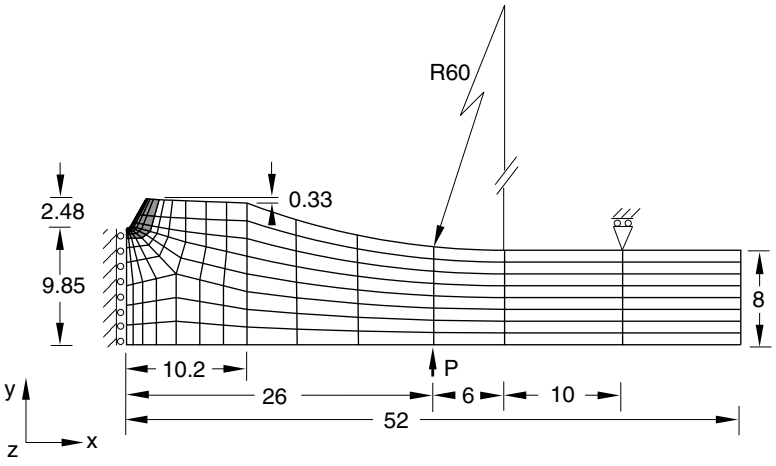


Figure 1.1.7–1 Undeformed mesh (dimensions in mm).

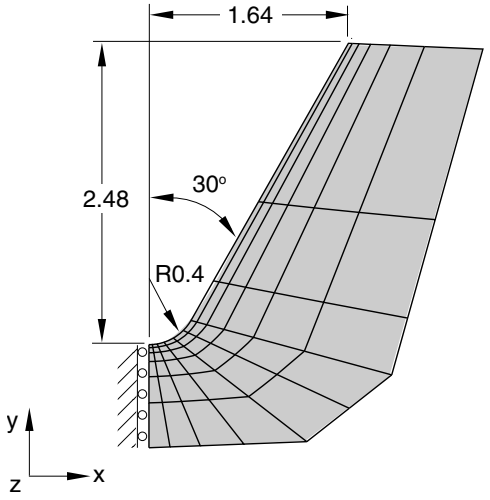


Figure 1.1.7–2 Magnified view of the root of the notch.

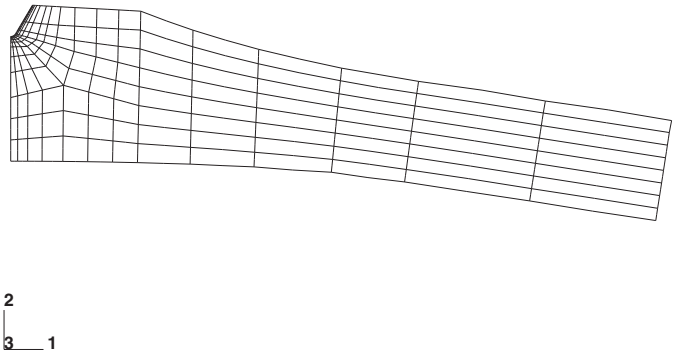


Figure 1.1.7–3 Deformed mesh at the conclusion of the simulation. Displacement magnification factor is 3.

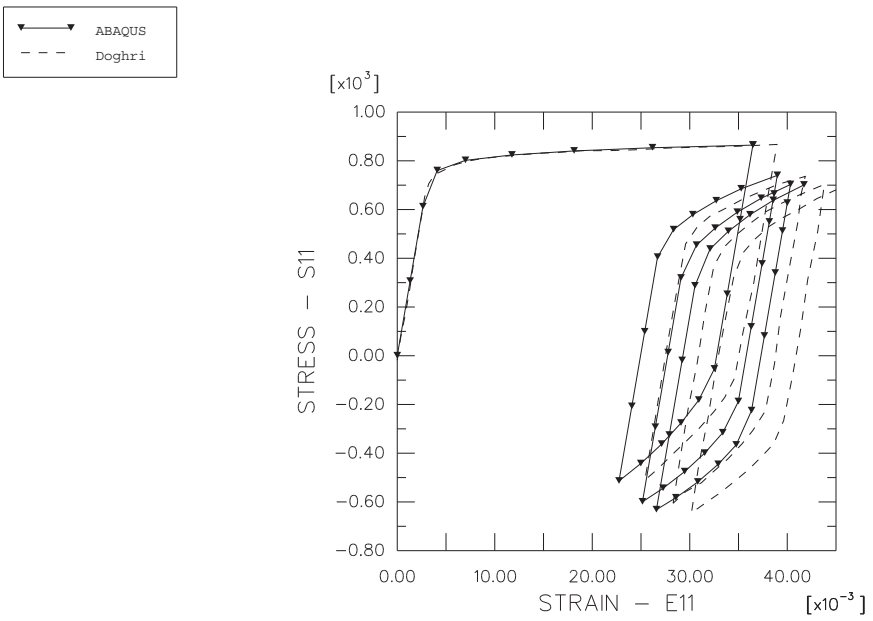


Figure 1.1.7–4 Evolution of stress versus strain in the vicinity of the root of the notch.

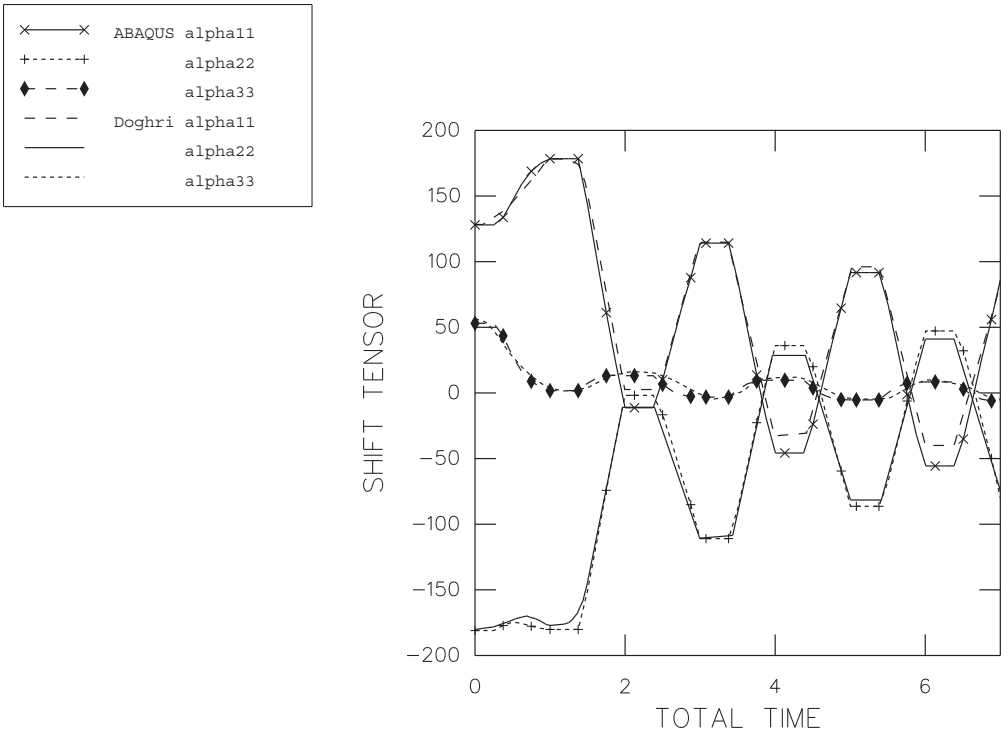


Figure 1.1.7–5 Evolution of the diagonal components of the deviatoric part of the backstress tensor.

1.1.8 UNIAXIAL RATCHETTING UNDER TENSION AND COMPRESSION

Product: Abaqus/Standard

Objectives

This example demonstrates the following Abaqus features and techniques:

- using the nonlinear isotropic/kinematic hardening model to predict deformation in a specimen subjected to monotonic and cyclic loading; and
- modeling the effect of ratchetting (accumulation of plastic strain under a cyclic load).

Application description

Preventing ratchetting is very important in the design of components subject to cyclic loading in the inelastic domain. The amount of plastic strain can accumulate continuously with an increasing number of cycles and may eventually cause material failure. Therefore, many cyclic plastic models have been developed with the goal of modeling ratchetting correctly. In this example we show that the combined isotropic/kinematic hardening model available in Abaqus can predict ratchetting and that the results obtained using this model correlate very well with experimental results.

This example considers two loading conditions: monotonic deformation and uniaxial cyclic tension and compression.

Geometry

The specimen studied is shown in Figure 1.1.8–1. All dimensions are specified in the figure. For the experiments (Portier et al., 2000) the specimens were obtained from a tube with an outer diameter of 130 mm and a wall thickness of 28 mm. The specimens were heat treated to ensure the initial isotropy of the material.

Materials

The specimen is made of austenitic type 316 stainless steel. The material mechanical properties are listed in Table 1.1.8–1. A detailed description of the calibration of parameters is given in “Material parameters determination” below.

Boundary conditions and loading

The specimen is constrained at the bottom surface in the longitudinal direction, and a load is applied to the top surface.

Abaqus modeling approaches and simulation techniques

In this example deformations of a specimen subject to monotonic and cyclic loads are studied. In both cases static analyses are performed. Taking advantage of the axial symmetry of the specimen, axisymmetric elements are used.

Summary of analysis cases

Case 1 Static analysis of a specimen subject to a monotonic load.

Case 2 Static analysis of a specimen subject to an unsymmetric cyclic load.

Case 1 Monotonic load

The experimental monotonic load data are used to calibrate the kinematic hardening model. The purpose of this case is to verify that the simulation results agree with the experimental results and to compare the accuracy of the results obtained using a model with one backstress and a model with two backstresses.

Analysis types

A static stress analysis is performed.

Mesh design

The specimen is meshed with CAX4R and CAX3 elements. The mesh is shown in Figure 1.1.8–2.

Material model

The combined isotropic/kinematic hardening model is used to model the response of the material. This material model requires that the elastic parameters (Young's modulus and Poisson's ratio), the initial yield stress, the isotropic hardening parameters, and the kinematic hardening parameters are specified.

Material parameters determination

The elastic parameters, the initial yield stress, and the isotropic hardening parameters are assumed to be equal to those reported in Portier et al. (2000) for the Ohno and Wang model. The kinematic hardening component is defined by specifying half-cycle test data, where the data are obtained by digitizing the results reported by Portier et al. The values of all the parameters, including the kinematic hardening parameters obtained from the test data, are presented in Table 1.1.8–1.

Boundary conditions

The specimen is fixed in the longitudinal direction at the bottom surface.

Loading

A displacement of 0.45 mm is applied to the top surface.

Results and discussion

The simulation and experimental results are presented graphically in Figure 1.1.8–3. The strains and stresses are computed by averaging the values in the elements lying at the center of the specimen. The experimental curve shows three distinct regions: a linear elastic region, an elastic-plastic transition zone, and an almost linear response region at large strain values. The model with two backstresses captures this response very well. One of the backstresses has a large value of the parameter γ, which captures the shape of the transition zone correctly, while the second backstress with a relatively small value of γ captures the nearly linear response at large strains correctly. The parameter γ in the model with one backstress has a relatively large value, which results in large discrepancies between the experimental and predicted responses at large strains.

Case 2 Uniaxial tension and compression cyclic analysis

The objective of this case is to show that the combined isotropic/kinematic hardening model can be used to predict the response of a material subject to a cyclic load accurately and, in particular, to predict the ratchetting effect. In addition, the results obtained using a model with one backstress are compared to those obtained using a model with two backstresses.

Analysis types

A static stress analysis is performed.

Mesh design

The mesh is the same as in Case 1.

Material model

The material model is the same as in Case 1.

Boundary conditions

The specimen is fixed in the longitudinal direction at the bottom surface.

Loading

A cyclic load of $\sigma_{mean} = 11.0$ MPa and $\Delta\sigma = 15.4$ MPa is applied to the top surface of the specimen. This load produces an approximate cyclic load of $\sigma_{mean} = 100$ MPa and $\Delta\sigma = 140$ MPa at the center part of the specimen.

Results and discussion

The simulation results obtained for the model with one backstress and the model with two backstresses, together with the experimental results, are depicted in Figure 1.1.8–4. The strains were computed by averaging the strains in the elements lying at the center of the specimen. The figure shows that both simulation models are capable of predicting ratchetting. It also shows that the results obtained using the model with two backstresses correlate better with the experimental results.

Discussion of results and comparison of cases

The results of the analyses show that the combined isotropic/kinematic hardening model can be used to predict the ratchetting effect accurately. In addition, a substantial improvement in the agreement between simulation and experimental results can be achieved by using a model with multiple backstresses instead of a model with a single backstress. The former model predicts more accurately the shape of the stress-strain curve in the monotonic loading case and the ratchetting strain in a cyclic loading case. In this example increasing the number of backstresses from one to two produced a substantial improvement in the results. However, further increasing the number of backstresses does not significantly improve the results.

Files

Case 1 Monotonic

ratch_axi_monotonic_1.inp	Input file to analyze a specimen subjected to a monotonic load using the model with one backstress.
ratch_axi_monotonic_2.inp	Input file to analyze a specimen subjected to a monotonic load using the model with two backstresses.

Case 2 Cyclic

ratch_axi_unsymcyclic_1.inp	Input file to analyze a specimen subjected to a cyclic load using the model with one backstress.
ratch_axi_unsymcyclic_2.inp	Input file to analyze a specimen subjected to a cyclic load using the model with two backstresses.

References

Abaqus Analysis User's Guide

- "Models for metals subjected to cyclic loading," Section 23.2.2 of the Abaqus Analysis User's Guide

Abaqus Keywords Reference Guide

- *CYCLIC HARDENING
- *PLASTIC

Other

- Portier, L., S. Calloch, D. Marquis, and P. Geyer, "Ratchetting Under Tension-Torsion Loadings: Experiments and Modelling," International Journal of Plasticity, vol. 16, pp. 303–335, 2000.

Table 1.1.8–1 Mechanical properties for 316 steel.

Material properties:	
Young's modulus	192.0 GPa
Poisson's ratio	0.3
Initial yield stress	120.0 MPa
Isotropic hardening parameters:	
Q_∞	120.0 MPa
b	13.2
Kinematic hardening parameters:	
Model with one backstress	
C	218.5 GPa
γ	1956.6
Model with two backstresses	
C_1	2.067 GPa
γ_1	44.7
C_2	246.2 GPa
γ_2	2551.4

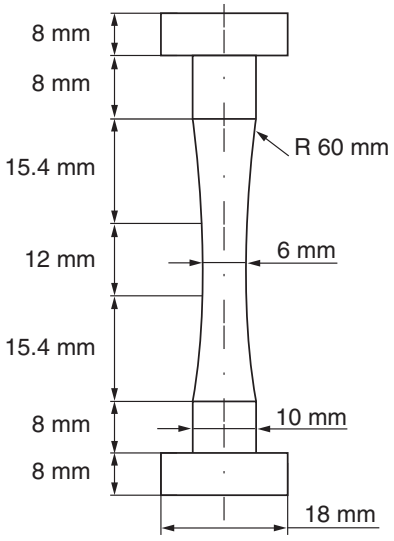


Figure 1.1.8–1 Geometry and size of the specimen.

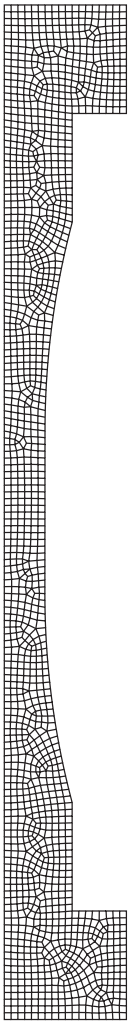


Figure 1.1.8–2 Finite element mesh of the specimen.

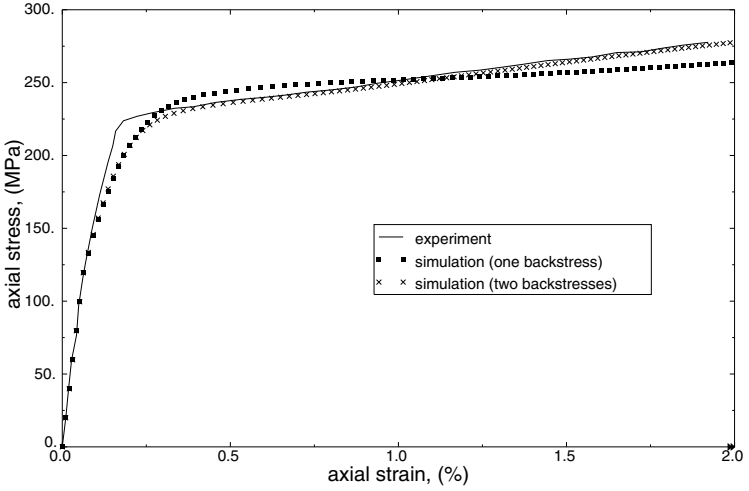


Figure 1.1.8–3 Stress-strain curves for monotonic tensile loading.

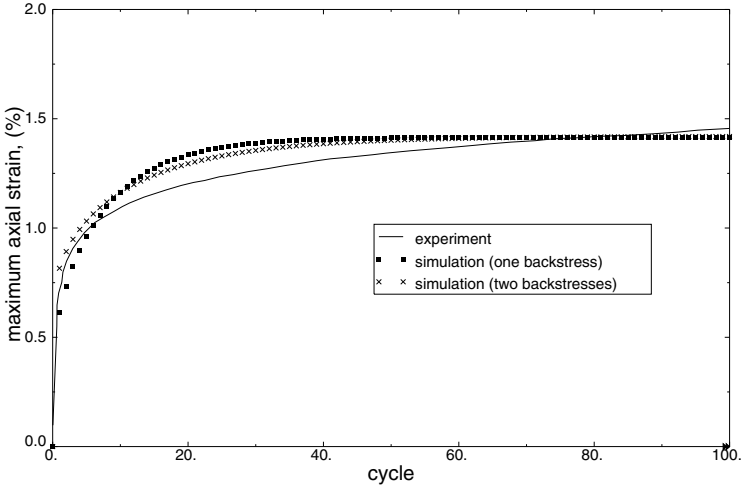


Figure 1.1.8–4 Maximum axial strain versus number of cycles.

1.1.9 HYDROSTATIC FLUID ELEMENTS: MODELING AN AIRSPRING

Products: Abaqus/Standard Abaqus/Explicit

Airsprings are rubber or fabric actuators that support and contain a column of compressed air. They are used as pneumatic actuators and vibration isolators. Unlike conventional pneumatic cylinders, airsprings have no pistons, rods, or dynamic seals. This makes them better suited to handle off-center loading and shock. In addition, airsprings are considerably more flexible than other types of isolators: the airspring's inflation pressure can be changed to compensate for different loads or heights without compromising isolation efficiency. Dils (1992) provides a brief discussion of various practical uses of airsprings.

In this section two examples of the analysis of a cord-reinforced rubber airspring are discussed. Static analyses are performed in Abaqus/Standard, and quasi-static analyses are performed in Abaqus/Explicit. The first example is a three-dimensional, half-symmetry model that uses finite-strain shell elements to model the rubber spring and rebar to model the multi-ply steel reinforcements in the rubber membrane. In addition, a three-dimensional, element-based rigid surface is used to define the contact between the airspring and the lateral metal bead. The cord-reinforced rubber membrane is modeled using a hyperelastic material model with steel rebar.

The second example is a two-dimensional, axisymmetric version of the first model that uses composite axisymmetric, finite-strain shell elements to model the cord-reinforced rubber spring and an axisymmetric, element-based rigid surface in the contact definition. This model uses a composite shell section consisting of a thin orthotropic elastic layer sandwiched between two hyperelastic layers. The orthotropic layer captures the mechanical properties of the rebar definition used in the three-dimensional model.

The orthotropic material constants have been obtained by performing simple tests on a typical element of the three-dimensional model. The three-dimensional shell model uses rebar with material properties that are initially identical to the properties of the composite shell section in the axisymmetric shell model.

For comparison, Abaqus/Standard input files that use finite-strain membrane elements instead of finite-strain shell elements to model the cord-reinforced rubber spring are also included for both the axisymmetric and three-dimensional models.

In all analyses the airspring cavity is modeled using the surface-based fluid cavity capability (see "Surface-based fluid cavities: overview," Section 11.5.1 of the Abaqus Analysis User's Guide) and air inside the cavity is modeled as a compressible or "pneumatic" fluid satisfying the ideal gas law.

Geometry and model

The dimensions of the airspring have been inferred from the paper by Fursdon (1990). This airspring, shown in Figure 1.1.9–1, is fairly large and is used in secondary suspension systems on railway bogies. However, the shape of the airspring is typical of airsprings used in other applications. The airspring's cross-section is shown in Figure 1.1.9–2. The airspring is toroidal in shape, with an inner radius of 200 mm and an outer radius of 400 mm. The airspring has been idealized in the model as consisting of two circular, metal disks connected to each other via a rubber component. The lower disk has a radius of 200 mm, and the upper disk has a radius of 362.11 mm. The disks are initially coaxial and are 100 mm

apart. The rubber component is doubly curved and toroidal in shape. The rubber is constrained in the radial direction by a circular bead 55 mm in radius that goes around the circumference of the upper disk.

The rubber "hose" in the half-symmetry, three-dimensional model is modeled with 550 S4R finite-strain shell elements. The mesh in the upper hemisphere of the hose is more refined than that in the lower hemisphere, because the rubber membrane undergoes a reversal in curvature in the upper region as it contours the circular bead attached to the upper disk. The circular bead is modeled using an axisymmetric, discrete rigid surface. Contact with the rubber is enforced by defining a contact pair between this rigid surface and a surface defined on the (deformable) shell mesh in the contacting region. The metal disks are assumed to be rigid relative to the rubber component of the airspring. The lower metal disk is modeled using boundary conditions, while the upper disk is modeled as part of the rigid surface. The meshes of the rubber membrane and the rigid surface are shown in Figure 1.1.9–3.

The fluid cavity is modeled using the surface-based fluid cavity capability (see "Surface-based fluid cavities: overview," Section 11.5.1 of the Abaqus Analysis User's Guide). To define the cavity completely and to ensure proper calculation of its volume, surface elements are defined in the three-dimensional models along the bottom and top rigid disk boundaries of the cavity, even though no displacement elements exist along those surfaces. Since Abaqus does not provide two-dimensional surface elements, the rigid disk boundaries are modeled with structural elements instead of surface elements in the axisymmetric models. The cavity reference node 50000 has a single degree of freedom representing the pressure inside the cavity. Because of symmetry only half of the cavity boundary has been modeled. The cavity reference node has been placed on the model's symmetry plane, $y = 0$, to assure proper calculation of the cavity volume. Figure 1.1.9–4 shows the mesh of the airspring's cavity.

To facilitate comparisons, the two-dimensional axisymmetric model uses the same cross-sectional mesh refinement as the 180° model. The rubber component is modeled with 25 SAX1 shell elements. The circular bead is modeled with an element-based rigid surface constructed of RAX2 rigid elements. Contact with the hose is enforced by defining a contact pair between this rigid surface and a surface defined on the (deformable) shell mesh in the contacting region. Once again, the lower rigid metal disk is modeled by boundary conditions, and the upper rigid metal disk is modeled as part of the rigid body. The mesh of the rubber membrane and the contact master surface is shown in Figure 1.1.9–5, and the mesh of the cavity is shown in Figure 1.1.9–6. For the membrane model the SAX1 elements are replaced with either MAX1 elements or MGAX1 elements.

Symmetry boundary conditions and initial shell curvature

Symmetry has been exploited in the three-dimensional airspring model, and the plane $y = 0$ has been made a plane of symmetry. Since S4R shell elements are true curved shell elements, accurate definition of the initial curvature of the surface being modeled is required, especially on the plane of symmetry. If the user does not provide this information by specifying the normal to the surface at the shell nodes, Abaqus will estimate the normal direction based on the coordinates of the surrounding nodes on the shell. Normals computed in this fashion will be inaccurate on the symmetry plane: they will have out-of-plane components, which will lead to convergence difficulties in Abaqus/Standard and inaccurate results. To avoid these difficulties, direction cosines have been specified for all shell nodes in the model.

Material properties

The walls of an airspring's rubber component are made from plies of symmetrically placed, positively and negatively oriented reinforcement cords. The walls of an actual component are made of several such layers. However, for the purposes of the three-dimensional example problem being considered, the airspring's wall is taken to be a rubber matrix with a single 6-mm-thick symmetric layer of positively and negatively oriented cords. The cords are modeled by uniformly spaced skew rebar in the shell elements. The rebar are assumed to be made of steel. The rubber is modeled as an incompressible Mooney-Rivlin (hyperelastic) material with $C_{10} = 3.2$ MPa and $C_{01} = 0.8$ MPa, and the steel is modeled as a linear elastic material with $E = 210.0$ GPa and $\nu = 0.3$.

Skew rebar orientations in shell elements are defined by giving the angle between the local 1-axis and the rebar. The default local 1-direction is the projection of the global x-axis onto the shell surface (see "Conventions," Section 1.2.2 of the Abaqus Analysis User's Guide). It is for this reason, and to make the rebar definition uniform for all elements, that the axis of revolution of the airspring model has been chosen to be the global x-axis. Two rebar layers, **PLSBAR** and **MNSBAR**, have been defined with orientation angles of 18° and −18°, respectively. The cross-sectional area of the rebar is 1 mm^2, and they are spaced every 3.5 mm in the shell surface.

The above rebar specification is simplified and somewhat unrealistic. The reinforced plies used in the manufacture of the airspring are located in an initially cylindrical tube with uniform rebar angles. However, the transformation of these layers from a cylindrical geometry to a toroidal one gives the airspring a variable rebar angle and rebar spacing that is dependent on the radius from the axis of revolution of the torus and on the initial rebar angle (see Fursdon, 1990). Hence, a more realistic simulation would require different rebar definitions in each ring of elements in the airspring model.

In the axisymmetric shell model the airspring walls are modeled by a three-layer composite shell section. The two outer layers are each 2.5 mm thick and made up of the same Mooney-Rivlin material that is used in the 180° model. The middle "rebar" layer is 1 mm thick and is made up of an orthotropic elastic material that captures the mechanical behavior of the positively and negatively oriented rebar definition used in the three-dimensional airspring model.

The plane stress orthotropic engineering constants are obtained by looking at the response of a typical element in the three-dimensional model (element 14) subjected to uniaxial extensions along the local 1- and 2-directions. Using a shell thickness of 1 mm, the in-plane states of stress and strain resulting from these two tests are

Test	ε_1	ε_2	σ_1 **(MPa)**	σ_2 **(MPa)**
1-direction	1.00×10^{-2}	-8.75×10^{-2}	2.48×10^{1}	-2.41×10^{-5}
2-direction	-1.05×10^{-3}	1.00×10^{-2}	-5.96×10^{-6}	2.86×10^{-1}

For a plane-stress orthotropic material the in-plane stress and strain components are related to each other as follows:

$$\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \end{pmatrix} = \begin{pmatrix} 1/E_1 & -\nu_{12}/E_1 \\ -\nu_{21}/E_2 & 1/E_2 \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \end{pmatrix},$$

where E_1, E_2, ν_{12}, and ν_{21} are engineering constants. Solving for these constants using the above stress-strain relation and the results of the two uniaxial tests yields

$$E_1 = 2.48 \times 10^3 \,\text{MPa}$$
$$E_2 = 2.86 \times 10^1 \,\text{MPa}$$

$$\nu_{12} = 9.1$$
$$\nu_{21} = 0.1$$

The remaining required engineering constants—G_{12}, G_{13}, and G_{23}—play no role in the rebar layer definition. Consequently, they have been arbitrarily set to be equal to the shear modulus of the rubber, which is given by $2(C_{10} + C_{01})$.

For the axisymmetric membrane model the bulk material is chosen to have the same material properties (Mooney-Rivlin hyperelastic) as those used in the 180° model and the axisymmetric shell model. The rebar parameters and material properties are chosen such that they capture the initial material properties of the sandwiched steel layer in the axisymmetric shell model. The principal material directions do not rotate in the axisymmetric shell model (they are the default element basis directions—the meridional and the hoop directions, respectively). However, they do rotate with finite strain in the axisymmetric membrane model as a result of the use of rebar. Initial stresses are applied to the rebar in the axisymmetric membrane model.

In all analyses the air inside the airspring cavity has been modeled as an ideal gas with molecular weight of 0.044 kg and molar heat capacity of 30 J/kg °K.

Loading

In the Abaqus/Standard model the airspring is first pressurized to 506.6×10^3 kPa (5 atms) while holding the upper disk fixed. This pressure is applied by prescribing boundary degree of freedom 8 at the cavity reference node. In this case the air volume is adjusted automatically to fill the cavity.

In the next step the prescribed boundary condition on the pressure degree of freedom is removed, thus sealing the cavity with the current air volume. In addition, during this step the boundary condition on the vertical displacement degree of freedom of the rigid body reference node is removed, and in its place a downward load of 150 kN is applied.

The next step is a static linear perturbation procedure. In the axisymmetric model two load cases are considered: one tests the axial stiffness of the airspring with the cavity pressure allowed to vary (closed cavity conditions) and the other tests its axial stiffness with the cavity pressure fixed. The linear perturbation step in the three-dimensional analysis contains three load cases, all under variable cavity pressure (closed cavity) conditions: the first tests the axial stiffness of the airspring, the second tests its lateral stiffness, and the third tests its rotational stiffness for rocking motion in the symmetry plane.

The axisymmetric analysis concludes with a general step in which the airspring is compressed by increasing the downward load to 240.0 kN. The three-dimensional analysis concludes with a general step in which the airspring is subjected to a lateral displacement of 20 mm.

The loading for the Abaqus/Explicit model is similar to that for the Abaqus/Standard model, except for the linear perturbation steps. The airspring is first pressurized to 506.6×10^3 kPa (5 atms) while holding the upper disk fixed. In the next step the boundary condition on the pressure degree of freedom is removed, thereby sealing the cavity with the current air volume. In addition, for both the axisymmetric and three-dimensional models, the boundary condition on the vertical displacement degree of freedom of the rigid body reference node is modified, so that a downward displacement is applied. This is in contrast to the Abaqus/Standard axisymmetric analysis, where a force was applied. Since the airspring is pressurized, a sudden change in applied force would cause a sudden change in acceleration and induce a low frequency transient response. As a result, the simulation time required for the transient effects to diminish would be very long. Hence, we apply a displacement instead of a force. In the axisymmetric analysis the downward displacement is chosen to be 75 mm, so that the increase in pressure is close to the increase seen in Step 4 of the Abaqus/Standard axisymmetric analysis. The downward displacement in the three-dimensional Abaqus/Explicit analysis is set to 20 mm so that the results from this step may be compared with those from Step 4 of the Abaqus/Standard three-dimensional analysis.

Results and discussion

Figure 1.1.9–7 and Figure 1.1.9–8 show displaced shape plots of the axisymmetric shell model at the end of the pressurization step. It is of interest to compare the results from this model with those from the 180° model to validate the material model that was used for the rebar reinforcements in the axisymmetric model. A close look at the nodal displacements reveals that the deformation is practically identical for the axisymmetric and three-dimensional models. The results are also virtually identical between corresponding Abaqus/Standard and Abaqus/Explicit models. Moreover, the axial reaction force at the rigid body reference node is 156 kN for the axisymmetric model and 155 kN for the 180° model (after multiplication by a factor of 2). The cavity volume predicted by the axisymmetric model is 8.22×10^{-2} m^3 versus 8.34×10^{-2} m^3 for the 180° model (again, after multiplication by a factor of 2).

Linearized stiffnesses for the airspring are obtained from the Abaqus/Standard linear perturbation load cases. The stiffness is computed by dividing the relevant reaction force at the rigid body reference node by the appropriate displacement. For the axisymmetric model the airspring's axial stiffness under variable cavity pressure conditions is 826 kN/m; its axial stiffness under fixed cavity pressure conditions is 134 kN/m. The difference in axial stiffness between these two cases (a factor of 6) is the result of differences in cavity pressure experienced during axial compression. Under variable cavity pressure conditions, a fixed mass of fluid (air) is contained in a cavity whose volume is decreasing; thus, the cavity pressure increases. Under fixed cavity pressure conditions, the pressure is prescribed as a constant value for the load case. For the 180° model the predicted stiffnesses under variable cavity pressure are as follows: the axial stiffness is 821 kN/m, the lateral stiffness is 3.31 MN/m, and the rotational stiffness is 273 kN/m.

Figure 1.1.9–9 shows a series of displaced shape plots associated with the compression of the axisymmetric Abaqus/Standard airspring model during Step 4. For comparison, Figure 1.1.9–10 shows a series of displaced shape plots associated with the compression of the axisymmetric Abaqus/Explicit model during Step 2. Figure 1.1.9–11 shows the corresponding load-deflection curves. Although the displacement of the rigid body in the Abaqus/Explicit analysis was applied over a short time period (which caused significant inertial effects in the model), there is still good agreement between the

slope of the load-displacement curves from the two analyses. The response of the airspring is only slightly nonlinear; consequently, there is good agreement between the axial stiffness obtained with the linear perturbation load case and that obtained from the slope of the load-displacement curve. Figure 1.1.9–12 shows a plot of cavity pressure versus the downward displacement of the rigid body in Step 4 of the Abaqus/Standard analysis and Step 2 of the Abaqus/Explicit analysis. The gauge pressure in the cavity increases by approximately 50% during this step. This pressure increase substantially affects the deformation of the airspring structure and cannot be specified as an externally applied load during the step since it is an unknown quantity. Figure 1.1.9–13 shows a plot of cavity volume versus the downward displacement of the rigid body in Step 4 of the Abaqus/Standard analysis and Step 2 of the Abaqus/Explicit analysis. The cavity pressure and the cavity volume results from the static Abaqus/Standard analysis and the quasi-static Abaqus/Explicit analysis are virtually identical. The corresponding results from the axisymmetric membrane model (not shown) are also in good agreement with the above results.

Figure 1.1.9–14 shows the displaced shape of the 180° Abaqus/Standard model at the end of Step 4, in which a lateral displacement was applied to the airspring. Figure 1.1.9–15 shows the corresponding displaced shape of the 180° model at the end of Step 2 of the Abaqus/Explicit analysis. Figure 1.1.9–16 shows a plot of the load-displacement curves obtained from these steps. Although there is a certain amount of noise that results from the contact conditions and the coarseness of the mesh, the load-deflection curve shows good agreement between the analyses performed quasi-statically in Abaqus/Explicit and statically in Abaqus/Standard. The Abaqus/Explicit analysis was run in double precision to eliminate some of the noise in the load-displacement curve.

Input files

hydrofluidairspring_s4r.inp	Three-dimensional Abaqus/Standard model using shell elements.
hydrofluidairspring_s4r_surf.inp	Three-dimensional Abaqus/Standard model using shell elements with surface-to-surface contact.
hydrofluidairspring_sax1.inp	Axisymmetric Abaqus/Standard model using shell elements.
hydrofluidairspring_sax1_surf.inp	Axisymmetric Abaqus/Standard model using shell elements with surface-to-surface contact.
airspring_exp_s4r_surfcav.inp	Three-dimensional Abaqus/Explicit model using shell elements.
airspring_exp_sax1_surfcav.inp	Axisymmetric Abaqus/Explicit model using shell elements.
hydrofluidairspring_m3d4.inp	Three-dimensional Abaqus/Standard model using membrane elements.
hydrofluidairspring_m3d4_surf.inp	Three-dimensional Abaqus/Standard model using membrane elements with surface-to-surface contact.
hydrofluidairspring_max1.inp	Axisymmetric Abaqus/Standard analysis using rebar reinforced membrane elements.

hydrofluidairspring_mgax1.inp	Axisymmetric Abaqus/Standard analysis using rebar reinforced membrane elements with twist.
airspring_s4r_gcont_surfcav.inp	Three-dimensional Abaqus/Explicit analysis using shell elements and general contact.

References

- Dils, M., “Air Springs vs. Air Cylinders,” Machine Design, May 7, 1992.
- Fursdon, P. M. T., “Modelling a Cord Reinforced Component with ABAQUS,” 6th UK ABAQUS User Group Conference Proceedings, 1990.

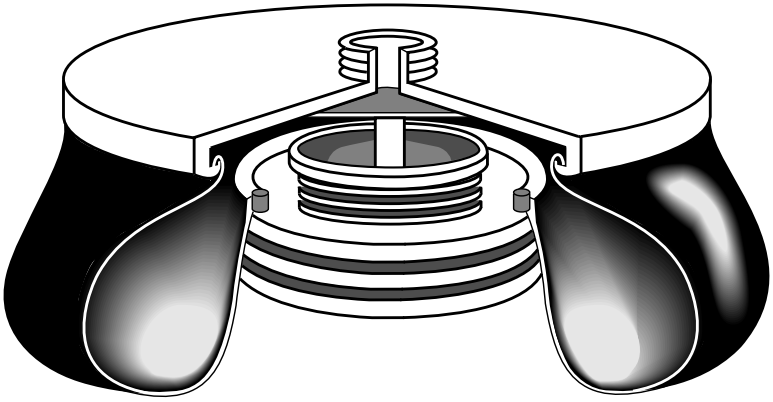


Figure 1.1.9–1 A cord-reinforced airspring.

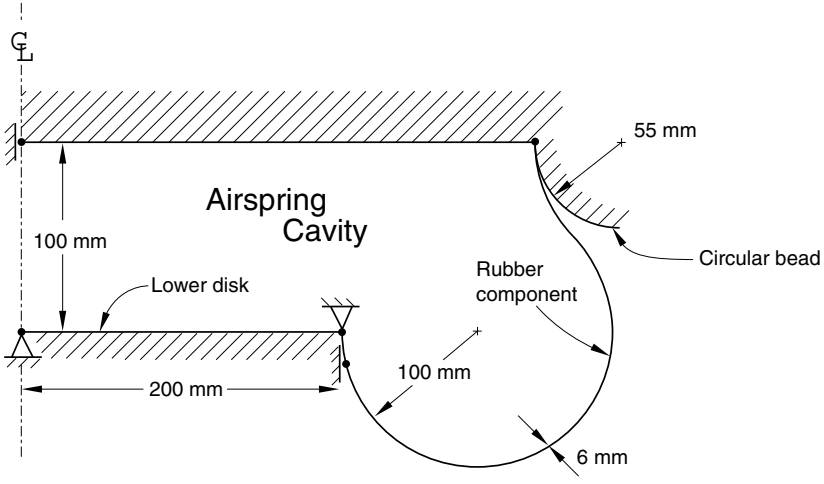


Figure 1.1.9–2 The airspring model cross-section.

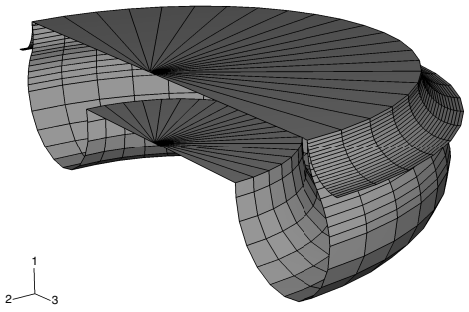


Figure 1.1.9–3 180° model: mesh of the rubber membrane and the contact master surface.

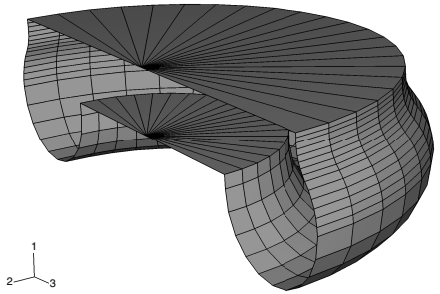


Figure 1.1.9–4 180° model: mesh of the airspring cavity.

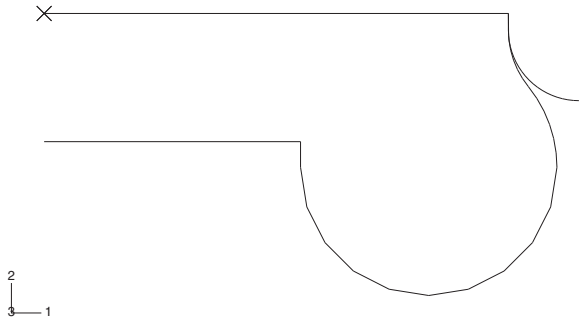


Figure 1.1.9–5 Axisymmetric model: mesh of the rubber membrane and the contact master surface.

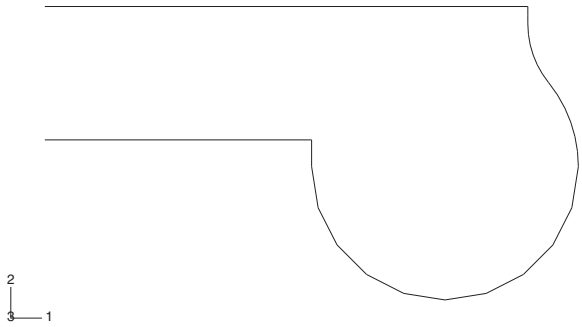


Figure 1.1.9–6 Axisymmetric model: mesh of the airspring cavity.

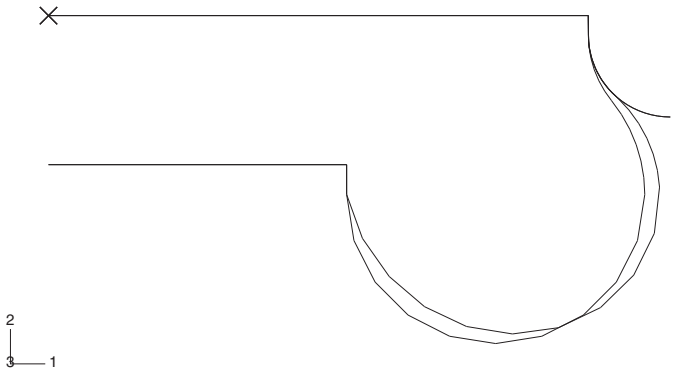


Figure 1.1.9–7 Axisymmetric Abaqus/Standard model: deformed configuration at the end of Step 1.

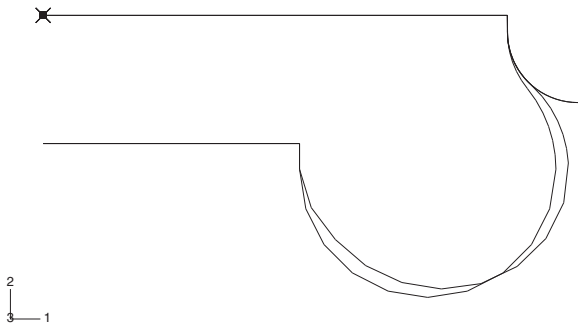


Figure 1.1.9–8 Axisymmetric Abaqus/Explicit model: deformed configuration at the end of Step 1.

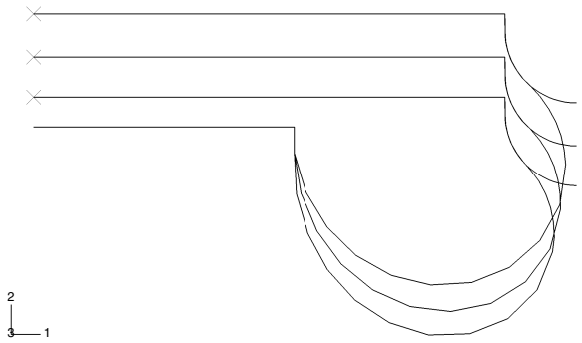


Figure 1.1.9–9 Axisymmetric Abaqus/Standard model: progressive deformed configurations during Step 4.

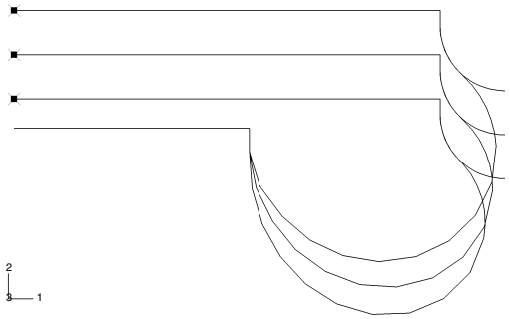


Figure 1.1.9–10 Axisymmetric Abaqus/Explicit model: progressive deformed configurations during Step 2.

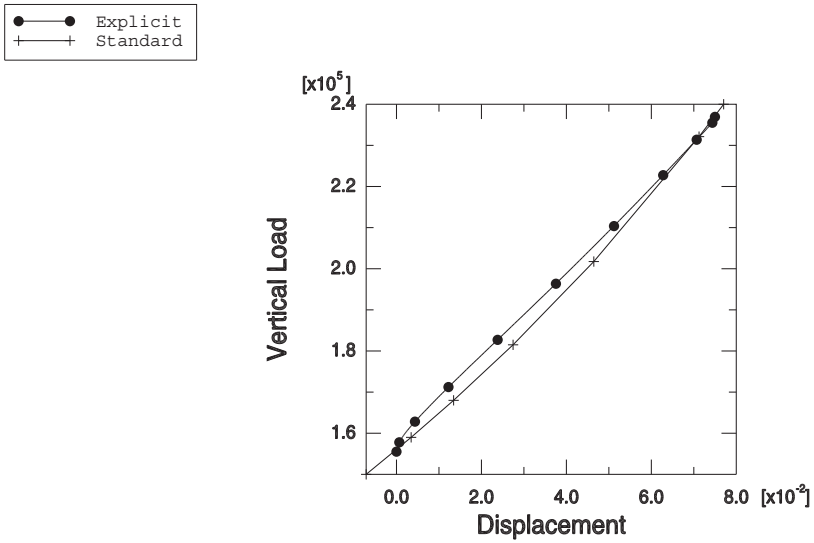


Figure 1.1.9–11 Load-displacement curves (axisymmetric models).

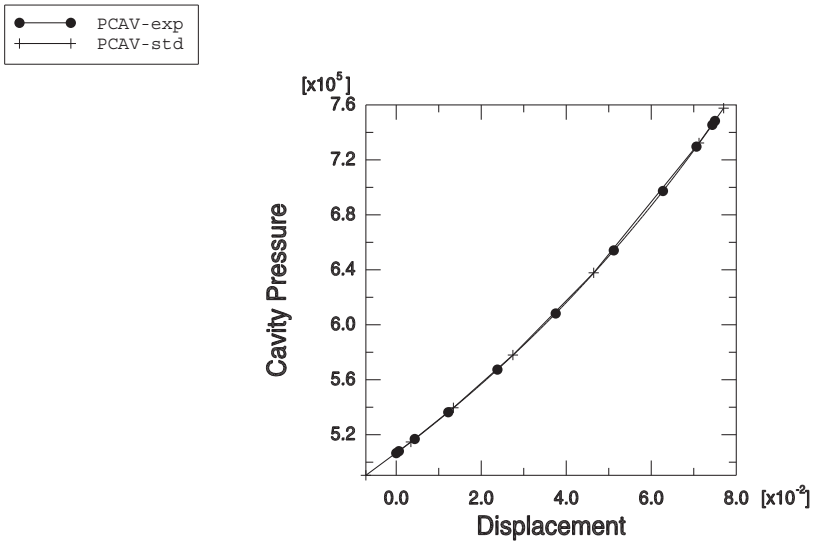


Figure 1.1.9–12 Cavity pressure versus downward displacement (axisymmetric models).

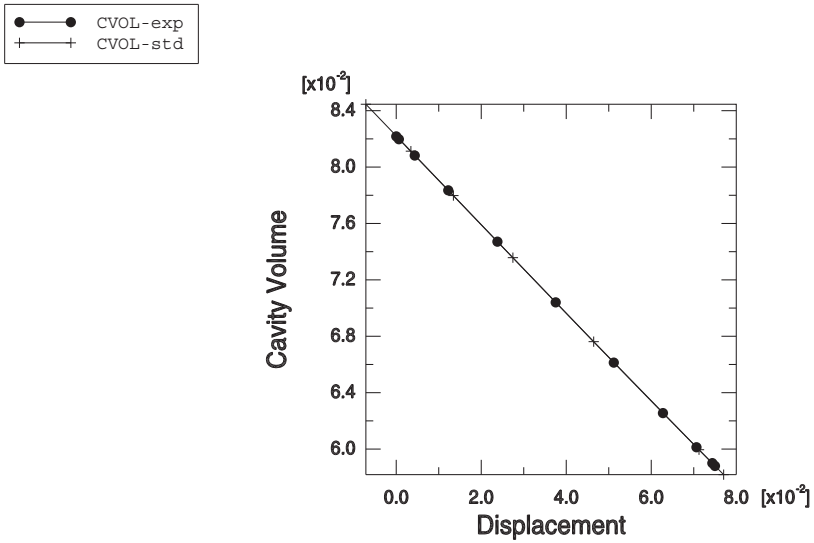


Figure 1.1.9–13 Cavity volume versus downward displacement (axisymmetric models).

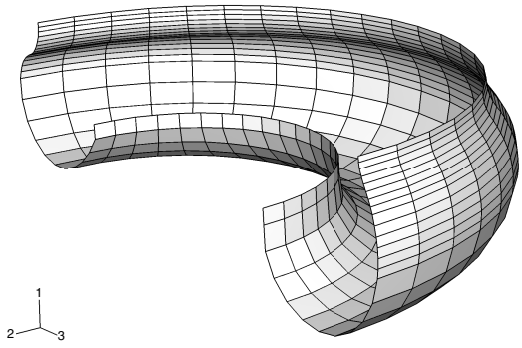


Figure 1.1.9–14 180° Abaqus/Standard model: deformed configuration at the end of Step 4.

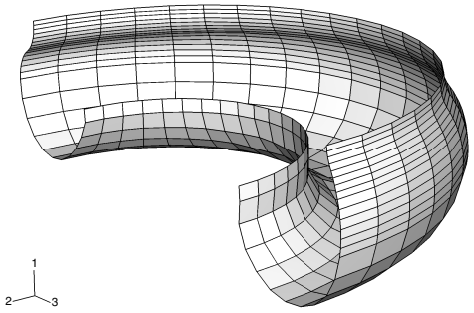


Figure 1.1.9–15 180° Abaqus/Explicit model: deformed configuration at the end of Step 2.

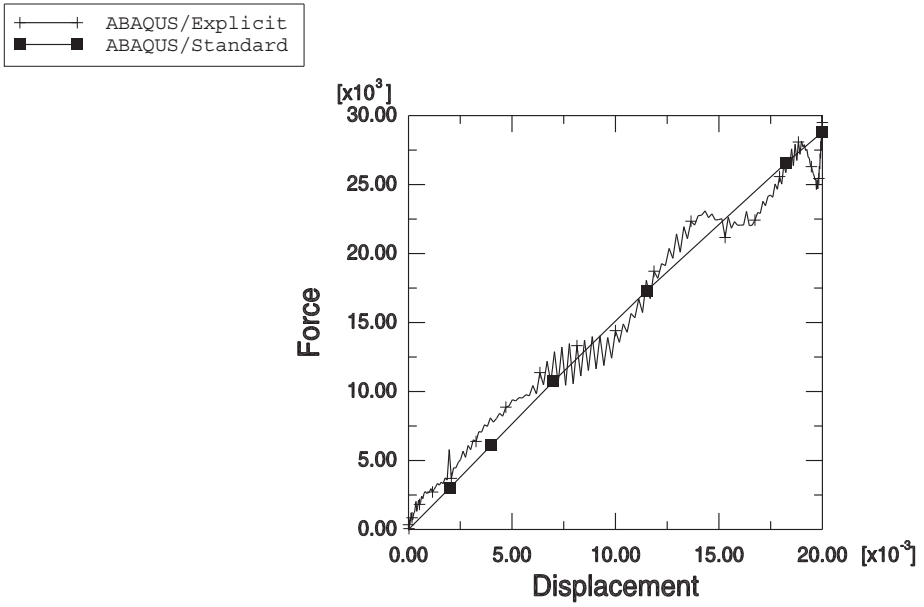


Figure 1.1.9–16 Load-displacement curves for the 180° analyses.

1.1.10 SHELL-TO-SOLID SUBMODELING AND SHELL-TO-SOLID COUPLING OF A PIPE JOINT

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

Submodeling is the technique used in Abaqus for analyzing a local part of a model with a refined mesh, based on interpolation of the solution from an initial global model (usually with a coarser mesh) onto the nodes on the appropriate parts of the boundary of the submodel. Shell-to-solid submodeling models a region with solid elements, when the global model is made up of shell elements. This example uses the scaling parameter in the submodel boundary condition to scale the values of prescribed boundary conditions for driven variables without requiring the global model to be rerun.

Shell-to-solid coupling is a feature in Abaqus by which three-dimensional shell meshes can be coupled automatically to three-dimensional solid meshes. Unlike shell-to-solid submodeling, which first performs a global analysis on a shell model followed by a submodel analysis with a continuum model, the shell-to-solid coupling model uses a single analysis, with solid and shell elements used in different regions.

Both shell-to-solid submodeling and shell-to-solid coupling provide cost-effective approaches to model enhancement. The purpose of this example is to demonstrate both capabilities in Abaqus. The analysis is tested as a static process in Abaqus/Standard and as a dynamic process in both Abaqus/Standard and Abaqus/Explicit. To demonstrate the shell-to-solid submodeling capability, the problem is solved quasi-statically in Abaqus/Explicit. The overall displacements are small, and to avoid noise-induced dynamic effects, the quasi-static Abaqus/Explicit analysis is run in double precision.

In addition, an Abaqus Scripting Interface script is included that creates a shell global model using Abaqus/CAE. The script then uses data from the output database created by the analysis of the global model to drive a solid submodel. The script ends by displaying an overlay plot of the global model and the submodel in the Visualization module.

Geometry and model

In this problem the joint between a pipe and a plate is analyzed. A pipe of radius 10 mm and thickness 0.75 mm is attached to a plate that is 10 mm long, 5 mm wide, and 1 mm thick. The pipe-plate intersection has a fillet radius of 1 mm. Taking advantage of the symmetry of the problem, only half the assembly is modeled. Both the pipe and the plate are assumed to be made of aluminum with $E = 69 \times 10^3$ MPa, $\nu = 0.3$, and $\rho = 2740$ kg/m^3.

The global model for the submodeling analysis is meshed with S4R elements as shown in Figure 1.1.10–1. The fillet radius is not taken into consideration in the shell model. The static submodel is meshed using three-dimensional C3D20R continuum elements (see Figure 1.1.10–2). A coarser mesh using C3D8R elements is chosen for the dynamic tests. The shell-to-solid coupling model is meshed with S4R shell elements and C3D20R continuum elements as shown in Figure 1.1.10–3. The continuum meshes used in the static submodeling and shell-to-solid coupling analyses are identical. The continuum meshes extend 10 mm along the pipe length, have a radius of 25 mm in the plane of the plate, and use four layers through the thickness. The continuum meshes accurately model the fillet radius at the joint. Hence, it is possible to calculate the stress concentration in the fillet. The problem

could be expanded by adding a ring of welded material to simulate a welded joint (for this case the submodel would have to be meshed with new element layers representing the welded material at the joint). The example could also be expanded by including plastic material behavior in the submodel while using an elastic global model solution.

A reference static solution consisting entirely of C3D20R continuum elements is also included (see Figure 1.1.10–4). The mesh of the reference solution in the vicinity of the joint is very similar to that used in the submodeling and shell-to-solid coupling analyses.

The geometry and material properties of the Abaqus/Explicit shell-to-solid coupling model are identical to the Abaqus/Standard models. The Abaqus/Explicit shell model is meshed with S4R elements, and the continuum model is meshed with C3D10M elements.

Loading

The pipe is subjected to a concentrated load acting in the x-direction applied at the free end, representing a shear load on the pipe. An edge-based surface is defined at the free edge of the pipe. This surface is coupled to a reference node that is defined at the center of the pipe using a distributing coupling constraint. The concentrated load is applied to the reference point. For the submodeling approach the load magnitude is 10 N for the global analysis and 10 N and 20 N, respectively, in the two steps of the submodel analysis (a scale factor of 2.0 is applied to the displacements of the driven nodes in the second step of the submodel analysis). For the shell-to-solid coupling approach the load magnitude is 10 N and 20 N, respectively, in the two steps, as in the submodel analysis. For the dynamic cases the load is applied gradually over the entire step time by using a smooth-step amplitude curve.

Kinematic boundary conditions

The plate is clamped along all edges. In the solid submodel, kinematic conditions are interpolated from the global model at two surfaces of the submodel: one lying within the pipe and the other within the plate. The default center zone size, equal to 10% of the maximum shell thickness, is used. Thus, only one layer of driven nodes lies within the center zone, and only these nodes have all three displacement components driven by the global solution. For the remaining driven nodes only the displacement components parallel to the global model midsurface are driven from the global model. Thus, a single row of nodes transmits the transverse shear forces from the shell solution to the solid model.

Results and discussion

The loading and boundary conditions are such that the pipe is subjected to bending. The end of the pipe that is attached to the plate leads to deformation of the plate itself (see Figure 1.1.10–5 and Figure 1.1.10–6). From a design viewpoint the area of interest is the pipe-plate joint where the pipe is bending the plate. Hence, this area is modeled with continuum elements to gain a better understanding of the deformation and stress state.

Figure 1.1.10–7 shows the contours of the out-of-plane displacement component in the plate for both the static submodel and the shell-to-solid coupling analyses. The submodel is in good agreement with the displacement of the global shell model around the joint. The out-of-plane displacement for the

shell-to-solid coupling is slightly less than that for the submodel analysis but is in good agreement with the reference solution shown in Figure 1.1.10–8.

The stress concentration in the fillet radius is obtained for the solid models. The maximum Mises stresses at the integration points and nodes for the reference solution, submodel, and shell-to-solid coupling analyses are shown in Table 1.1.10–1. As illustrated in Table 1.1.10–1 and Figure 1.1.10–9, the Mises stress computed in the shell-to-solid coupling analysis agrees very well with the reference solution. The continuity of displacements and the minimal distortion of the stress field at the shell-to-solid interface indicates that the shell-to-solid coupling has been modeled accurately. The difference in the maximum Mises stress between the submodel analysis and the shell-to-solid coupling solution (see Figure 1.1.10–10) can be partially attributed to the fact that the global shell model is more flexible than the shell-to-solid coupling model. The x-displacement at the distributing coupling reference node for the global shell model due to the 10 N load is .605 mm compared to .513 mm for the shell-to-solid coupling analysis and .512 mm for the reference solution. Thus, the static submodel mesh is subjected to slightly higher deformation. If the global shell analysis is run with an x-displacement boundary condition of .513 mm on the reference node instead of a concentrated load of 10 N, the subsequent maximum nodal Mises stress in the submodel analysis drops to 80.2 MPa, which is in better agreement with the reference solution. Figure 1.1.10–11 and Figure 1.1.10–12 show, respectively, the comparison of the out-of-plane displacement component in the continuum-mesh plate and the comparison of the Mises stress for the submodel with scaled boundary condition and the shell-to-solid coupling model with scaled load. The submodel results are in good agreement with the shell-to-solid coupling results.

The relatively large difference between the maximum Mises stresses at the integration points and the nodes in the region of the fillet (as illustrated in Table 1.1.10–1) indicates that the mesh in the fillet region is probably too coarse and should be refined. No such refinement was performed in this example.

Overall, the Abaqus/Explicit shell-to-solid coupling analysis is in good agreement with the Abaqus/Standard shell-to-solid coupling results. The continuity of displacements and the minimal distortion of the stress field at the shell-to-solid interface indicate that the shell-to-solid coupling has been modeled accurately. The out-of-plane displacements of the plate predicted by Abaqus/Explicit are very close to the Abaqus/Standard values. The maximum nodal Mises stress in the fillet region is 56 MPa, and the x-displacement at the coupling constraint reference node for the global shell model due to the 10 N load is .423 mm. The Abaqus/Explicit analysis is solved quasi-statically by assigning a nominal density of 500 kg/m^3 to the pipe-plate material and ramping up the load over 12,000 increments. Closer approximation to the static limit, achieved by reducing the density of the pipe-plate material to 50 kg/m^3, results in a maximum Mises stress of 87 MPa in the fillet region, which is very close to the Abaqus/Standard result.

The results for the submodel dynamic cases agree well with the global results. Both Abaqus/Standard and Abaqus/Explicit submodels read the results of the same global Abaqus/Explicit analysis. Good agreement is also found between the Abaqus/Explicit and Abaqus/Standard submodel analyses.

Input files

Static and quasi-static input files

pipe_submodel_s4r_global.inp	S4R global model.
pipe_submodel_s4r_global_n.inp	Node definitions for the S4R global model.
pipe_submodel_s4r_global_e.inp	Element definitions for the S4R global model.
pipe_submodel_c3d20r_sub_s4r.inp	C3D20R submodel that uses the S4R global model. The scaling parameter is used in the second step.
pipe_submodel_c3d20r_sub_s4r_n.inp	Node definitions for the C3D20R submodel that uses the S4R global model.
pipe_submodel_c3d20r_sub_s4r_e.inp	Element definitions for the C3D20R submodel that uses the S4R global model.
pipe_submodel_s4_global.inp	S4 global model.
pipe_submodel_s4_global_n.inp	Node definitions for the S4 global model.
pipe_submodel_s4_global_e.inp	Element definitions for the S4 global model.
pipe_submodel_c3d20r_sub_s4.inp	C3D20R submodel that uses the S4 global model.
pipe_submodel_c3d20r_sub_s4_n.inp	Node definitions for the C3D20R submodel that uses the S4 global model.
pipe_submodel_c3d20r_sub_s4_e.inp	Element definitions for the C3D20R submodel that uses the S4 global model.
pipe_cae_c3d20rsub_s4.py	Python script that creates an S4 global model and a C3D20R submodel using Abaqus/CAE.
pipe_shell2solid_c3d20r_s4r.inp	Shell-to-solid coupling model with C3D20R and S4R elements. The load is scaled in the second step.
pipe_shell2solid_c3d20r_s4r_n1.inp	Node definitions for the shell-to-solid coupling model with C3D20R and S4R elements.
pipe_shell2solid_c3d20r_s4r_n2.inp	Node definitions for the shell-to-solid coupling model with C3D20R and S4R elements.
pipe_shell2solid_c3d20r_s4r_n3.inp	Node definitions for the shell-to-solid coupling model with C3D20R and S4R elements.
pipe_shell2solid_c3d20r_s4r_e1.inp	Element definitions for the shell-to-solid coupling model with C3D20R and S4R elements.
pipe_shell2solid_c3d20r_s4r_e2.inp	Element definitions for the shell-to-solid coupling model with C3D20R and S4R elements.
pipe_shell2solid_c3d20r_s4r_e3.inp	Element definitions for the shell-to-solid coupling model with C3D20R and S4R elements.
pipe_shell2solid_c3d10_s4r.inp	Shell-to-solid coupling model with C3D10 and S4R elements.
pipe_shell2solid_c3d10_s4r_n1.inp	Node definitions for the shell-to-solid coupling model with C3D10 and S4R elements.

pipe_shell2solid_c3d10_s4r_n2.inp	Node definitions for the shell-to-solid coupling model with C3D10 and S4R elements.
pipe_shell2solid_c3d10_s4r_n3.inp	Node definitions for the shell-to-solid coupling model with C3D10 and S4R elements.
pipe_shell2solid_c3d10_s4r_e1.inp	Element definitions for the shell-to-solid coupling model with C3D10 and S4R elements.
pipe_shell2solid_c3d10_s4r_e2.inp	Element definitions for the shell-to-solid coupling model with C3D10 and S4R elements.
pipe_shell2solid_c3d10_s4r_e3.inp	Element definitions for the shell-to-solid coupling model with C3D10 and S4R elements.
pipe_shell2solidx_c3d10m_s4r.inp	Abaqus/Explicit shell-to-solid coupling model with C3D10M and S4R elements.
pipe_shell2solidx_c3d10m_s4r_n1.inp	Node definitions for the Abaqus/Explicit shell-to-solid coupling model with C3D10M and S4R elements.
pipe_shell2solidx_c3d10m_s4r_n2.inp	Node definitions for the Abaqus/Explicit shell-to-solid coupling model with C3D10M and S4R elements.
pipe_shell2solidx_c3d10m_s4r_n3.inp	Node definitions for the Abaqus/Explicit shell-to-solid coupling model with C3D10M and S4R elements.
pipe_shell2solidx_c3d10m_s4r_e1.inp	Element definitions for the Abaqus/Explicit shell-to-solid coupling model with C3D10M and S4R elements.
pipe_shell2solidx_c3d10m_s4r_e2.inp	Element definitions for the Abaqus/Explicit shell-to-solid coupling model with C3D10M and S4R elements.
pipe_shell2solidx_c3d10m_s4r_e3.inp	Element definitions for the Abaqus/Explicit shell-to-solid coupling model with C3D10M and S4R elements.
pipe_c3d20r.inp	Reference model with C3D20R elements.
pipe_c3d20r_n.inp	Node definitions for the reference model with C3D20R elements.
pipe_c3d20r_e.inp	Element definitions for the reference model with C3D20R elements.

Dynamic input files

pipe_submodelx_s4r_global.inp	Abaqus/Explicit S4R global model.
pipe_submodelx_s4r_global_n.inp	Node definitions for the Abaqus/Explicit S4R global model.
pipe_submodelx_s4r_global_e.inp	Element definitions for the Abaqus/Explicit S4R global model.
pipe_submodelx_c3d8r_sub_s4r.inp	Abaqus/Explicit C3D8R submodel.
pipe_submodelx_c3d8r_sub_s4r_n.inp	Node definitions for the Abaqus/Explicit C3D8R submodel.
pipe_submodelx_c3d8r_sub_s4r_e.inp	Element definitions for the Abaqus/Explicit C3D8R submodel.
pipe_submodel_c3d8r_sub_s4r.inp	Abaqus/Standard C3D8R submodel.

pipe_submodel_c3d8r_sub_s4r_n.inp	Node definitions for the Abaqus/Standard C3D8R submodel.
pipe_submodel_c3d8r_sub_s4r_e.inp	Element definitions for the Abaqus/Standard C3D8R submodel.

Table 1.1.10–1 Mises stress comparison for static analyses.

	Maximum integration point Mises stress (MPa)	Maximum nodal Mises stress (MPa)
Shell-to-solid submodeling	80.1	97.5
Shell-to-solid coupling	59.8	72.6
Reference	59.9	73.6

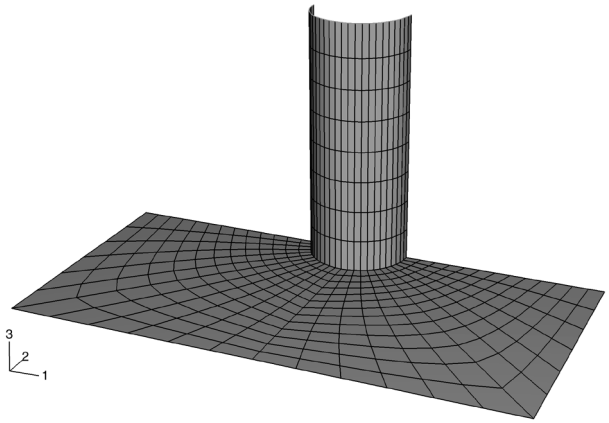


Figure 1.1.10–1 Global shell model of pipe-plate structure.

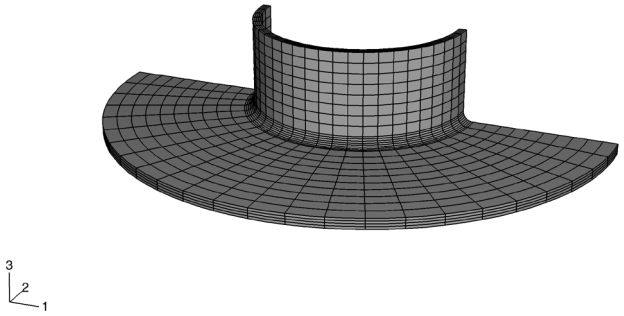


Figure 1.1.10–2 Magnified solid submodel of the pipe-plate joint.

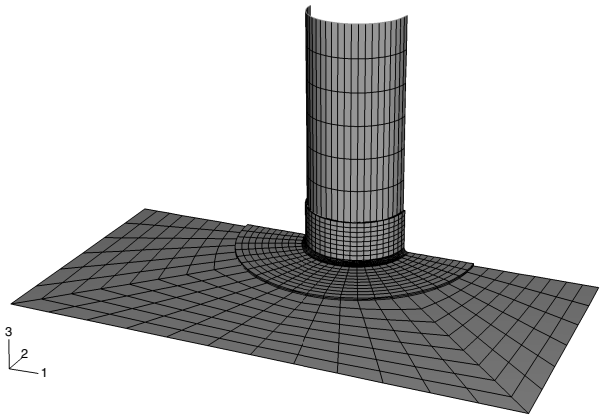


Figure 1.1.10–3 Shell-to-solid coupling model of the pipe-plate joint.

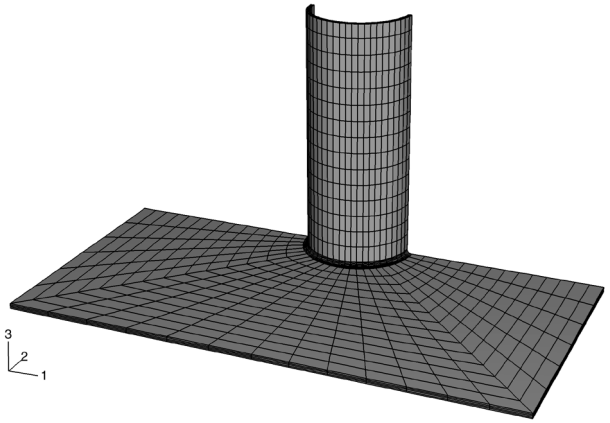


Figure 1.1.10–4 Solid reference model of the pipe-plate joint.

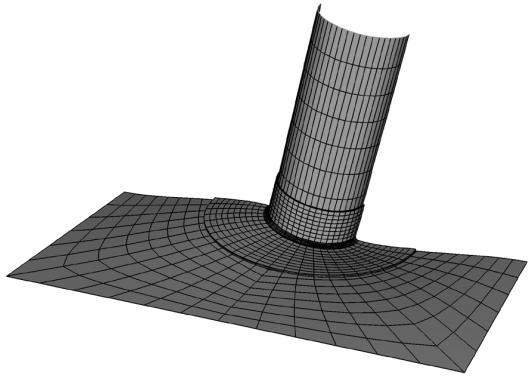


Figure 1.1.10–5 Solid submodel overlaid on the shell model in the deformed state, using a magnification factor of 20.

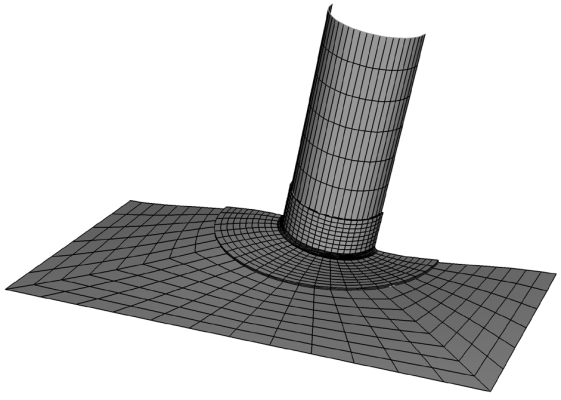


Figure 1.1.10–6 Shell-to-solid coupling model in the deformed state, using a magnification factor of 20.

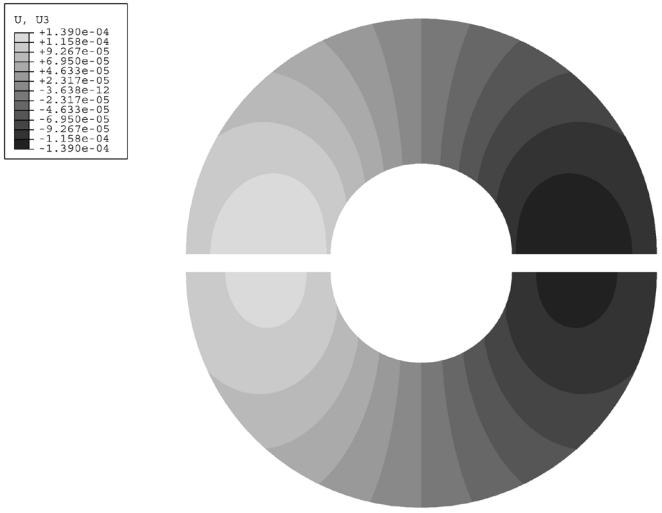


Figure 1.1.10–7 Comparison of out-of-plane displacement in the continuum mesh plate for the submodel (top) and the shell-to-solid coupling analysis (bottom).

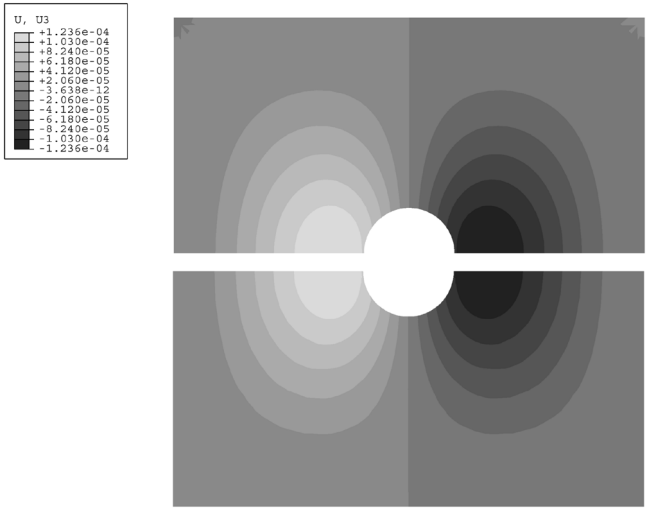


Figure 1.1.10–8 Comparison of out-of-plane displacement in the plate for the reference solution (top) and the shell-to-solid coupling analysis (bottom).

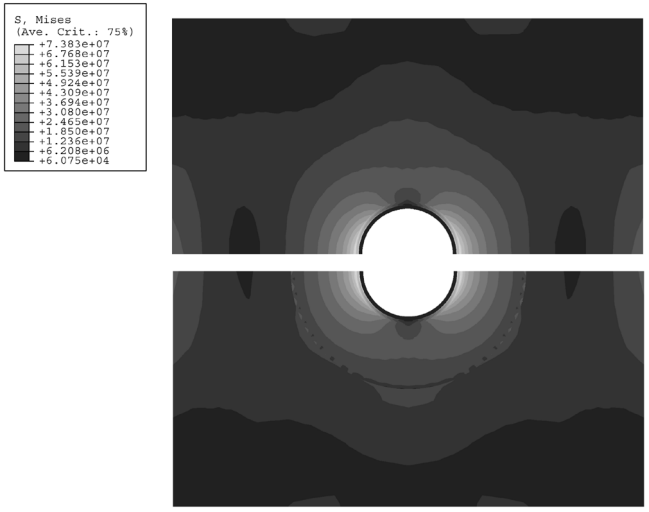


Figure 1.1.10–9 Comparison of the Mises stress in the plate for the reference solution (top) and the shell-to-solid coupling analysis (bottom).

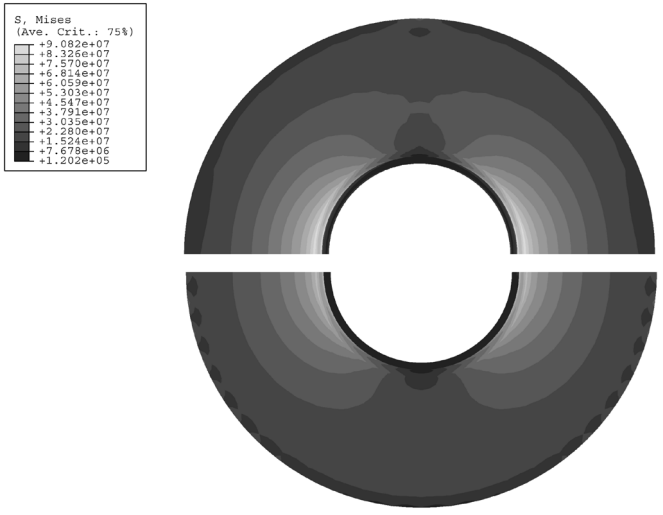


Figure 1.1.10–10 Comparison of the Mises stress in the continuum mesh plate for the submodel (top) and the shell-to-solid coupling analysis (bottom).

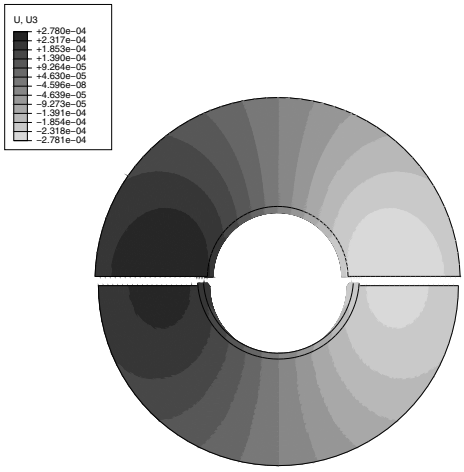


Figure 1.1.10–11 Comparison of out-of-plane displacement in the continuum mesh plate for the submodel with scaled boundary (top) and the shell-to-solid coupling analysis with scaled load (bottom).

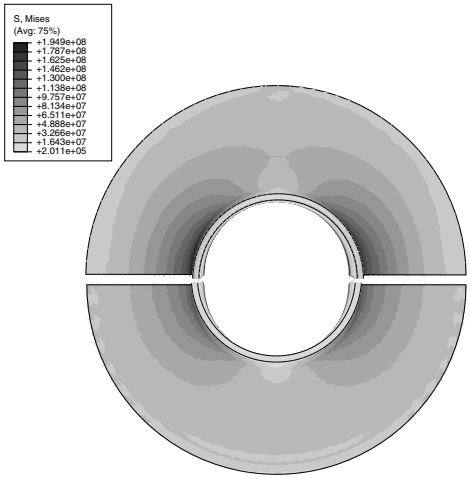


Figure 1.1.10–12 Comparison of the Mises stress in the continuum mesh plate for the submodel with scaled boundary condition (top) and the shell-to-solid coupling analysis with scaled load (bottom).

1.1.11 STRESS-FREE ELEMENT REACTIVATION

Product: Abaqus/Standard

This example demonstrates element reactivation for problems where new elements are to be added in a stress-free state. Typical examples include the construction of a gravity dam, in which unstressed layers of material are added to a mesh that has already deformed under geostatic load, or a tunnel in which a concrete or steel support liner is installed. Element pair reactivation during a step ("Element and contact pair removal and reactivation," Section 11.2.1 of the Abaqus Analysis User's Guide) provides for this type of application directly because the strain in newly added elements corresponds to the deformation of the mesh since the reactivation.

Verification of the element pair reactivation capability is provided in "Model change," Section 3.10 of the Abaqus Verification Guide.

Problem description

The example considers the installation of a concrete liner to support a circular tunnel. Practical geotechnical problems usually involve a complex sequence of construction steps. The construction details determine the appropriate analysis method to represent these steps accurately. Such details have been avoided here for the sake of simplifying the illustration.

The tunnel is assumed to be excavated in clay, with a Young's modulus of 200 MPa and a Poisson's ratio of 0.2 (see Figure 1.1.11–1). The diameter of the tunnel is 8 m, and the tunnel is excavated 20 m below ground surface. The material surrounding the excavation is discretized with first-order 4-node plane strain elements (element type CPE4). The infinite extent of the soil is represented by a 30-m-wide mesh that extends from the surface to a depth of 50 m below the surface. The left-hand boundary represents a vertical symmetry axis. Far-field conditions on the bottom and right-hand-side boundaries are modeled by infinite elements (element type CINPE4). No mesh convergence studies have been performed to establish if these boundary conditions are placed far enough away from the excavation.

An initial stress field due to gravitational and tectonic forces exists through the depth of the soil. It is assumed that this stress varies linearly with depth and that the ratio between the horizontal and vertical stress components is 0.5. The self weight of the clay is 20.0 kN/m^3.

The excavation of the tunnel material is accomplished by applying the forces that are required to maintain equilibrium with the initial stress state in the surrounding material as loads on the perimeter of the tunnel. These loads are then reduced to zero to simulate the excavation. The three-dimensional effect of face advancement during excavation is taken into account by relaxing the forces gradually over several steps. The liner is installed after 40% relaxation of the loads. Further deformation continues to occur as the face of the excavation advances. This ongoing deformation loads the liner.

In the first input file the 150-mm-thick liner is discretized with one layer of incompatible mode elements (element type CPE4I). These elements are recommended in regions where bending response must be modeled accurately. In the second input file beam elements are used to discretize the liner. The liner is attached rigidly to the tunnel. The concrete is assumed to have cured to a strength represented by

the elastic properties shown in Figure 1.1.11–1 by the time the liner is loaded. The liner is not shown in this diagram.

It is expected that an overburden load representing the weight of traffic and buildings exists after the liner is installed.

Analysis method

The excavation and installation of the liner is modeled in four analysis steps. In the first step the initial stress state is applied and the liner elements are removed. Concentrated loads that are in equilibrium with the initial stress field are applied on the perimeter of the tunnel. These forces were obtained from an independent analysis where the displacements on the tunnel perimeter were constrained. The reaction forces at the constrained nodes are the loads applied here. The second step begins the tunnel excavation by reducing the concentrated loads on the tunnel surface. The loads are reduced by 40% in this step before the liner is installed in the third step. No deformation takes place in the soil or liner during the third step. In the fourth step the surface load is applied, and the excavation is completed by removing the remainder of the load on the tunnel perimeter.

In problems involving geometric nonlinearities with finite deformation, it is important to recognize that element reactivation occurs in the configuration at the start of the reactivation step. If the NLGEOM parameter were used in this problem, the thickness of the liner, when modeled with the continuum elements, would have a value at reactivation that would be different from its original value. This result would happen because the outside nodes (the nodes on the tunnel/liner interface) displace with the mesh, whereas the inside nodes remain at their current locations since liner elements are inactive initially. This effect is not relevant in this problem because geometric nonlinearities are not included. However, it may be significant for problems involving finite deformation, and it may lead to convergence problems in cases where elements are severely distorted upon reactivation. This problem would not occur in the model with beam elements because they have only one node through the thickness. In the model where the liner is modeled with continuum elements, the problem can be eliminated if the inner nodes are allowed to follow the outer nodes prior to reactivation, which can be accomplished by applying displacement boundary conditions on the inner nodes. Alternatively, the liner can be overlaid with (elastic) elements of very low stiffness. These elements use the same nodes as the liner but are so compliant that their effect on the analysis is negligible when the liner is present. They remain active throughout the analysis and ensure that the inner nodes follow the outer nodes, thereby preserving the liner thickness.

Results and discussion

Figure 1.1.11–2 shows the stress state at a material point in the liner. The figure clearly indicates that the liner remains unstressed until reactivated.

Figure 1.1.11–3 compares the axial stress obtained from the CPE4I and beam elements at the top and bottom of the liner section. A local cylindrical coordinate system (“Orientations,” Section 2.2.5 of the Abaqus Analysis User’s Guide) is used to orient the liner stresses in the continuum element model along the beam axis so that these stresses can be compared directly with the results of the beam element model. The small difference between the results can be attributed to the element type used in the discretization

of the liner: the beam element model uses a plane stress condition, and the continuum element model uses a plane strain condition.

Input files

modelchangedemo_continuum.inp	*MODEL CHANGE with continuum elements.
modelchangedemo_beam.inp	*MODEL CHANGE with beam elements.
modelchangedemo_node.inp	Nodal coordinates for the soil.
modelchangedemo_element.inp	Element definitions for the soil.

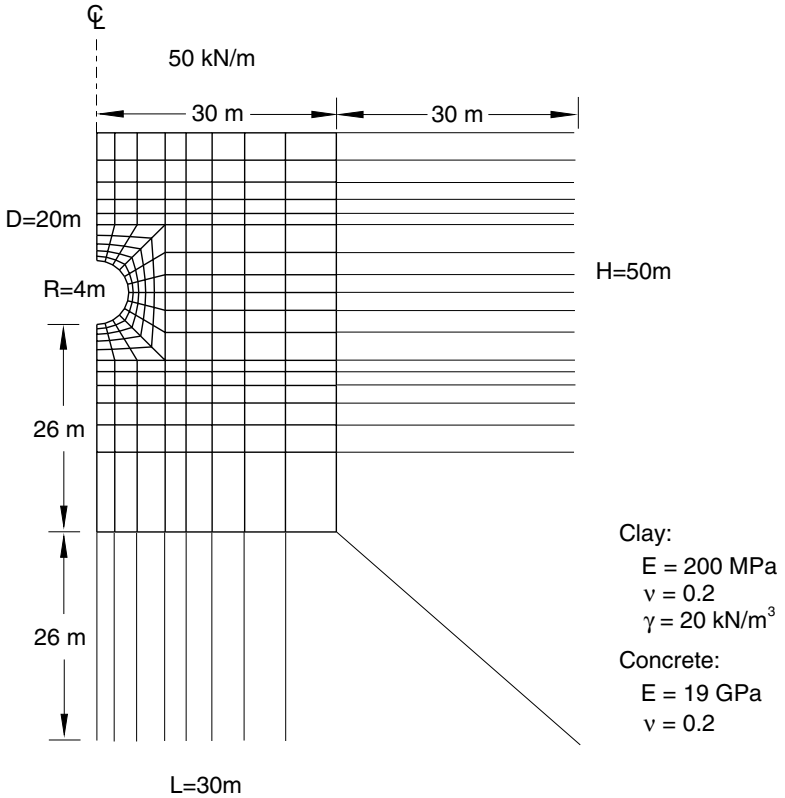


Figure 1.1.11–1 Geometry and finite element discretization.

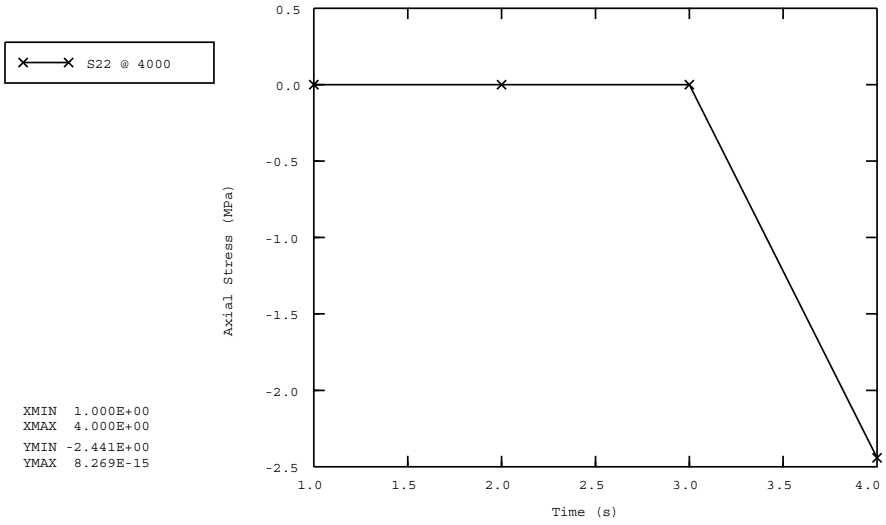


Figure 1.1.11–2 Liner stress during analysis history.

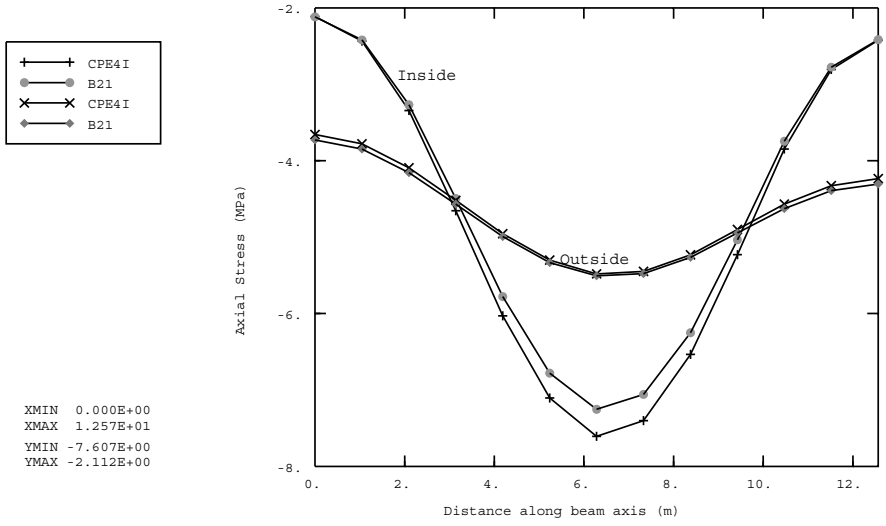


Figure 1.1.11–3 Axial stress along beam inside and outside.

1.1.12 TRANSIENT LOADING OF A VISCOELASTIC BUSHING

Product: Abaqus/Standard

This example demonstrates the automatic incrementation capability provided for integration of time-dependent material models and the use of the viscoelastic material model in conjunction with large-strain hyperelasticity in a typical design application. The structure is a bushing, modeled as a hollow, viscoelastic cylinder. The bushing is glued to a rigid, fixed body on the outside and to a rigid shaft on the inside, to which the loading is applied. A static preload is applied to the shaft, which moves the inner shaft off center. This load is held for sufficient time for steady-state response to be obtained. Then a torque is applied instantaneously and held for a long enough period of time to reach steady-state response. We compute the bushing's transient response to these events.

Geometry and model

The viscoelastic bushing has an inner radius of 12.7 mm (0.5 in) and an outer radius of 25.4 mm (1.0 in). We assume that the bushing is long enough for plane strain deformation to occur. The problem is modeled with first-order reduced-integration elements (CPE4R). The mesh is regular, consisting of 6 elements radially, repeated 56 times to cover the 360° span in the hoop direction. The mesh is shown in Figure 1.1.12–1. No mesh convergence studies have been performed.

The fixed outer body is modeled by fixing both displacement components at all the outside nodes. The nodes in the inner boundary of the bushing are connected, using a kinematic coupling constraint, to a node located in the center of the model. This node, thus, defines the inner shaft as a rigid body.

Material

The material model is not defined from any particular physical material.

The instantaneous behavior of the viscoelastic material is defined by hyperelastic properties. A polynomial model with $N = 1$ (a Mooney-Rivlin model) is used for this, with the constants $C_{10} = 27.56$ MPa (4000 psi), $C_{01} = 6.89$ MPa (1000 psi), and $D_1 = 0.0029$ MPa^{-1} (0.00002 psi^{-1}).

The viscous behavior is modeled by a time-dependent shear modulus, $G_R(t)$, and a time-dependent bulk modulus, $K_R(t)$, each of which is expanded in a Prony series in terms of the corresponding instantaneous modulus,

$$G_R(t)/G_0 = 1 - \sum_{i=1}^{2} \bar{g}_i^P \left(1 - \exp\left(-\frac{t}{\tau_i}\right)\right)$$

$$K_R(t)/K_0 = 1 - \sum_{i=1}^{2} \bar{k}_i^P \left(1 - \exp\left(-\frac{t}{\tau_i}\right)\right).$$

The relative moduli $\bar{g}_i^P$ and $\bar{k}_i^P$ and time constants τ_i are

i	$\bar{g}_i^P$	$\bar{k}_i^P$	τ_i, **sec**
1	0.2	0.5	0.1
2	0.1	0.2	0.2

This model results in an initial instantaneous Young's modulus of 206.7 MPa (30000 psi) and Poisson's ratio of 0.45. It relaxes pressures faster than shear stresses.

Analysis

The analysis is done in four steps. The first step is a preload of 222.4 kN (50000 lbs) applied in the x-direction to the node in the center of the model in 0.001 sec with a static procedure ("Static stress analysis," Section 6.2.2 of the Abaqus Analysis User's Guide). The static procedure does not allow viscous material behavior, so this response is purely elastic. During the second step the load stays constant and the material is allowed to creep for 1 sec by using the quasi-static procedure ("Quasi-static analysis," Section 6.2.5 of the Abaqus Analysis User's Guide). Since 1 sec is a long time compared with the material time constants, the solution at that time should be close to steady state. The accuracy of the automatic time incrementation during creep response can be controlled. This accuracy tolerance is an upper bound on the allowable error in the creep strain increment in each time increment. It is chosen as 5×10^{-4}, which is small compared to the elastic strains. The third step is another static step. Here the loading is a torque of 1129.8 N-m (10000 lb-in) applied in 0.001 sec. The fourth step is another quasi-static step with a time period of 1 sec.

Results and discussion

Figure 1.1.12–2 through Figure 1.1.12–5 depict the deformed shape of the bushing at the end of each step. Each of the static loads produces finite amounts of deformation, which are considerably expanded during the holding periods. Figure 1.1.12–6 shows the displacement of the center of the bushing in the x-direction and its rotation as functions of time.

Input file

viscobushing.inp Input data for the analysis.

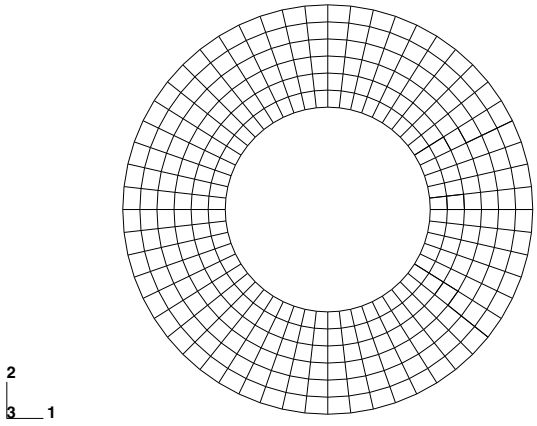


Figure 1.1.12–1 Finite element model of viscoelastic bushing.

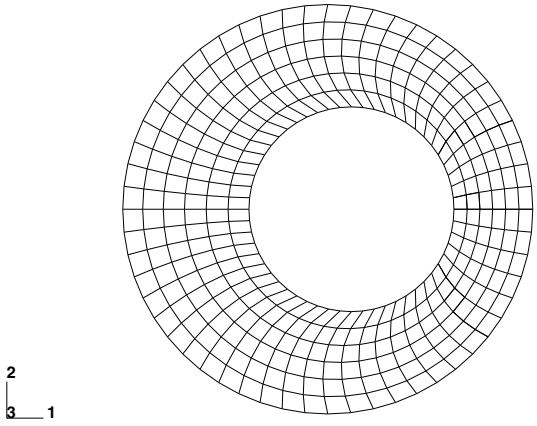


Figure 1.1.12–2 Deformed model after horizontal static loading.

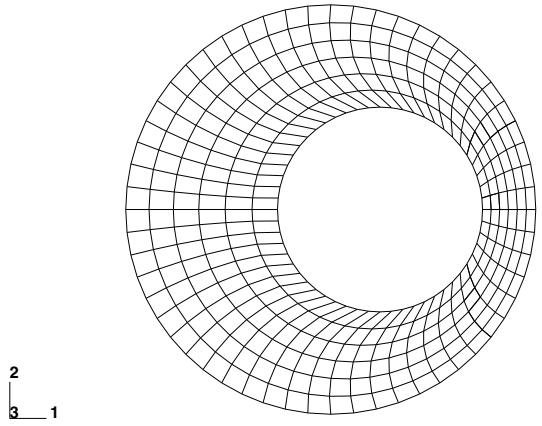


Figure 1.1.12–3 Deformed model after first holding period.

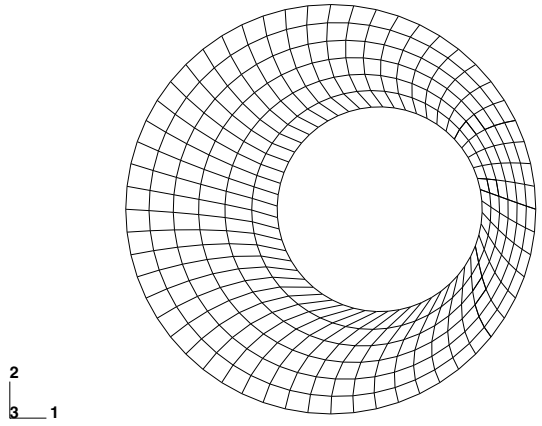


Figure 1.1.12–4 Deformed model after static moment loading.

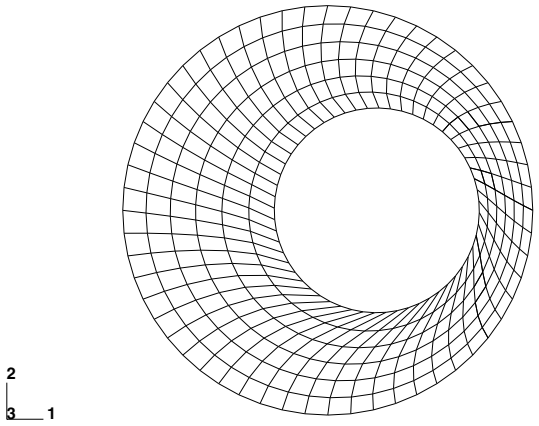


Figure 1.1.12–5 Deformed model after second holding period.

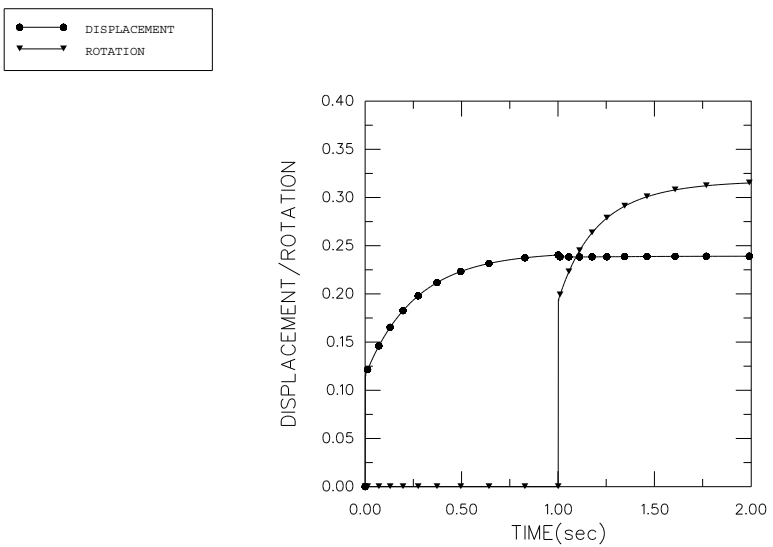


Figure 1.1.12–6 Displacement and rotation of center of bushing.

1.1.13 INDENTATION OF A THICK PLATE

Product: Abaqus/Explicit

This example illustrates the use of adaptive meshing and distortion control in deep indentation problems.

Problem description

A deep indentation problem is solved for both axisymmetric and three-dimensional geometries, as shown in Figure 1.1.13–1. Each model consists of a rigid punch and a deformable blank. The punch has a semicircular nose section and a radius of 100 mm. The blank is modeled as a crushable foam with the elastic response given as follows (see Schluppkotten, 1999):

$E = 7.5$ MPa (Young's modulus) and
$\nu = 0.0$ (elastic Poisson's ratio).

The material parameters for the isotropic hardening are given as

$k = \sigma_c^0/p_c^0 = 1.0$ (yield strength ratio) and
$\nu_p = 0.0$ (plastic Poisson's ratio),

and the density is

$\rho = 60$ kg/m^3.

In both cases the punch is fully constrained except in the vertical direction. A deep indentation is made by moving the punch into the blank to a depth of 250 mm when adaptive meshing is used and to a depth of 285 mm when distortion control is used. The displacement of the punch is prescribed using a smooth-step amplitude so that a quasi-static response is generated.

Case 1: Axisymmetric model

The blank is meshed with CAX4R elements and measures 300 × 300 mm. The punch is modeled as an analytical rigid surface using a planar analytical surface in conjunction with a rigid body constraint. The bottom of the blank is constrained in the *x*- and *z*-directions, and symmetry boundary conditions are prescribed at *r*=0.

Case 2: Three-dimensional models

Two models are analyzed. For one model the blank is meshed uniformly, while for the other a graded mesh is used. For both models the blank is meshed with C3D8R elements and measures 600 × 300 × 600 mm. The punch is modeled as an analytical rigid surface using a three-dimensional surface of revolution in conjunction with a rigid body constraint. The bottom of the blank is fully constrained.

Adaptive meshing

A single adaptive mesh domain that incorporates the entire blank is used for each model. A Lagrangian boundary region type (the default) is used to define the constraints along the bottom of the plate for both models and along the axis of symmetry in two dimensions. A sliding boundary region (the default) is used to define the contact surface on the plate. To obtain a good mesh throughout the simulation, the number of mesh sweeps is increased to 3 as part of the specification of the adaptive mesh domain. For the graded three-dimensional model a graded smoothing objective is specified to preserve the gradation of the mesh while adaptive meshing is performed.

Distortion control

In contrast to the adaptive meshing technique, distortion control does not attempt to maintain a high-quality mesh throughout an analysis but instead tries to prevent negative element volumes or other excessive distortion from occurring during an analysis. By using distortion control, it is possible to prevent an analysis from failing prematurely when the mesh is coarse relative to the strain gradients and the amount of compression. The distortion control capability is tested for axisymmetric and three-dimensional models with a uniformly meshed blank.

Results and discussion

Figure 1.1.13–2 to Figure 1.1.13–4 show the initial configurations for the axisymmetric model, the three-dimensional uniform mesh model, and the three-dimensional graded mesh model. Although the punch is not shown in these figures, it is initially in contact with the plate. Figure 1.1.13–5 shows the final deformed mesh for the axisymmetric indentation. The meshing algorithm attempts to minimize element distortion both near and away from the contact surface with the punch. Figure 1.1.13–6 and Figure 1.1.13–7 show the deformed mesh of the entire blank and a quarter-symmetry, cutaway view, respectively, for the three-dimensional model with an initially uniform mesh. Even under this depth of indentation, elements appear to be nicely shaped both on the surface and throughout the cross-section of the plate.

Figure 1.1.13–8 and Figure 1.1.13–9 show the deformed mesh of the entire plate and a quarter-symmetry, cutaway view, respectively, for the three-dimensional case with an initially graded mesh. Adaptive meshing with the graded smoothing objective preserves the mesh gradation throughout the indentation process while simultaneously minimizing element distortion. Preserving mesh gradation in adaptivity problems is a powerful capability that allows mesh refinement to be concentrated in the areas of highest strain gradients. A contour plot of equivalent plastic strain for the graded mesh case is shown in Figure 1.1.13–10.

Figure 1.1.13–11 shows the final deformed mesh for the axisymmetric indentation using distortion control without adaptive meshing. Figure 1.1.13–12 and Figure 1.1.13–13 show the deformed mesh of the entire blank and a quarter-symmetry, cutaway view, respectively, for the three-dimensional model with an initially uniform mesh using distortion control without adaptive meshing. The distortion control simply prevents element distortion near the contact surface with the punch. Without distortion control both of the analyses fail prematurely under this depth of indentation.

Input files

ale_indent_axi.inp	Case 1 using adaptive meshing.
ale_indent_sph.inp	Case 2 with a uniform mesh using adaptive meshing.
ale_indent_gradedsph.inp	Case 2 with a graded mesh using adaptive meshing.
ale_indent_sphelset.inp	External file referenced by Case 2.
dis_indent_axi.inp	Case 1 using distortion control.
dis_indent_sph.inp	Case 2 with a uniform mesh using distortion control.

Reference

- Schluppkotten, J., *Investigation of the ABAQUS Crushable Foam Plasticity Model,* Internal report of BMW AG, 1999.

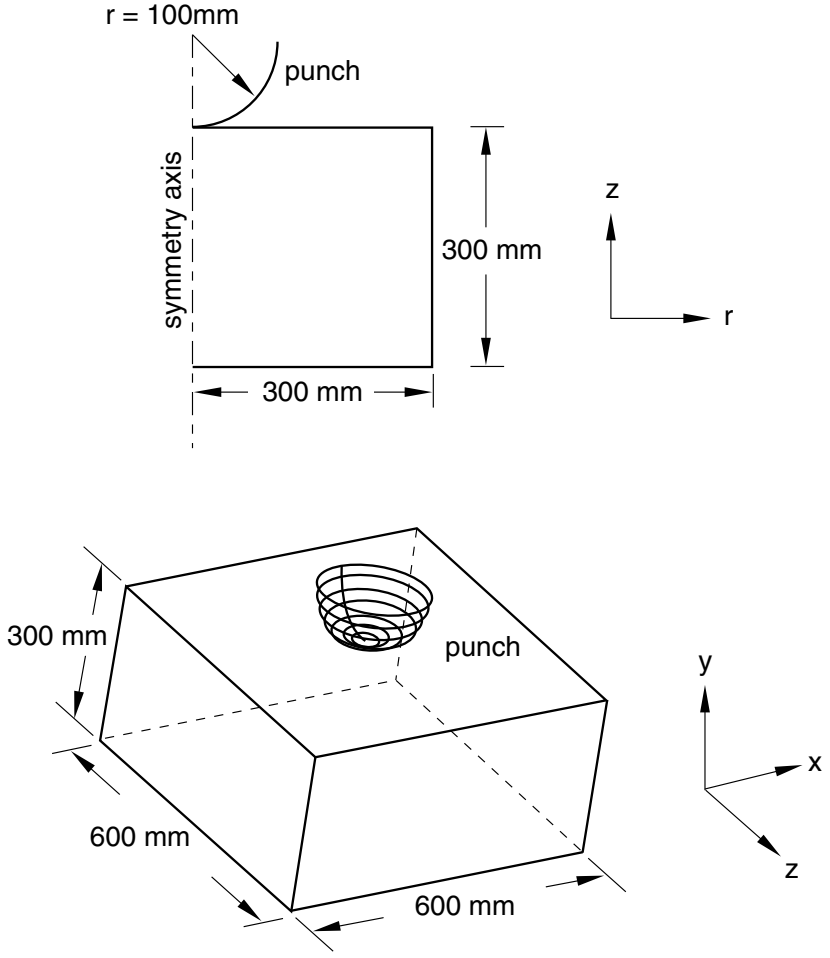


Figure 1.1.13–1 Axisymmetric and three-dimensional model geometries.

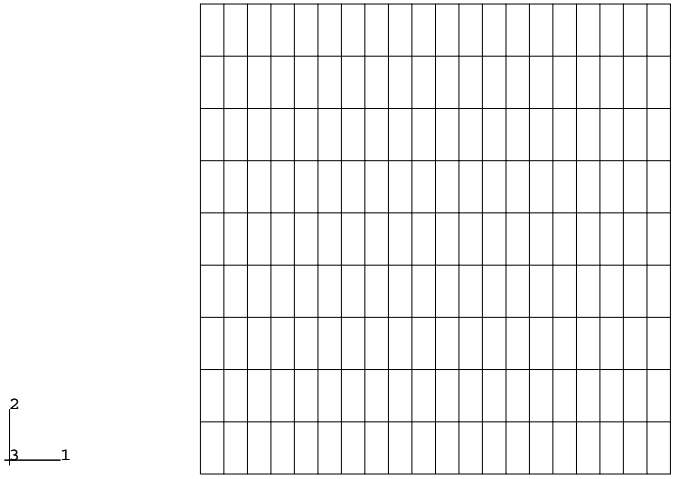


Figure 1.1.13–2 Initial configuration for the axisymmetric model.

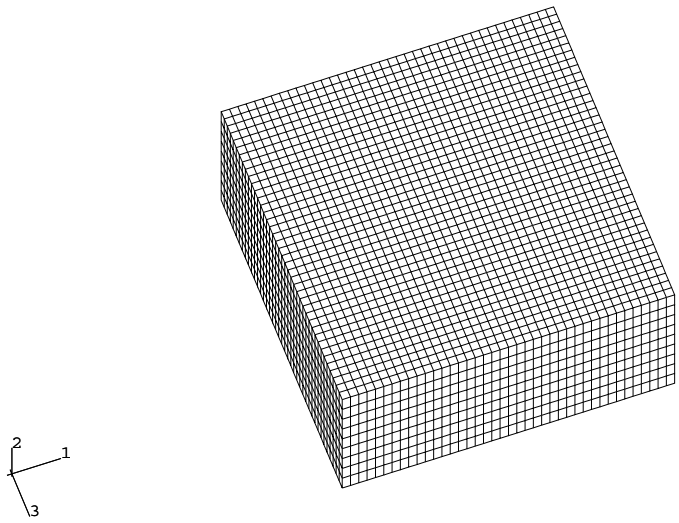


Figure 1.1.13–3 Initial configuration for the three-dimensional model with a uniform mesh.

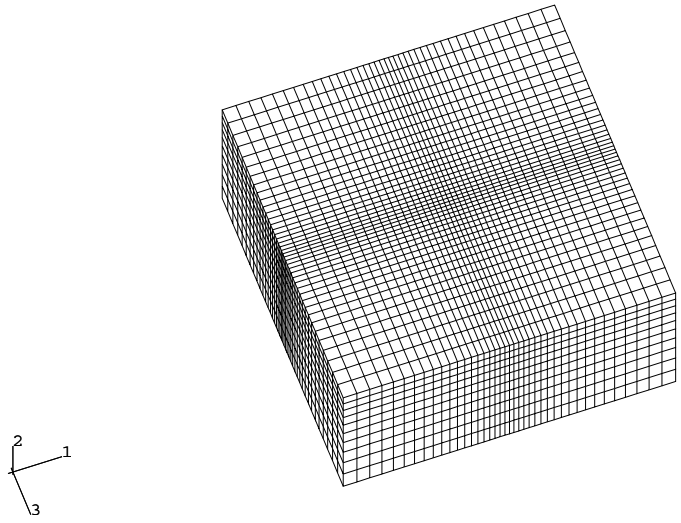


Figure 1.1.13–4 Initial configuration for the three-dimensional model with a graded mesh.

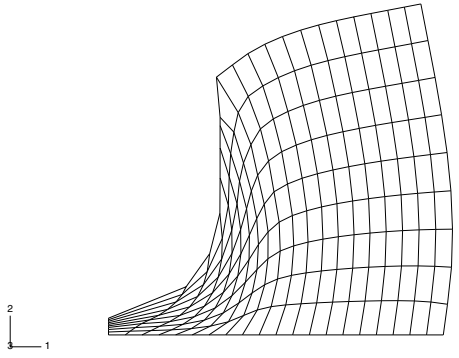


Figure 1.1.13–5 Deformed configuration for the axisymmetric model.

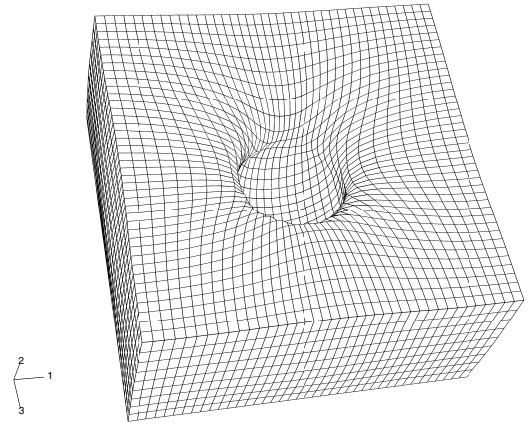


Figure 1.1.13–6 Deformed configuration for the three-dimensional model with an initially uniform mesh.

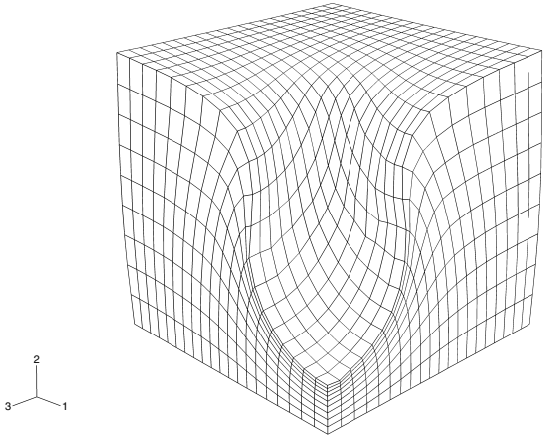


Figure 1.1.13–7 Quarter-symmetry, cutaway view of the deformed configuration for the three-dimensional model with an initially uniform mesh.

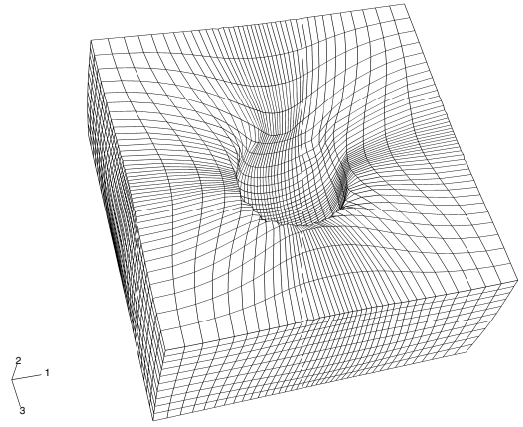


Figure 1.1.13–8 Deformed configuration for the three-dimensional model with an initially graded mesh.

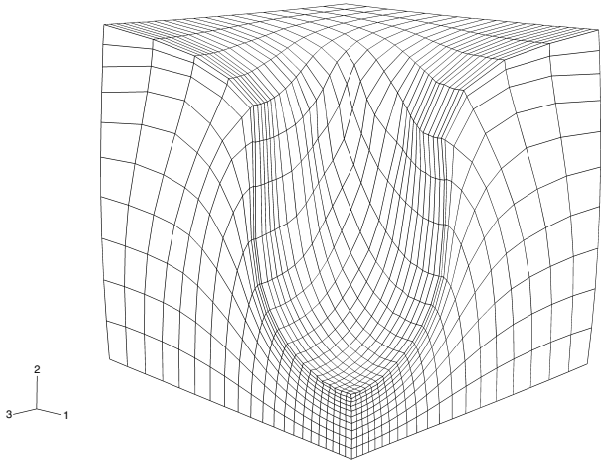


Figure 1.1.13–9 Quarter-symmetry, cutaway view of the deformed configuration for the three-dimensional model with an initially graded mesh.

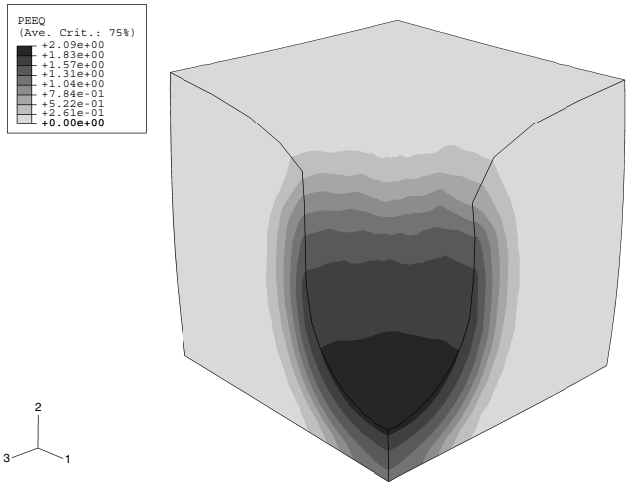


Figure 1.1.13–10 Contours of equivalent plastic strain for the three-dimensional model with an initially graded mesh.

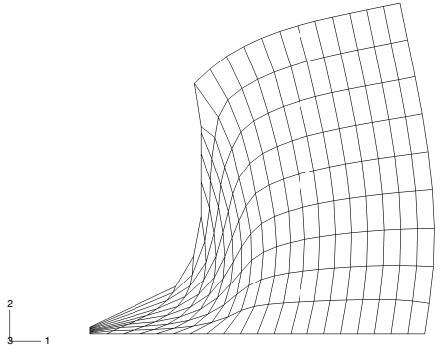


Figure 1.1.13–11 Deformed configuration for the axisymmetric model using distortion control.

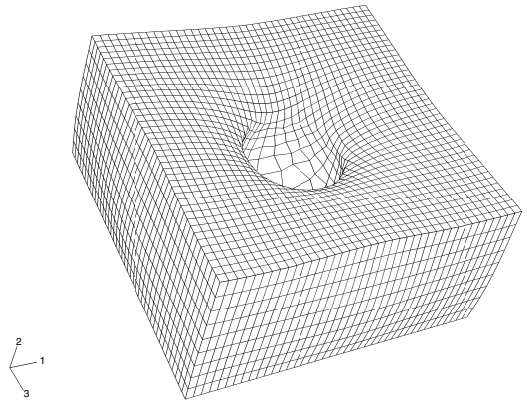


Figure 1.1.13–12 Deformed configuration for the three-dimensional model with an initially uniform mesh using distortion control.

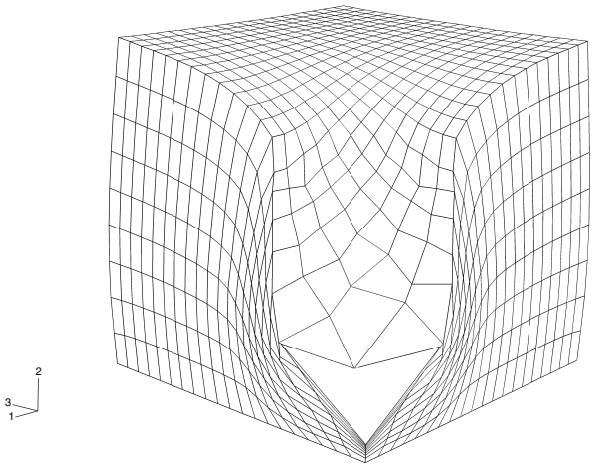


Figure 1.1.13–13 Quarter-symmetry, cutaway view of the deformed configuration for the three-dimensional model with an initially uniform mesh using distortion control.

1.1.14 DAMAGE AND FAILURE OF A LAMINATED COMPOSITE PLATE

Products: Abaqus/Standard Abaqus/Explicit

This example demonstrates how the nonlinear material behavior of a composite laminate can be specified as a function of solution-dependent variables. The user subroutines **USDFLD** in Abaqus/Standard and **VUSDFLD** in Abaqus/Explicit can be used to modify the standard linear elastic material behavior (for example, to include the effects of damage) or to change the behavior of the nonlinear material models in Abaqus. The material model in this example includes damage, resulting in nonlinear behavior. It also includes various modes of failure, resulting in abrupt loss of stress carrying capacity (Chang and Lessard, 1989). The analysis results are compared with experimental results.

Problem description and material behavior

A composite plate with a hole in the center is subjected to in-plane compression. The plate is made of 24 plies of T300/976 graphite-epoxy in a $[(-45/+45)_6]_s$ layup. Each ply has a thickness of 0.1429 mm (0.005625 in); thus, the total plate thickness is 3.429 mm (0.135 in). The plate has a length of 101.6 mm (4.0 in) and a width of 25.4 mm (1.0 in), and the diameter of the hole is 6.35 mm (0.25 in). The plate is loaded in compression in the length direction. The thickness of the plate is sufficient that out-of-plane displacements of the plate can be ignored. The compressive load is measured, as well as the length change between two points, originally a distance of 25.4 mm (1.0 in) apart, above and below the hole. The plate geometry is shown in Figure 1.1.14–1.

The material behavior of each ply is described in detail by Chang and Lessard. The initial elastic ply properties are longitudinal modulus E_x=156512 MPa (22700 ksi), transverse modulus E_y=12962 MPa (1880 ksi), shear modulus G_{xy}=6964 MPa (1010 ksi), and Poisson's ratio ν_{xy}=0.23. The material accumulates damage in shear, leading to a nonlinear stress-strain relation of the form

$$\gamma_{xy} = G_{xy}^{-1}\sigma_{xy} + \alpha\sigma_{xy}^{3},$$

where G_{xy} is the (initial) ply shear modulus and the nonlinearity is characterized by the factor α=2.44×10^{-8} MPa^{-3} (0.8×1^{-5} ksi^{-3}).

Failure modes in laminated composites are strongly dependent on geometry, loading direction, and ply orientation. Typically, one distinguishes in-plane failure modes and transverse failure modes (associated with interlaminar shear or peel stress). Since this composite is loaded in-plane, only in-plane failure modes need to be considered, which can be done for each ply individually. For a unidirectional ply as used here, five failure modes can be considered: matrix tensile cracking, matrix compression, fiber breakage, fiber matrix shearing, and fiber buckling. All the mechanisms, with the exception of fiber breakage, can cause compression failure in laminated composites.

The failure strength in laminates also depends on the ply layup. The effective failure strength of the layup is at a maximum if neighboring plies are orthogonal to each other. The effective strength decreases as the angle between plies decreases and is at a minimum if plies have the same direction. (This is called a ply cluster.) Chang and Lessard have obtained some empirical formulas for the effective

transverse tensile strength; however, in this model we ignore such effects. Instead, we use the following strength properties for the T300/976 laminate: transverse tensile strength Y_t=102.4 MPa (14.86 ksi), ply shear strength S_c=106.9 MPa (15.5 ksi), matrix compressive strength Y_c=253.0 MPa (36.7 ksi), and fiber buckling strength X_c=2707.6 MPa (392.7 ksi).

The strength parameters can be combined into failure criteria for multiaxial loading. Four different failure modes are considered in the model analyzed here.

- *Matrix tensile cracking* can result from a combination of transverse tensile stress, σ_y, and shear stress, σ_{xy}. The failure index, e_m, can be defined in terms of these stresses and the strength parameters, Y_t and S_c. When the index exceeds 1.0, failure is assumed to occur. Without nonlinear material behavior, the failure index has the simple form,

$$e_m^2 = \left(\frac{\sigma_y}{Y_t}\right)^2 + \left(\frac{\sigma_{xy}}{S_c}\right)^2.$$

  With nonlinear shear behavior taken into consideration, the failure index takes the more complex form,

$$e_m^2 = \left(\frac{\sigma_y}{Y_t}\right)^2 + \frac{2\sigma_{xy}^2/G_{xy} + 3\alpha\sigma_{xy}^4}{2S_c^2/G_{xy} + 3\alpha S_c^4}.$$

- *Matrix compressive failure* results from a combination of transverse compressive stress and shear stress. The failure criterion has the same form as that for matrix tensile cracking:

$$e_m^2 = \left(\frac{\sigma_y}{Y_c}\right)^2 + \frac{2\sigma_{xy}^2/G_{xy} + 3\alpha\sigma_{xy}^4}{2S_c^2/G_{xy} + 3\alpha S_c^4}.$$

  The same failure index is used since the previous two failure mechanisms cannot occur simultaneously at the same point. After the failure index exceeds 1.0, both the transverse stiffness and Poisson's ratio of the ply drop to zero.

- *Fiber-matrix shearing failure* results from a combination of fiber compression and matrix shearing. The failure criterion has essentially the same form as the other two criteria:

$$e_{fs}^2 = \left(\frac{\sigma_x}{X_c}\right)^2 + \frac{2\sigma_{xy}^2/G_{xy} + 3\alpha\sigma_{xy}^4}{2S_c^2/G_{xy} + 3\alpha S_c^4}.$$

  This mechanism can occur simultaneously with the other two criteria; hence, a separate failure index is used. Shear stresses are no longer supported after the failure index exceeds 1.0, but direct stresses in the fiber and transverse directions continue to be supported.

- *Fiber buckling failure* occurs when the maximum compressive stress in the fiber direction ($-\sigma_x$) exceeds the fiber buckling strength, X_c, independent of the other stress components:

$$e_b = -\frac{\sigma_x}{X_c}.$$

It is obvious that, unless the shear stress vanishes exactly, fiber-matrix shearing failure occurs prior to fiber buckling. However, fiber buckling may follow subsequent to fiber shearing because only the shear stiffness degrades after fiber-matrix shearing failure. Fiber buckling in a layer is a catastrophic mode of failure. Hence, after this failure index exceeds 1.0, it is assumed that the material at this point can no longer support any loads.

In this example the primary loading mode is shear. Therefore, failure of the plate occurs well before the fiber stresses can develop to a level where fiber buckling takes place, and this failure mode need not be taken into consideration.

Chang and Lessard assume that after failure occurs, the stresses in the failed directions drop to zero immediately, which corresponds to brittle failure with no energy absorption. This kind of failure model usually leads to immediate, unstable failure of the composite. This assumption is not very realistic: in reality, the stress-carrying capacity degrades gradually with increasing strain after failure occurs. Hence, the behavior of the composite after onset of failure is not likely to be captured well by this model. Moreover, the instantaneous loss of stress-carrying capacity also makes the postfailure analysis results strongly dependent on the refinement of the finite element mesh and the finite element type used.

Material model implementation

To simulate the shear nonlinearity and the failure modes (matrix failure in tension or compression and fiber-matrix shear failure), the elastic properties are made linearly dependent on three field variables. The first field variable represents the matrix failure index, the second represents the fiber-matrix shear failure index, and the third represents the shear nonlinearity (damage) prior to failure. The dependence of the elastic material properties on the field variables is shown in Table 1.1.14–1.

To account for the nonlinearity, the nonlinear stress-strain relation must be expressed in a different form: the stress at the end of the increment must be given as a linear function of the strain. The most obvious way to do this is to linearize the nonlinear term, leading to the relation

$$\gamma_{xy}^{(i+1)} = \left(G_{xy}^{-1} + \alpha (\sigma_{xy}^{(i)})^2 \right) \sigma_{xy}^{(i+1)},$$

where i represents the increment number. This relation can be written in inverted form as

$$\sigma_{xy}^{(i+1)} = \frac{G_{xy}}{1 + \alpha G_{xy} (\sigma_{xy}^{(i)})^2} \gamma_{xy}^{(i+1)},$$

thus providing an algorithm to define the effective shear modulus.

However, this algorithm is not very suitable because it is unstable at higher strain levels, which is readily demonstrated by stability analysis. Consider an increment where the strain does not change; i.e., $\gamma_{xy}^{(i+1)} = \gamma_{xy}^{(i)} = \gamma_{xy}$. Let the stress at increment i have a small perturbation from ${\sigma_{xy}^{e}}^{(i)}$, the exact solution at that increment: $\sigma_{xy}^{(i)} = {\sigma_{xy}^{e}}^{(i)} + \Delta\sigma_{xy}^{(i)}$. Similarly, at increment i+1, $\sigma_{xy}^{(i+1)} = {\sigma_{xy}^{e}}^{(i+1)} + \Delta\sigma_{xy}^{(i+1)}$. For the algorithm to be stable, $\Delta\sigma_{xy}^{(i+1)}$ should not be larger than $\Delta\sigma_{xy}^{(i)}$. The perturbation in increment i+1 is calculated by substituting $\sigma_{xy}^{(i)}$ in the effective shear modulus equation and linearizing it about ${\sigma_{xy}^{e}}^{(i)}$:

$$\Delta\sigma_{xy}^{(i+1)} = \frac{-2\alpha G_{xy}^2 \sigma_{xy}\gamma_{xy}}{(1+\alpha G_{xy}\sigma_{xy}^2)^2}\Delta\sigma_{xy}^{(i)},$$

where $\sigma_{xy} = \sigma_{xy}^{e\ (i)}$. The perturbation in increment i+1 is larger than the perturbation in increment i if

$$2\alpha G_{xy}^2 \sigma_{xy}\gamma_{xy} > (1+\alpha G_{xy}\sigma_{xy}^2)^2,$$

which, after elimination of γ_{xy}, reduces to the expression

$$\alpha\sigma_{xy}^3 > G_{xy}^{-1}\sigma_{xy}.$$

Hence, instability occurs when the "nonlinear" part of the shear strain is larger than the "linear" part of the shear strain.

To obtain a more stable algorithm, we write the nonlinear stress-strain law in the form

$$\gamma_{xy} + \beta\sigma_{xy}^3 = G_{xy}^{-1}\sigma_{xy} + (\alpha+\beta)\sigma_{xy}^3,$$

where β is an as yet unknown coefficient. In linearized form this leads to the update algorithm

$$(1+\beta(\sigma_{xy}^{(i)})^3/\gamma_{xy}^{(i)})\gamma_{xy}^{(i+1)} = \left(G_{xy}^{-1} + (\alpha+\beta)(\sigma_{xy}^{(i)})^2\right)\sigma_{xy}^{(i+1)},$$

or, in inverted form,

$$\sigma_{xy}^{(i+1)} = \frac{1+\beta(\sigma_{xy}^{(i)})^3/\gamma_{xy}^{(i)}}{1+(\alpha+\beta)G_{xy}(\sigma_{xy}^{(i)})^2}G_{xy}\gamma_{xy}^{(i+1)}.$$

Following the same procedure as that for the original update algorithm, it is readily derived that a small perturbation, $\Delta\sigma_{xy}^{(i)}$, in increment i reduces to zero in increment i+1 if $\beta = 2\alpha$. Hence, the optimal algorithm appears to be

$$\sigma_{xy}^{(i+1)} = \frac{1+2\alpha(\sigma_{xy}^{(i)})^3/\gamma_{xy}^{(i)}}{1+3\alpha G_{xy}(\sigma_{xy}^{(i)})^2}G_{xy}\gamma_{xy}^{(i+1)}.$$

Finally, this relation is written in terms of the damage parameter d:

$$\sigma_{xy}^{(i+1)} = (1-d)G_{xy}\gamma_{xy}^{(i+1)},$$

where

$$d = \frac{3\alpha G_{xy}(\sigma_{xy}^{(i)})^2 - 2\alpha(\sigma_{xy}^{(i)})^3/\gamma_{xy}^{(i)}}{1+3\alpha G_{xy}(\sigma_{xy}^{(i)})^2}.$$

This relation is implemented in user subroutines **USDFLD** and **VUSDFLD**, and the value of the damage parameter is assigned directly to the third field variable used for definition of the elastic properties.

The failure indices are calculated with the expressions discussed earlier, based on the stresses at the start of the increment:

$$e_m^2 = \left(\frac{\sigma_y^{(i)}}{Y_t}\right)^2 + \frac{2(\sigma_{xy}^{(i)})^2/G_{xy} + 3\alpha(\sigma_{xy}^{(i)})^4}{2S_c^2/G_{xy} + 3\alpha S_c^4} \quad \text{if} \quad \sigma_y^{(i)} > 0,$$

$$e_m^2 = \left(\frac{\sigma_y^{(i)}}{Y_c}\right)^2 + \frac{2(\sigma_{xy}^{(i)})^2/G_{xy} + 3\alpha(\sigma_{xy}^{(i)})^4}{2S_c^2/G_{xy} + 3\alpha S_c^4} \quad \text{if} \quad \sigma_y^{(i)} < 0,$$

$$e_{fs}^2 = \left(\frac{\sigma_x^{(i)}}{X_c}\right)^2 + \frac{2(\sigma_{xy}^{(i)})^2/G_{xy} + 3\alpha(\sigma_{xy}^{(i)})^4}{2S_c^2/G_{xy} + 3\alpha S_c^4}.$$

The values of the failure indices are not assigned directly to the field variables: instead, they are stored as solution-dependent state variables. Only if the value of a failure index exceeds 1.0 is the corresponding user-defined field variable set equal to 1.0. After the failure index has exceeded 1.0, the associated user-defined field variable continues to have the value 1.0 even though the stresses may reduce significantly, which ensures that the material does not "heal" after it has become damaged.

Finite element model

The plate consists of 24 plies of T300/976 graphite-epoxy in a $[(-45/+45)_6]_s$ layup. Instead of modeling each ply individually, we combine all plies in the −45° direction and all plies in the +45° direction. Consequently, only two layers need to be modeled separately:

1. A layer in the −45° direction with a thickness of 1.715 mm (0.0675 in).
2. A layer in the +45° direction with a thickness of 1.715 mm (0.0675 in).

The corresponding finite element model consists of two layers of CPS4 plane stress elements, with thicknesses and properties as previously discussed. The quarter-symmetry finite element model is shown in Figure 1.1.14–1.

The implementation of nonlinear material behavior with user-defined field variables is *explicit*: the nonlinearity is based on the state at the start of the increment. Hence, in Abaqus/Standard analyses the user must ensure that the time increments are sufficiently small, which is particularly important because the automatic time increment control in Abaqus/Standard is ineffective with the explicit nonlinearity implemented in **USDFLD**. If automatic time incrementation is used, the maximum time increment can be controlled from within subroutine **USDFLD** with the variable **PNEWDT**. This capability is useful if there are other nonlinearities that require automatic time incrementation. In this example the only significant nonlinearity is the result of the material behavior. Hence, fixed time incrementation can be used effectively. In Abaqus/Explicit analyses the stable time increment is usually sufficiently small to ensure good accuracy.

Results and discussion

For this problem experimental load-displacement results were obtained by Chang and Lessard. The experimental results, together with the numerical results obtained with Abaqus/Standard, are shown in Figure 1.1.14–2. The agreement between the experimental and numerical results is excellent up to the point where the load maximum is reached. After that, the numerical load-displacement curve drops off sharply, whereas the experimental data indicate that the load remains more or less constant. Chang and Lessard also show numerical results: their results agree with the results obtained by Abaqus but do not extend to the region where the load drops off. The dominant failure mode in this plate is fiber/matrix shear: failure occurs first at a load of approximately 12.15 kN (2700 lbs) and continues to grow in a stable manner until a load of approximately 13.5 kN (3000 lbs) is reached. Figure 1.1.14–3 shows the extent of the damage in the Abaqus/Standard finite element model at the point of maximum load. In this figure an element is shaded if fiber/matrix shear failure has occurred at at least three integration points. These results also show excellent agreement with the results obtained by Chang and Lessard.

As discussed earlier, the sharp load drop-off in the numerical results is the result of the lack of residual stress carrying capacity after the failure criterion is exceeded. Better agreement could be reached only if postfailure material data were available. Without postfailure data the results are very sensitive to the mesh and element type, which is clearly demonstrated by changing the element type from CPS4 (full integration) to CPS4R (reduced integration). The results are virtually identical up to the point where first failure occurs. After that point the damage in the CPS4R model spreads more rapidly than in the CPS4 model until a maximum load of about 12.6 kN (2800 lbs) is reached. The load then drops off rapidly.

The problem is also analyzed with Abaqus/Standard models consisting of S4R and S4 elements. The elements have a composite section with two layers, with each layer thickness equal to the thickness of the plane stress elements in the CPS4 and CPS4R models. The results that were obtained with the S4R and S4 element models are indistinguishable from those obtained with the CPS4R element model.

The numerical results obtained with Abaqus/Explicit using the CPS4R element model (not shown) are consistent with those obtained with Abaqus/Standard.

Input files

Abaqus/Standard input files

damagefailcomplate_cps4.inp	CPS4 elements.
damagefailcomplate_cps4.f	User subroutine **USDFLD** used in damagefailcomplate_cps4.inp.
damagefailcomplate_node.inp	Node definitions.
damagefailcomplate_element.inp	Element definitions.
damagefailcomplate_cps4r.inp	CPS4R elements.
damagefailcomplate_cps4r.f	User subroutine **USDFLD** used in damagefailcomplate_cps4r.inp.
damagefailcomplate_s4.inp	S4 elements.
damagefailcomplate_s4.f	User subroutine **USDFLD** used in damagefailcomplate_s4.inp.

damagefailcomplate_s4r.inp	S4R elements.
damagefailcomplate_s4r.f	User subroutine **USDFLD** used in damagefailcomplate_s4r.inp.

Abaqus/Explicit input files

damagefailcomplate_cps4r_xpl.inp	CPS4R elements.
damagefailcomplate_cps4r_xpl.f	User subroutine **VUSDFLD** used in damagefailcomplate_cps4r_xpl.inp.
damagefailcomplate_node.inp	Node definitions.
damagefailcomplate_element.inp	Element definitions.

Reference

- Chang, F-K., and L. B. Lessard, "Damage Tolerance of Laminated Composites Containing an Open Hole and Subjected to Compressive Loadings: Part I—Analysis," Journal of Composite Materials, vol. 25, pp. 2–43, 1991.

Table 1.1.14–1 Dependence of the elastic material properties on the field variables.

Material State	Elastic Properties				FV1	FV2	FV3
No failure	E_x	E_y	ν_{xy}	G_{xy}	0	0	0
Matrix failure	E_x	0	0	G_{xy}	1	0	0
Fiber/matrix shear	E_x	E_y	0	0	0	1	0
Shear damage	E_x	E_y	ν_{xy}	0	0	0	1
Matrix failure and fiber/matrix shear	E_x	0	0	0	1	1	0
Matrix failure and shear damage	E_x	0	0	0	1	0	1
Fiber/matrix shear and shear damage	E_x	E_y	0	0	0	1	1
All failure modes	E_x	0	0	0	1	1	1

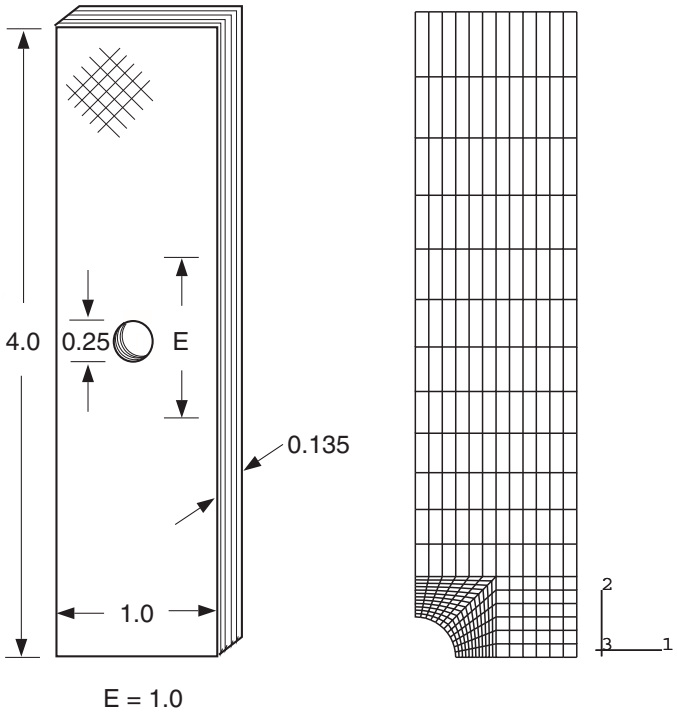


Figure 1.1.14–1 Plate geometry.

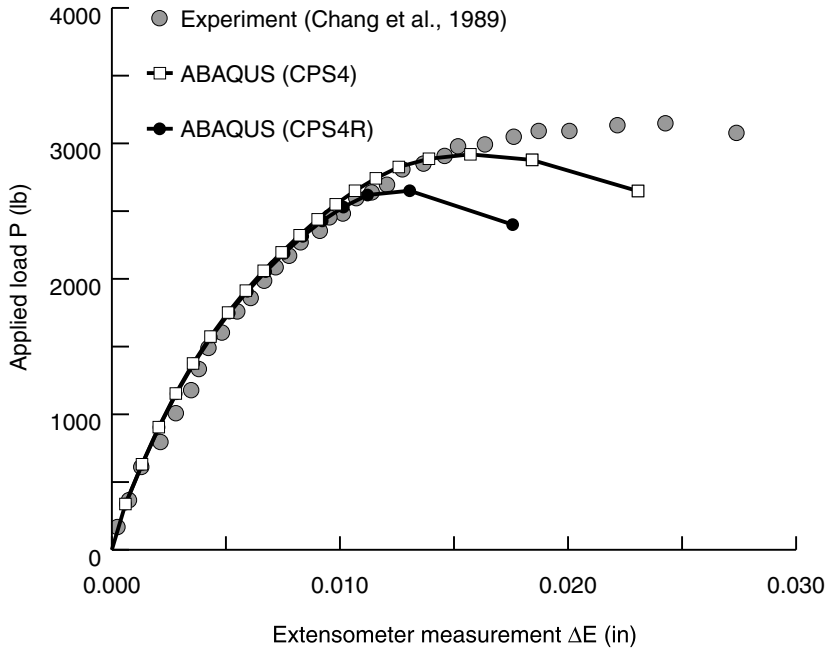


Figure 1.1.14–2 Experimental and numerical (Abaqus/Standard) load displacement curves.

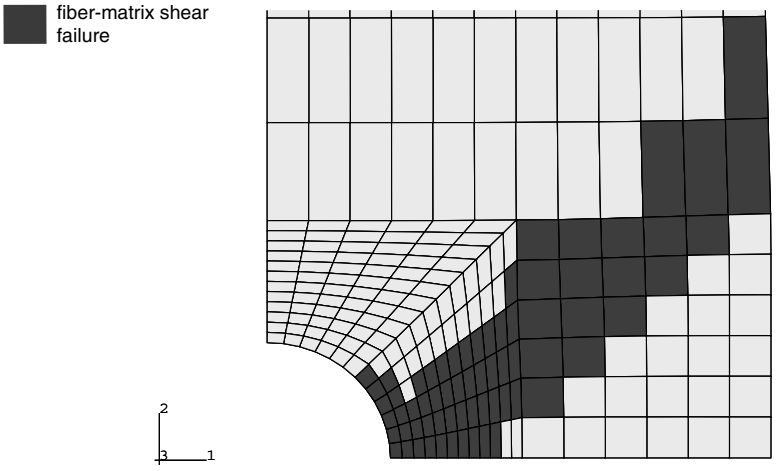


Figure 1.1.14–3 Distribution of material damage at maximum load obtained with Abaqus/Standard.

1.1.15 ANALYSIS OF AN AUTOMOTIVE BOOT SEAL

Product: Abaqus/Standard

Boot seals are used to protect constant velocity joints and steering mechanisms in automobiles. These flexible components must accommodate the motions associated with angulation of the steering mechanism. Some regions of the boot seal are always in contact with an internal metal shaft, while other areas come into contact with the metal shaft during angulation. In addition, the boot seal may also come into contact with itself, both internally and externally. The contacting regions affect the performance and longevity of the boot seal.

In this example the deformation of the boot seal, caused by a typical angular movement of the shaft, is studied. It provides a demonstration and verification of the finite-sliding capability in three-dimensional deformable-to-deformable contact and self-contact in Abaqus. This problem also demonstrates how to model a hyperelastic material using the **`UMAT`** user subroutine.

Geometry and model

The boot seal with the internal shaft is shown in Figure 1.1.15–1. The corrugated shape of the boot seal tightly grips the steering shaft at one end, while the other end is fixed. The rubber seal is modeled with first-order, hybrid brick elements with two elements through the thickness using symmetric model generation. The seal has a nonuniform thickness varying from a minimum of 3.0 mm to a maximum of 4.75 mm at the fixed end. The internal shaft is considered to be rigid and is modeled as an analytical rigid surface; the radius of the shaft is 14 mm. The rigid body reference node is located precisely in the center of the constant velocity joint.

The rubber is modeled as a slightly compressible neo-Hookean (hyperelastic) material with C_{10}=0.752 MPa and D_1=0.026 MPa^{-1}. For illustrative purposes an input file using the Marlow model is included; the model is defined using uniaxial test data generated by running a uniaxial test with the neo-Hookean model.

Contact is specified between the rigid shaft and the inner surface of the seal. Self-contact is specified on the inner and outer surfaces of the seal.

Loading

The mounting of the boot seal and the angulation of the shaft are carried out in a three-step analysis. The inner radius at the neck of the boot seal is smaller than the radius of the shaft so as to provide a tight fit between the seal and the shaft. In the first step the initial interference fit is resolved, corresponding to the assembly process of mounting the boot seal onto the shaft. The automatic “shrink” fit method is utilized. The second step simulates the angulation of the shaft by specifying a finite rotation of 20° at the rigid body reference node of the shaft. During the third step the angulated shaft travels around the entire circumference to demonstrate the robustness of the algorithm.

User subroutine for neo-Hookean hyperelasticity

In Abaqus/Standard user subroutine **UHYPER** is used to define a hyperelastic material. However, in this problem we illustrate the use of user subroutine **UMAT** as an alternative method of defining a hyperelastic material. In particular, we consider the neo-Hookean hyperelastic material model. The form of the neo-Hookean strain energy density function is given by

$$U(\overline{I}_1, \overline{I}_2, J) = C_{10}(\overline{I}_1 - 3) + \frac{1}{D_1}(J-1)^2.$$

Here, $\overline{I}_1$, $\overline{I}_2$, and J are the strain invariants of the deviatoric left Cauchy-Green deformation tensor $\overline{\mathbf{B}}$. This tensor is defined as $\overline{\mathbf{B}} \stackrel{\text{def}}{=} \overline{\mathbf{F}} \cdot \overline{\mathbf{F}}^T$, where $\overline{\mathbf{F}} = J^{-\frac{1}{3}}\mathbf{F}$ is the distortion gradient. "Hyperelastic material behavior," Section 4.6.1 of the Abaqus Theory Guide, contains detailed explanations of these quantities.

The constitutive equation for a neo-Hookean material is

$$\boldsymbol{\sigma} = \frac{2}{J}C_{10}\left(\overline{\mathbf{B}} - \frac{1}{3}\text{trace}(\overline{\mathbf{B}})\mathbf{I}\right) + \frac{2}{D_1}(J-1)\mathbf{I},$$

where $\boldsymbol{\sigma}$ is the Cauchy stress. The material Jacobian, $\mathbf{C}$, is defined by the variation of the Kirchhoff stress

$$\delta(J\boldsymbol{\sigma}) = J\mathbf{C} \cdot \delta\mathbf{D},$$

where $\delta\mathbf{D}$ is the virtual rate of deformation and is defined as

$$\delta\mathbf{D} \stackrel{\text{def}}{=} \text{sym}(\delta\mathbf{F} \cdot \mathbf{F}^{-1}).$$

For a neo-Hookean material the components of $\mathbf{C}$ are given by

$$C_{ijkl} = \frac{2}{J}C_{10}\left(\frac{1}{2}(\delta_{ik}\overline{B}_{jl} + \overline{B}_{ik}\delta_{jl} + \delta_{il}\overline{B}_{jk} + \overline{B}_{il}\delta_{jk}) - \frac{2}{3}\delta_{ij}\overline{B}_{kl} - \frac{2}{3}\overline{B}_{ij}\delta_{kl} + \frac{2}{9}\delta_{ij}\delta_{kl}\overline{B}_{mm}\right) + \frac{2}{D_1}(2J-1)\delta_{ij}\delta_{kl}.$$

Results and discussion

Figure 1.1.15–2 shows the deformed configuration of the model. The rotation of the shaft causes the stretching of one side and compression on the other side of the boot seal. The surfaces have come into self-contact on the compressed side. Figure 1.1.15–3 shows the contours of maximum principal stresses in the boot seal.

Comparison of the analysis times when using fixed and automated contact patches shows that both analyses complete in approximately the same amount of time. This can be expected for this type of problem since the fixed contact patches are limited in size to a few elements. For the case with fixed

contact patches the wavefront is somewhat larger, requiring more memory and solution time per iteration. However, this is offset by the time required to form new contact patches and to reorder the equations for the case with automatic contact patches. The results obtained with the model that uses user subroutine **UMAT** are identical to those obtained using the built-in Abaqus material model.

Input files

bootseal.inp	Analysis with node-to-surface contact.
bootseal_surf.inp	Analysis with surface-to-surface contact.
bootseal_2d.inp	Two-dimensional model for symmetric model generation in bootseal.inp.
bootseal_2d_surf.inp	Two-dimensional model for symmetric model generation in bootseal.inp using surface-to-surface contact.
bootseal_umat.inp	Analysis with user subroutine **UMAT**.
bootseal_2d_umat.inp	Two-dimensional model for symmetric model generation in bootseal_umat.inp.
bootseal_umat.f	**UMAT** for the neo-Hookean hyperelasticity model.
bootseal_marlow.inp	Analysis with Marlow hyperelasticity model.
bootseal_2d_marlow.inp	Two-dimensional model for symmetric model generation in bootseal_marlow.inp.

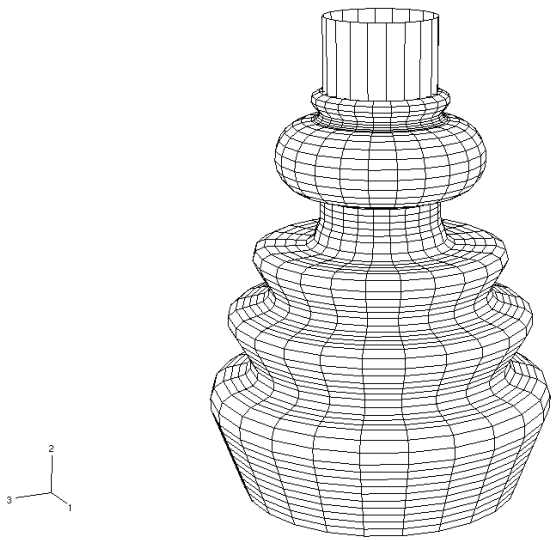


Figure 1.1.15–1 Undeformed model.

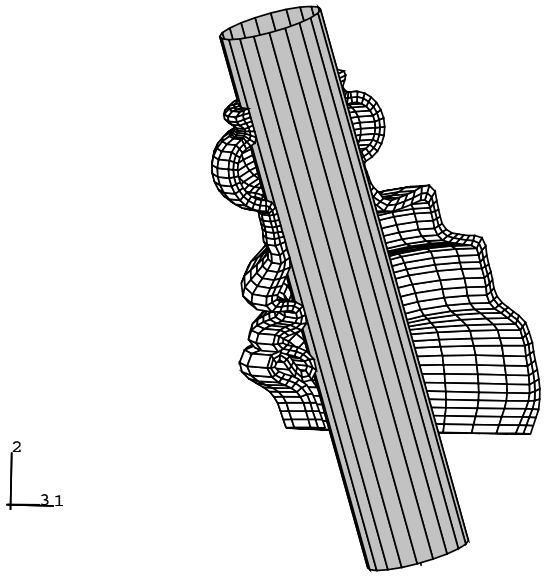


Figure 1.1.15–2 Deformed configuration of half the model.

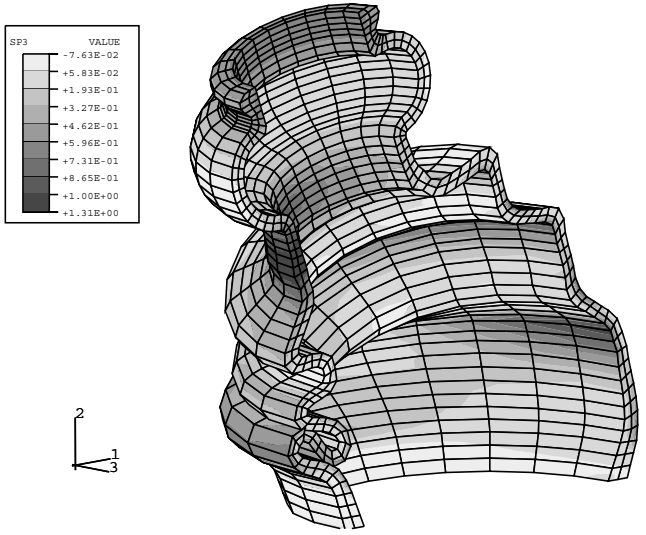


Figure 1.1.15–3 Contours of maximum principal stress in the seal.

1.1.16 PRESSURE PENETRATION ANALYSIS OF AN AIR DUCT KISS SEAL

Product: Abaqus/Standard

Seals are common structural components that often require design analyses. Abaqus can be used to perform nonlinear finite element analyses of seals and provide information needed to determine the seal performance. Information such as a load-deflection curve, seal deformation and stresses, and contact pressure distribution is readily obtained in these analyses. Abaqus allows for pressure penetration effects between the seal and the contacting surfaces to be considered in these analyses, making routine analyses of seals more realistic and accurate. Analyses of clutch seals, threaded connectors, car door seals, and air duct kiss seals are some applications where pressure penetration effects are important.

The surface-based pressure penetration capability is used to simulate pressure penetration between contacting surfaces. It is invoked by using the pressure penetration option, which is described in "Pressure penetration loading," Section 37.1.7 of the Abaqus Analysis User's Guide. This capability is provided for simulating cases where a joint between two deforming bodies (for example, between two components threaded onto each other) or between a deforming body and a rigid surface (such as a soft gasket used in a joint) is exposed at one or multiple ends to a fluid or air pressure. This air pressure will penetrate into the joint and load the surfaces forming the joint until some area of the surfaces is reached where the contact pressure between the abutting surfaces exceeds the critical value specified on the pressure penetration option, cutting off further penetration.

Geometry and model

The major consideration in an air duct kiss seal design is to provide sealing while avoiding excessive closure force. A poorly designed air duct seal that minimizes the amount of effort to close the fan cowl door may fail to prevent leakage and reduce wind noise. The model used in this example is a simplified version of an air duct kiss seal. It illustrates how pressure penetration effects can be modeled using Abaqus.

The seal modeled is a rolled shape seal. An axisymmetric model of the seal is developed first, as shown in Figure 1.1.16–1. A three-dimensional model of the seal is also developed with only a 5-degree fraction of the seal discretized, as shown in Figure 1.1.16–2. The top horizontal rigid surface represents the air fan cowl door, and the bottom horizontal rigid surface represents the seal groove. The rolled seal is 2.54 mm (0.1 in) thick and 74.66 mm (2.9 in) high; and its inner diameters at the top and bottom surfaces are 508.5 mm (20 in) and 528.3 mm (20.8 in), respectively. A folded metal clip is partially bonded to the top surface of the seal. The thickness of the metal clip is 0.48 mm (0.019 in).

The material of the seal is taken to be an incompressible rubberlike material. To obtain the material constants, the Ogden form of the strain energy function with $N = 4$ is used to fit the uniaxial test data. The metal clip is made of steel, with a Young's modulus of 206.8 GPa (3.0×10^7 lb/in^2) and a Poisson's ratio of 0.3. CAX4H elements are used to model the seal and the metal clip in the axisymmetric model, and C3D8H elements are used in the three-dimensional model. The contact pair approach is used to model the contact between the top surface of the metal clip and the top rigid surface representing the fan cowl door, where the pressure penetration is likely to occur. The contact pair approach is also used

to model the contact between the seal and the bottom rigid surface, the contact between the seal and the unbonded portion of the metal clip, and the self-contact of the seal. The mechanical interaction between the contact surfaces is assumed to be frictional contact. Therefore, the friction option is used to specify friction coefficients. To increase computational efficiency, the slip tolerance (the ratio of allowable maximum elastic slip to characteristic contact surface face dimension) on the friction option is specified for the contact surfaces between the seal and the metal clip because the dimensions of these elements vary greatly. Fixed boundary conditions are applied initially to the reference node of the top rigid surface, 5001, and the reference node of the bottom rigid surface, 5002. The vertical edge at the bottom of the seal is constrained such that it cannot be moved in the 1-direction. The bottom node of the vertical edge, 1, touches the bottom rigid surface and is held fixed in the 2-direction. The top rigid surface is located initially 1.27 mm (0.05 in) above the top surface of the metal clip.

The seal and the unbonded portion of the clip are loaded by air pressure on all of their inner surfaces and by contact pressure generated by closing the air fan cowl door. Two nonlinear static steps, all of which include large-displacement effects, are used to simulate these loading conditions.

In the first step the top rigid surface moves 35.56 mm (1.4 in) downward in the y-direction, simulating the closing of the fan cowl door.

In the second step the inner surface of the seal is subjected to a uniform air pressure load of 206.8 KPa (30.0 lb/in^2) since some gaps between the seal and the top rigid surface have been closed. The pressure penetration is simulated between the top surface of the metal clip (**PPRES**), which includes 31 elements, and the top rigid surface (**CFACE**). Air pressure penetration does not need to be modeled between the metal clip and the seal because they are well bonded.

The pressure penetration option is invoked to define the node exposed to the air pressure, the magnitude of the air pressure, and the critical contact pressure. The surface **PPRES** is exposed to the air pressure at node 597, with a pressure magnitude of 206.8 KPa (30.0 lb/in^2). A default value of zero for the critical contact pressure is used, indicating that the pressure penetration occurs only when contact at a slave node is lost.

Results and discussion

The deformed configuration and the contours of the contact pressures on the seal at the end of Step 1 are shown in Figure 1.1.16–3 and Figure 1.1.16–4 for the axisymmetric model and in Figure 1.1.16–5 for the three-dimensional model. A nonuniform contact pressure is observed along the surface of the seal. The contact pressure at the first five slave nodes is zero.

The penetrating pressure loads are applied during Step 2. The air pressure is applied immediately to elements associated with the first five slave nodes since the contact pressure there is zero and the pressure penetration criterion is satisfied. For the axisymmetric model, the spread of the penetration is captured in Figure 1.1.16–6 through Figure 1.1.16–14, which show the deformed seal, the contact pressure profile, and the air pressure profiles corresponding to load increments 2, 10, and 16 of Step 2. The pressures applied to the surface corresponding to these three increments are 1.296 KPa (0.188 lb/in^2), 13.96 KPa (2.03 lb/in^2), and 70.88 KPa (10.28 lb/in^2), respectively. For the three-dimensional model, the spread of the penetration is captured in Figure 1.1.16–15 through Figure 1.1.16–17, which show the contact pressure profiles corresponding to load increments 2, 6, and 14 of Step 2.

Increased penetrating pressure loads applied in Step 2 further reduce the contact pressure, eventually causing complete air penetration through the seal. The seal was lifted off from the air fan cowl door except at the last slave node, 663, where the contact pressure is well maintained due to imposed boundary conditions and the air pressures. For the axisymmetric model, the development of the weakening of the sealing is captured in Figure 1.1.16–18 through Figure 1.1.16–21, which show the deformed seal and the contact pressure profile corresponding to load increment 20 and at the end of Step 2. The pressures applied to the surface corresponding to these two increments are 112.3 KPa (16.28 lb/in^2) and 206.8 KPa (30.0 lb/in^2), respectively. For the three-dimensional model, the development of the weakening of the sealing is captured in Figure 1.1.16–22 through Figure 1.1.16–23, which show the contact pressure profile corresponding to load increment 19 and at the end of Step 2.

The behavior of the seal throughout the loading histories can be best described by plotting the air penetration distance as a function of the air pressure, as shown in Figure 1.1.16–24 for both axisymmetric and three-dimensional models. It is clear that air penetration into the seal accelerates only when the pressure is on the order of 51.7 KPa (7.5 lb/in^2). The air completely penetrates through the seal when the pressure is 82.7 KPa (12.0 lb/in^2), which is approximately equal to 80% of the sea level atmospheric pressure.

In addition, the same model is analyzed with the adaptive automatic stabilization scheme, which improves the robustness by automatically adjusting the damping factor based on the convergence history while having very little effect on the results. The dissipated stabilization energy is found to be small when the adaptive stabilization scheme is used.

Input files

presspenairductseal.inp	Pressure penetration simulation of an air duct kiss seal.
presspenairductseal_stabil_adap.inp	Same as presspenairductseal.inp with adaptive automatic stabilization.
presspenairductseal_node.inp	Node definitions for the seal model.
presspenairductseal_elem_metal.inp	Element definitions for the metal part of the seal model.
presspenairductseal_elem_rub.inp	Element definitions for the rubber part of the seal model.
presspenairductseal_c3d8h.inp	Pressure penetration simulation of an air duct kiss seal in three dimensions.

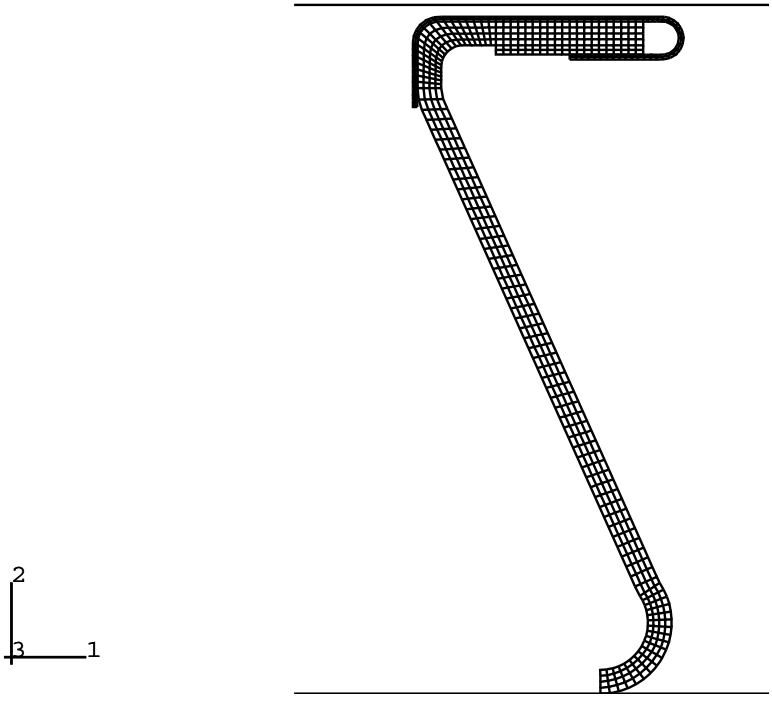


Figure 1.1.16–1 Axisymmetric model of air duct kiss seal.

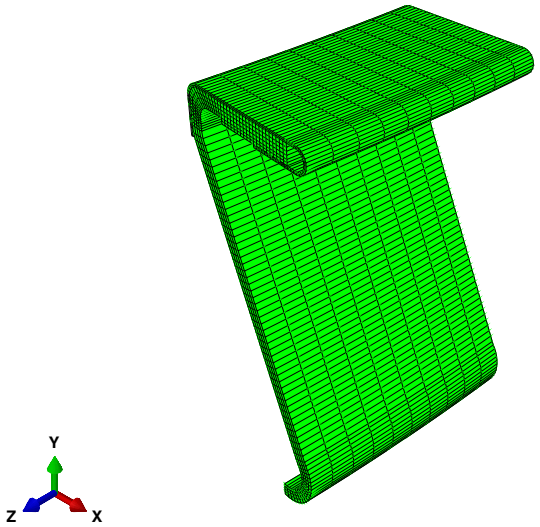


Figure 1.1.16–2 Three-dimensional model of air duct kiss seal.

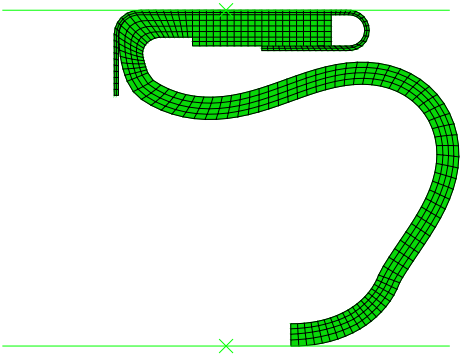


Figure 1.1.16–3 For the axisymmetric model, deformed configuration of the seal at the end of Step 1.

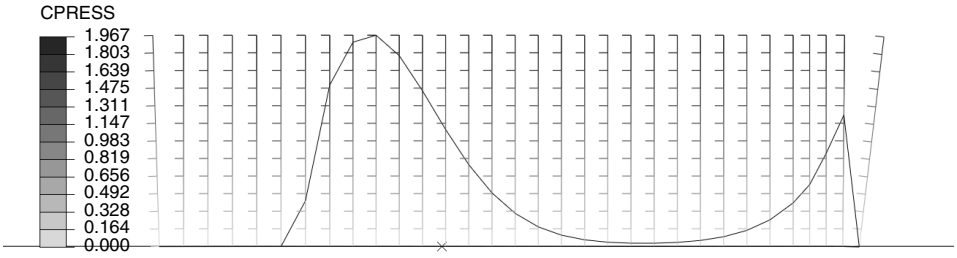


Figure 1.1.16–4 For the axisymmetric model, contact stress contours in the seal at the end of Step 1.

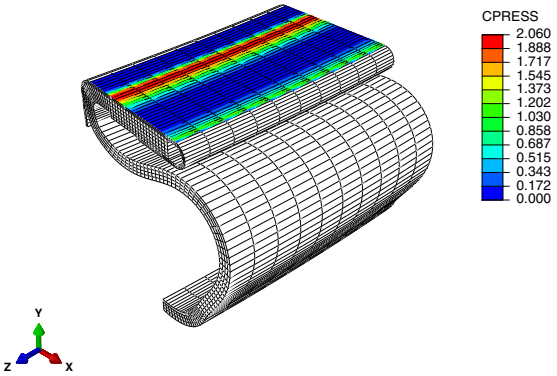


Figure 1.1.16–5 For the three-dimensional model, deformed configuration and contact stress contours of the seal at the end of Step 1.

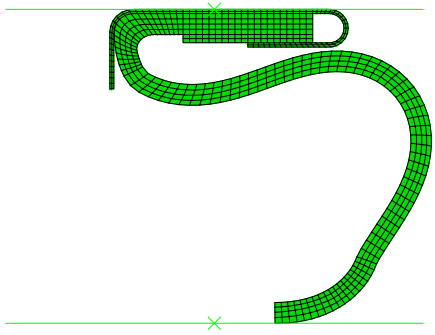


Figure 1.1.16–6 For the axisymmetric model, deformed configuration of the seal at Step 2, increment 2.

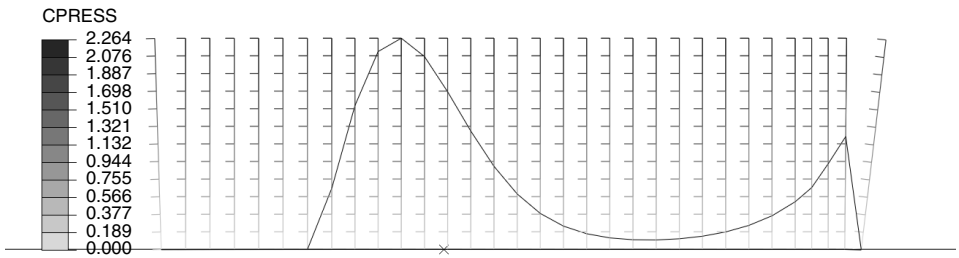


Figure 1.1.16–7 For the axisymmetric model, contact stress contours in the seal at Step 2, increment 2.

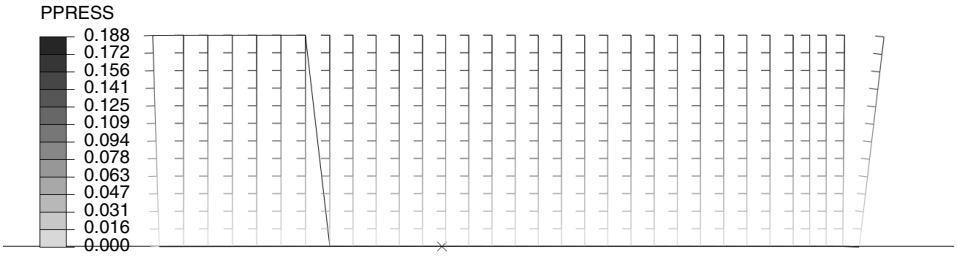


Figure 1.1.16–8 For the axisymmetric model, air pressure contours in the seal at Step 2, increment 2.

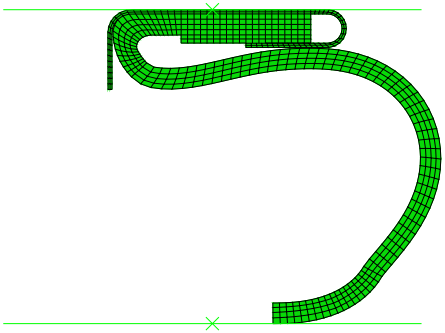


Figure 1.1.16–9 For the axisymmetric model, deformed configuration of the seal at Step 2, increment 10.

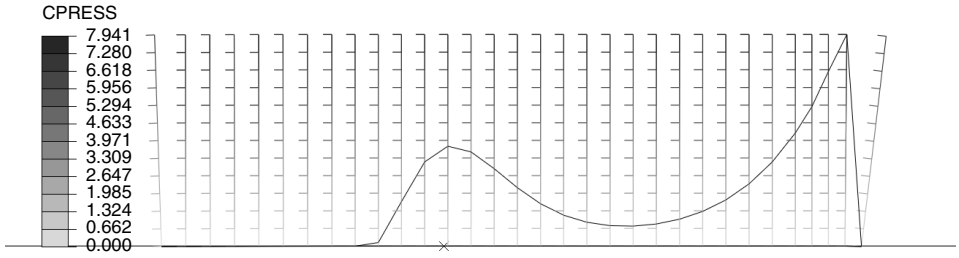


Figure 1.1.16–10 For the axisymmetric model, contact stress contours in the seal at Step 2, increment 10.

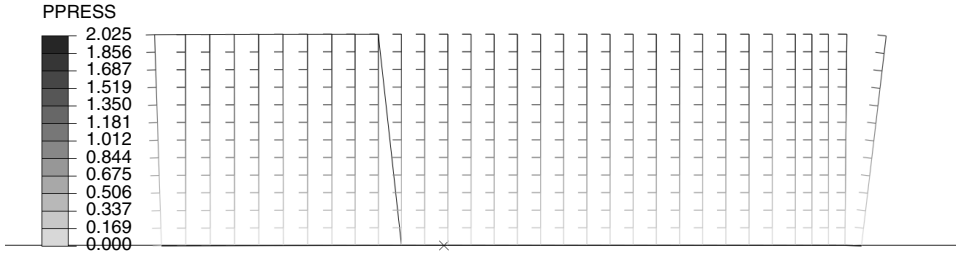


Figure 1.1.16–11 For the axisymmetric model, air pressure contours in the seal at Step 2, increment 10.

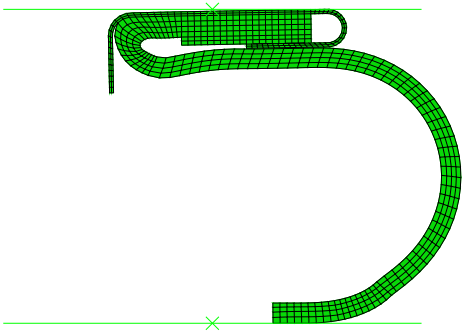


Figure 1.1.16–12 For the axisymmetric model, deformed configuration of the seal at Step 2, increment 16.

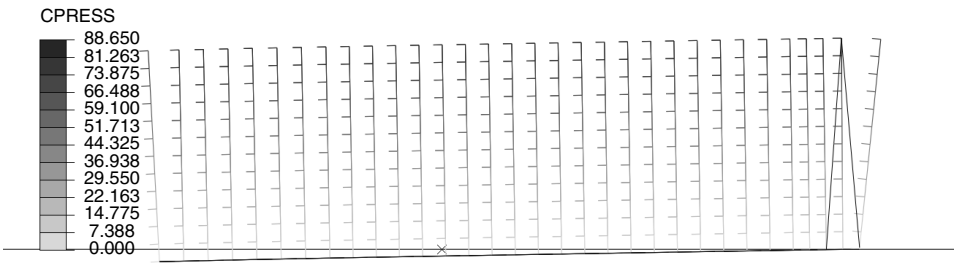


Figure 1.1.16–13 For the axisymmetric model, contact stress contours in the seal at Step 2, increment 16.

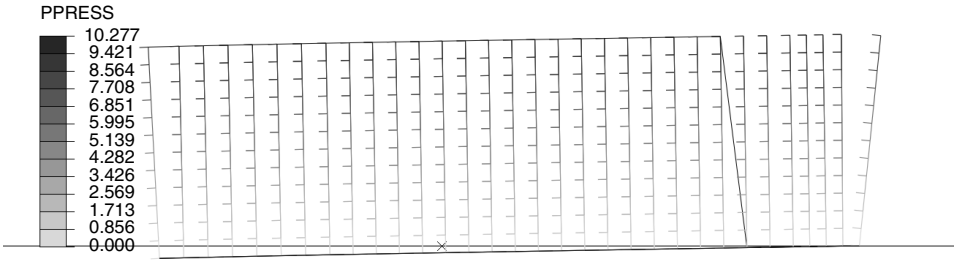


Figure 1.1.16–14 For the axisymmetric model, air pressure contours in the seal at Step 2, increment 16.

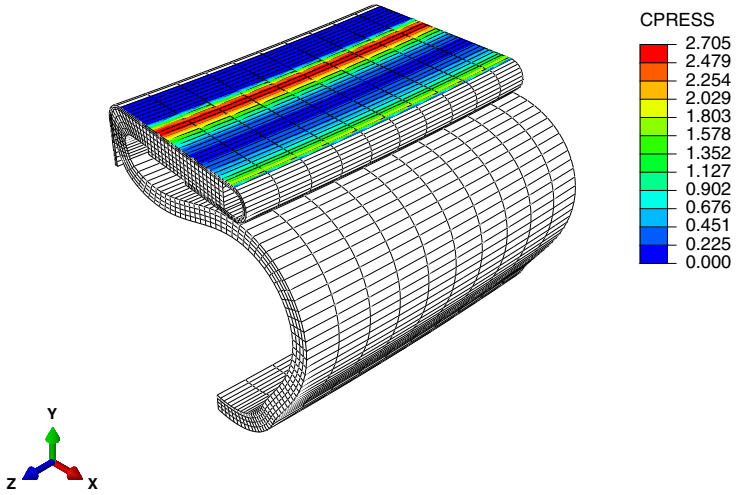


Figure 1.1.16–15 For the three-dimensional model, deformed configuration and contact stress contours of the seal at Step 2, increment 2.

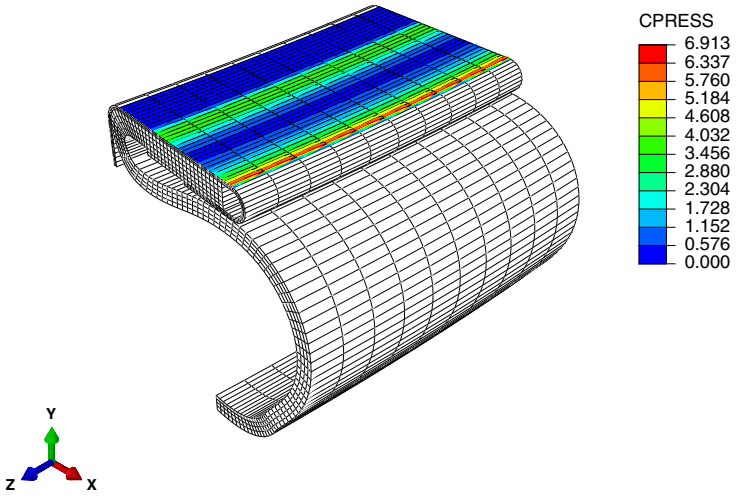


Figure 1.1.16–16 For the three-dimensional model, deformed configuration and contact stress contours of the seal at Step 2, increment 6.

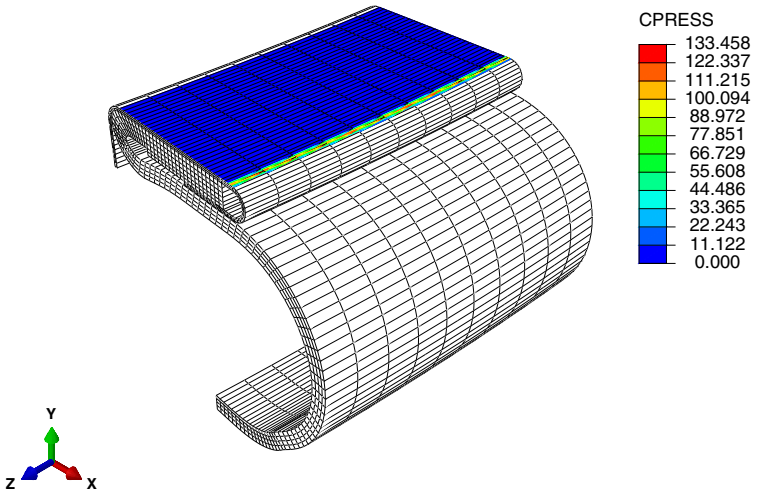


Figure 1.1.16–17 For the three-dimensional model, deformed configuration and contact stress contours of the seal at Step 2, increment 14.

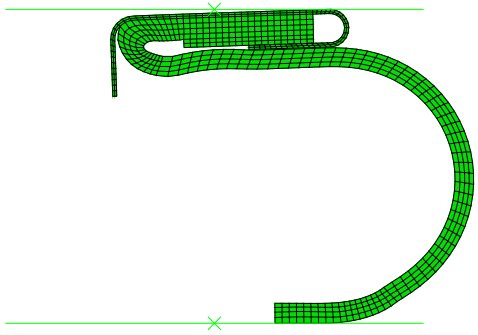


Figure 1.1.16–18 For the axisymmetric model, deformed configuration of the seal at Step 2, increment 20.

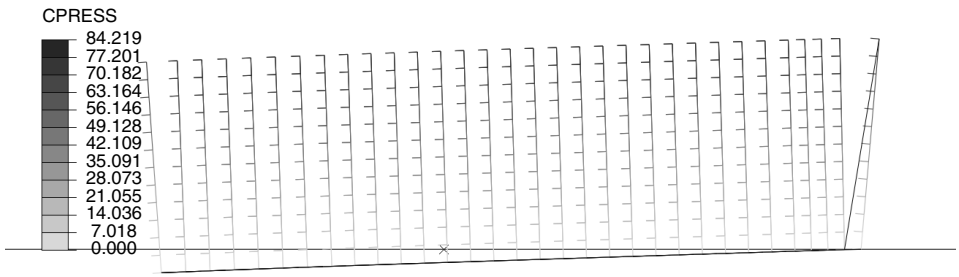


Figure 1.1.16–19 For the axisymmetric model, contact stress contours in the seal at Step 2, increment 20.

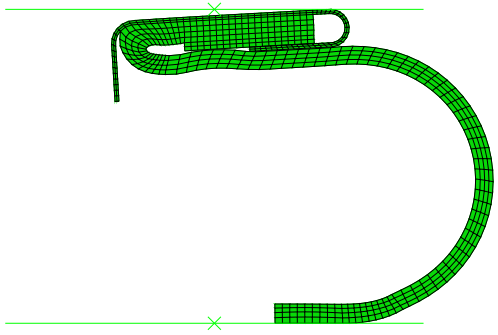


Figure 1.1.16–20 For the axisymmetric model, deformed configuration of the seal at the end of Step 2.

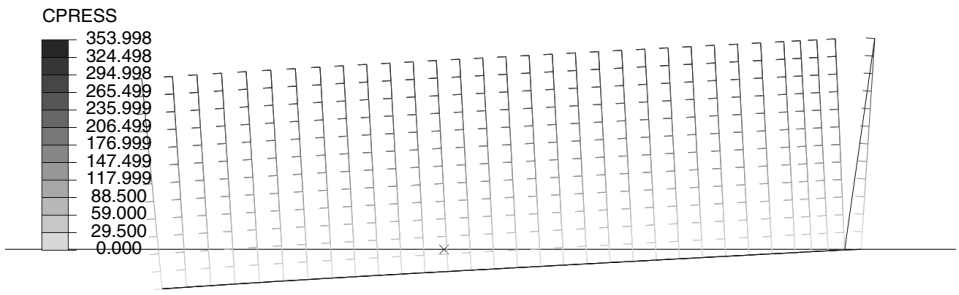


Figure 1.1.16–21 For the axisymmetric model, contact stress contours in the seal at the end of Step 2.

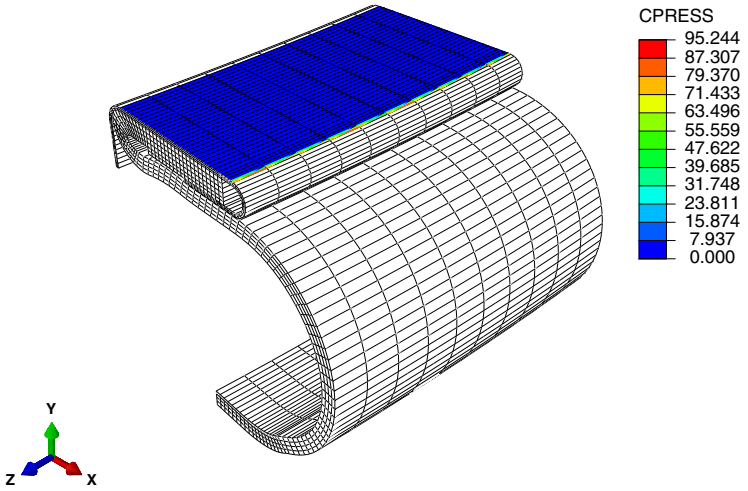


Figure 1.1.16–22 For the three-dimensional model, deformed configuration and contact stress contours of the seal at Step 2, increment 19.

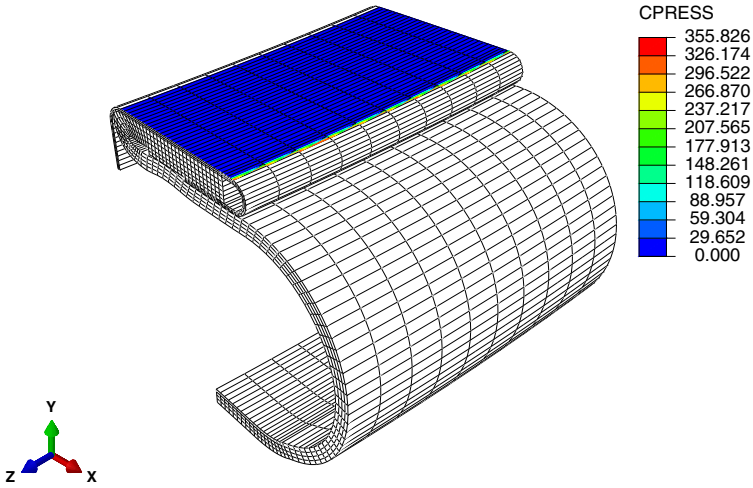


Figure 1.1.16–23 For the three-dimensional model, deformed configuration and contact stress contours of the seal at the end of Step 2.

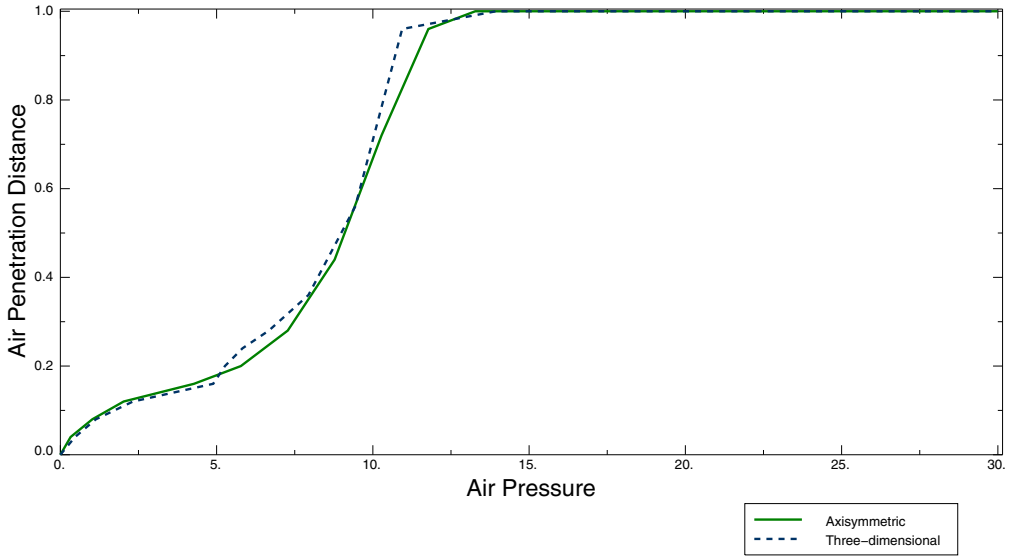


Figure 1.1.16–24 Air penetration distance as a function of air pressure in the seal

1.1.17 SELF-CONTACT IN RUBBER/FOAM COMPONENTS: JOUNCE BUMPER

Products: Abaqus/Standard Abaqus/Explicit

The self-contact capability in Abaqus is illustrated with two examples derived from the automotive component industry: this problem and the following one, which discusses a rubber gasket. These examples demonstrate the use of the single-surface contact capability available for large-sliding analysis in Abaqus. Components that deform and change their shape substantially can fold and have different parts of the surface come into contact with each other. In such cases it can be difficult to predict at the outset of the analysis where such contact may occur and, therefore, it can be difficult to define two independent surfaces to make up a contact pair.

A jounce bumper, sometimes referred to as a "helper spring," is a highly compressible component that is used as part of the shock isolation system in a vehicle. It is typically located above the coil spring that connects the wheels to the frame. Microcellular material is used because of its high compressibility and low Poisson's ratio value at all but fully compressed configurations.

The bumper is mounted on a mandrel with a diameter larger than the bumper's inner diameter (Figure 1.1.17–1). The first step of the analysis solves this interference fit problem. The bumper initially sits against a fixed flat rigid surface on one end; on the other end, another flat rigid surface is used to model the compression of the bumper. The geometry of the bumper is such that it folds in three different locations. Separate surfaces are defined at the locations where self-contact is expected. This modeling technique produces an economical analysis because the scope of contact searches is limited.

Geometry and model

The bumper is 76.0 mm (3.0 in) long and has an inside diameter of 20.0 mm (.8 in). The mandrel, which is modeled as a rigid surface, has a diameter of 22.0 mm (.9 in). The bumper is modeled with the hyperfoam material model. The compressible, nonlinear elastic behavior is described by a strain energy function. A mesh of either CAX3 or CAX4R elements is produced using an automatic mesh generator. Figure 1.1.17–1 and Figure 1.1.17–2 show the initial mesh with CAX3 elements and CAX4R elements, respectively. In addition to the portions of the bumper's surface used to define self-contact, additional regions are defined to model contact between the bumper and the fixed surface, the bumper and the mandrel, and the bumper and the moving rigid surface. A small amount of friction (a Coulomb coefficient of 0.05) is applied to all of the surfaces.

Results and discussion

The bumper analysis is a two-step process. In the first step the interference between the bumper's inner diameter and the mandrel is resolved. In the Abaqus/Standard analysis the automatic "shrink" fit method (see "Common difficulties associated with contact modeling in Abaqus/Standard," Section 39.1.2 of the Abaqus Analysis User's Guide) is used: the calculated initial penetration is allowed at the beginning of the step and scaled linearly to zero at the end of the step. In the Abaqus/Explicit analysis the interference resolution step is performed in one of two ways. In the first approach the mandrel is

positioned so that no contact or overclosure exists between the bumper and the mandrel at the outset of the analysis. The rigid surface representing the mandrel is then moved in the radial direction to simulate the compression of the bumper due to the interference fit. In the second approach the shrink fit solution from Abaqus/Standard is imported into Abaqus/Explicit. A comparison of the Mises stresses predicted by Abaqus/Standard and Abaqus/Explicit at the end of the interference step shows that the results are very similar (see Figure 1.1.17–3 and Figure 1.1.17–4).

In the second step the bottom surface compresses the bumper 42.0 mm (1.7 in) as a result of the application of displacement boundary conditions to the reference node of the surface. Figure 1.1.17–5, Figure 1.1.17–6, and Figure 1.1.17–7 show the final deformed shape of the bumper; the high compressibility of the material is apparent, as well as the folding of the surface onto itself. Although a general knowledge of where the folding would occur was used in the definition of the self-contacting surfaces, it is not necessary to know exactly where the kinks in the surface will form.

The final deformed shapes predicted by Abaqus/Standard and Abaqus/Explicit for the CAX3 element mesh are the same (see Figure 1.1.17–5 and Figure 1.1.17–6, respectively). A similar shape is predicted by Abaqus/Explicit when CAX4R elements are used (see Figure 1.1.17–7). However, the solution obtained with CAX4R elements in Abaqus/Explicit reveals that local buckling occurs in the upper-left folding radius. This makes a similar analysis using CAX4R elements in Abaqus/Standard very difficult. The local buckling is not captured in the CAX3 analysis due to the stiffer nature of these elements.

The load versus displacement curves of the bottom surface are shown in Figure 1.1.17–8. The results obtained with CAX3 elements in Abaqus/Standard and with CAX3 and CAX4R elements in Abaqus/Explicit are very similar. The energy absorption capacity of the bumper is seen through these curves.

Input files

selfcontact_bump_std_cax3.inp	Jounce bumper model for Abaqus/Standard using CAX3 elements.
selfcontact_bump_surf.inp	Jounce bumper model for Abaqus/Standard using CAX3 elements and surface-to-surface contact.
selfcontact_bump_std_imp_surf.inp	Jounce bumper model for Abaqus/Standard using CAX3 elements and surface-to-surface contact. This input file depends on selfcontact_bump_surf.inp.
selfcontact_bump_xpl_cax3.inp	Jounce bumper model for Abaqus/Explicit using CAX3 elements.
selfcontact_bump_xpl_cax4r.inp	Jounce bumper model for Abaqus/Explicit using CAX4R elements.
selfcontact_bump_std_resinter_cax4r.inp	Jounce bumper model for Abaqus/Standard using CAX4R elements to resolve the interference fit.
selfcontact_bump_std_imp_cax3.inp	Jounce bumper model for Abaqus/Standard using CAX3 elements; interference fit solution imported from Abaqus/Standard.

selfcontact_bump_xpl_imp_cax3.inp	Jounce bumper model for Abaqus/Explicit using CAX3 elements; interference fit solution imported from Abaqus/Standard.
selfcontact_bump_xpl_imp_cax4r.inp	Jounce bumper model for Abaqus/Explicit using CAX4R elements; interference fit solution imported from Abaqus/Standard.
selfcontact_bump_node_cax3.inp	Node definitions for the bumper model with CAX3 elements.
selfcontact_bump_node_cax4r.inp	Node definitions for the bumper model with CAX4R elements.
selfcontact_bump_element_cax3.inp	Element definitions for the bumper model with CAX3 elements.
selfcontact_bump_element_cax4r.inp	Element definitions for the bumper model with CAX4R elements.
selfcontact_bump_surfdef_cax3.inp	Surface definitions for the bumper model with CAX3 elements.
selfcontact_bump_surfdef_cax4r.inp	Surface definitions for the bumper model with CAX4R elements.

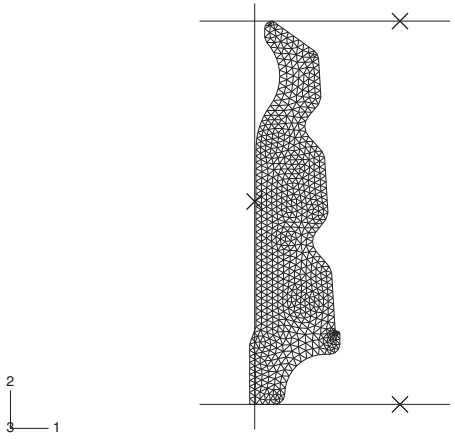


Figure 1.1.17–1 Jounce bumper initial mesh with CAX3 elements (Abaqus/Standard).

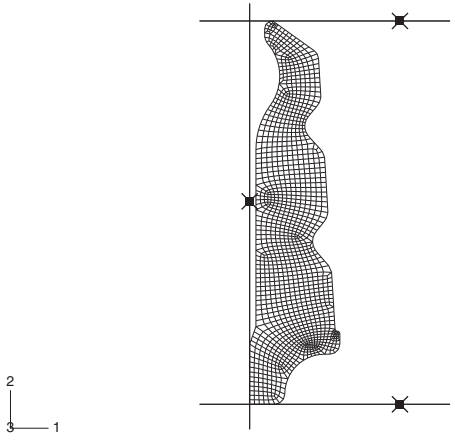


Figure 1.1.17–2 Jounce bumper initial mesh with CAX4R elements (Abaqus/Explicit).

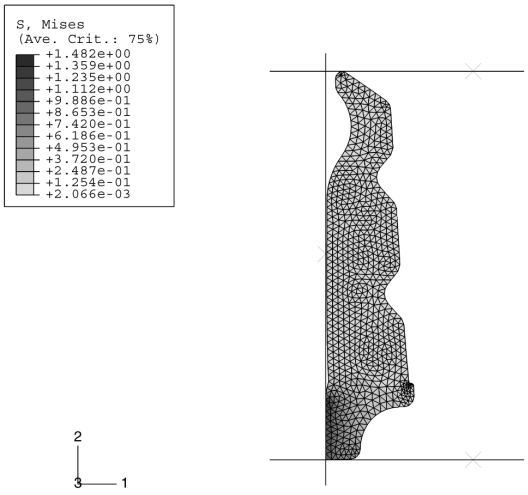


Figure 1.1.17–3 Mises stresses in bumper after interference is resolved with Abaqus/Standard automatic shrink fit option.

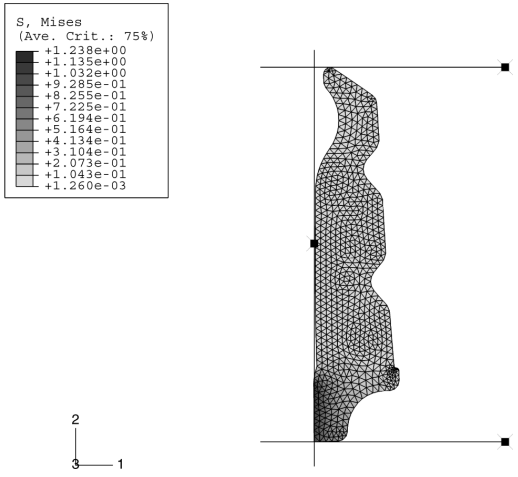


Figure 1.1.17–4 Mises stresses in bumper after interference is resolved with Abaqus/Explicit.

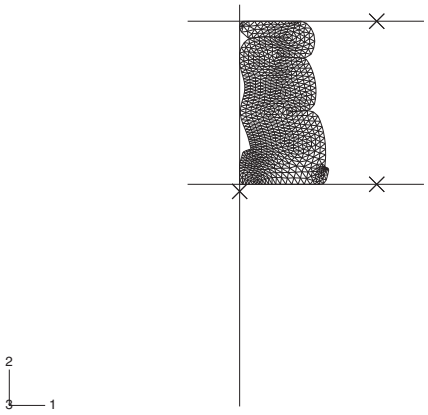


Figure 1.1.17–5 Bumper mesh after crushing (Abaqus/Standard; CAX3 elements).

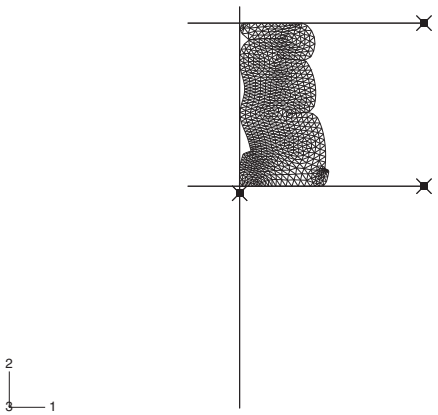


Figure 1.1.17–6 Bumper mesh after crushing (Abaqus/Explicit; CAX3 elements).

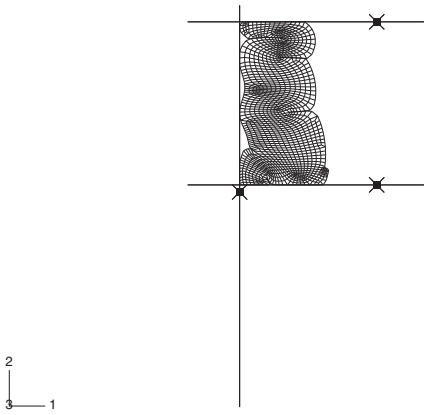


Figure 1.1.17–7 Bumper mesh after crushing (Abaqus/Explicit; CAX4R elements).

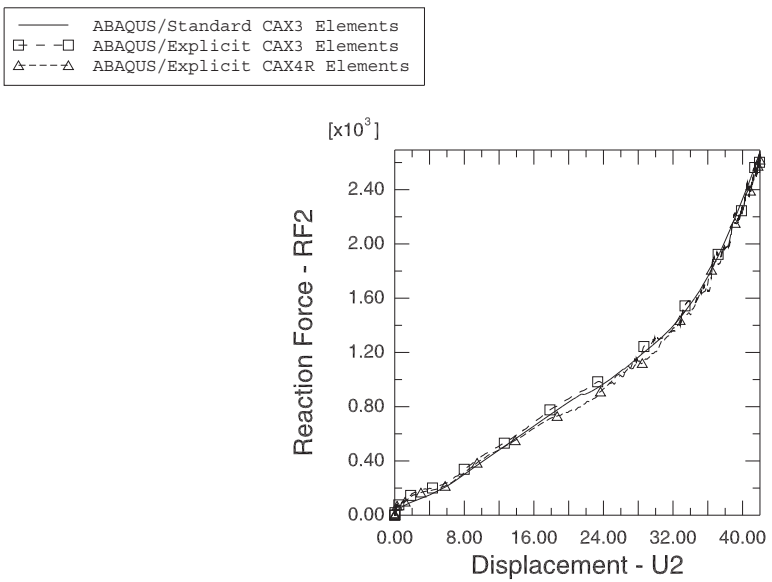


Figure 1.1.17–8 Bumper load-displacement curve.

1.1.18 SELF-CONTACT IN RUBBER/FOAM COMPONENTS: RUBBER GASKET

Products: Abaqus/Standard Abaqus/Explicit

The self-contact capability in Abaqus is illustrated with two examples derived from the automotive component industry: this problem and the preceding one, which discusses a jounce bumper. These examples demonstrate the use of the single-surface contact capability available for two-dimensional large-sliding analysis. Components that deform and change their shape substantially can fold and have different parts of the surface come into contact with each other. In such cases it can be difficult to predict at the outset of the analysis where such contact may occur and, therefore, it can be difficult to define two independent surfaces to make up a contact pair.

This model is used to analyze an oil pan gasket, which enhances the sealing of the oil pan against the engine block. A primary objective of gasket designers is to reach or exceed a threshold value of contact pressure at the gasket bead/cover/engine block interfaces. Experience shows that, above such a threshold, oil will not leak. Another item of interest is the load-deflection curve obtained when compressing the gasket cross-section since it is indicative of the bolt load required to attain a certain gap between the oil pan and the engine block. Finally, the analysis provides details to ensure that stresses and strains are within acceptable bounds.

The rubber gasket is embedded in a plastic backbone. It has two planes of symmetry and a bead that, when compressed, provides the sealing effect (Figure 1.1.18–1). A flat rigid surface, parallel to one of the symmetry planes, pushes the gasket into the backbone. The geometry of the gasket is such that it folds in two different locations. In this model the entire free surface of the gasket and of the backbone is declared as a single surface allowed to contact itself. This modeling technique, although very simple, is more expensive because of the extensive contact searches required, as well as a larger wavefront of the equation system when using Abaqus/Standard.

The analysis is performed using both Abaqus/Standard and Abaqus/Explicit.

Geometry and model

The rubber gasket is modeled as a quarter of a plane strain section, initially in contact with a flat rigid surface. The clearance between the plastic backbone and the surface is 0.612 mm (.024 in). The height of the bead in the gasket is 1.097 mm (.043 in). The backbone is modeled with a linear elastic material with a Young's modulus of 8000.0 MPa (1160 ksi) and a Poisson's ratio of 0.4. In Abaqus/Standard the gasket is modeled as a fully incompressible hyperelastic material, which is much softer than the backbone material at all strain levels. In Abaqus/Explicit a small amount of compressibility is assumed for the gasket material. The nonlinear elastic behavior of the gasket is described by a strain energy function that is a first-order polynomial in the strain invariants. The model is discretized with first-order quadrilaterals. Standard elements are used for the backbone. In Abaqus/Standard full-integration hybrid elements are used for the gasket, while reduced-integration elements are used to model the gasket in Abaqus/Explicit. The interface between the gasket and the backbone is assumed to be glued with no special treatment required. A single surface definition covers all of the free surface of the gasket and the backbone. Through the definition of contact pairs, this surface is allowed to contact both the rigid surface

and itself. A small amount of friction (Coulomb coefficient of 0.05) is applied to the interface with the rigid surface, which is assumed to be lubricated. Sticking surface behavior, through the specification of rough friction ("Frictional behavior," Section 37.1.5 of the Abaqus Analysis User's Guide), is applied when the gasket contacts itself, denoting a clean surface.

Results and discussion

The gasket analysis is a single-step procedure in which the rigid surface moves down almost all of the backbone clearance (0.61 mm or .024 in). The relative rigidity of the backbone forces the rubber gasket to fit inside the cavity provided by the backbone, folding in two regions (Figure 1.1.18–2 and Figure 1.1.18–3). Although the general vicinity of the location of the folds can be estimated from the initial configuration, their exact locations are difficult to predict.

The deformed shape of the gasket and the locations of the folds predicted by Abaqus/Standard and Abaqus/Explicit agree well. The rigid surface load-displacement curve is also in good agreement, as shown in Figure 1.1.18–4.

Acknowledgments

SIMULIA would like to thank Mr. DeHerrera of Freudenberg-NOK General Partnership for providing these examples.

Input files

selfcontact_gask.inp	Gasket model for Abaqus/Standard.
selfcontact_gask_xpl.inp	Gasket model for Abaqus/Explicit.
selfcontact_gask_node.inp	Node definitions for the gasket model.
selfcontact_gask_element1.inp	Element definitions for the rubber part of the gasket model.
selfcontact_gask_element2.inp	Element definitions for the backbone part of the gasket model.
selfcontact_gask_c3d8h.inp	Three-dimensional gasket model for Abaqus/Standard.

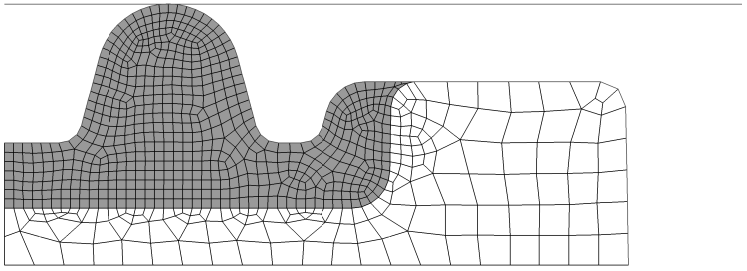


Figure 1.1.18–1 Gasket initial mesh.

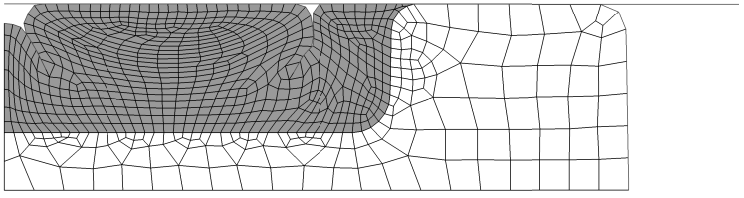


Figure 1.1.18–2 Gasket mesh after loading as predicted by Abaqus/Standard.

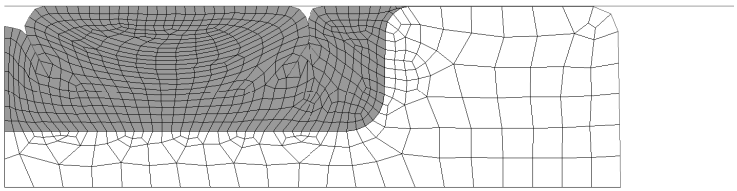


Figure 1.1.18–3 Gasket mesh after loading as predicted by Abaqus/Explicit.

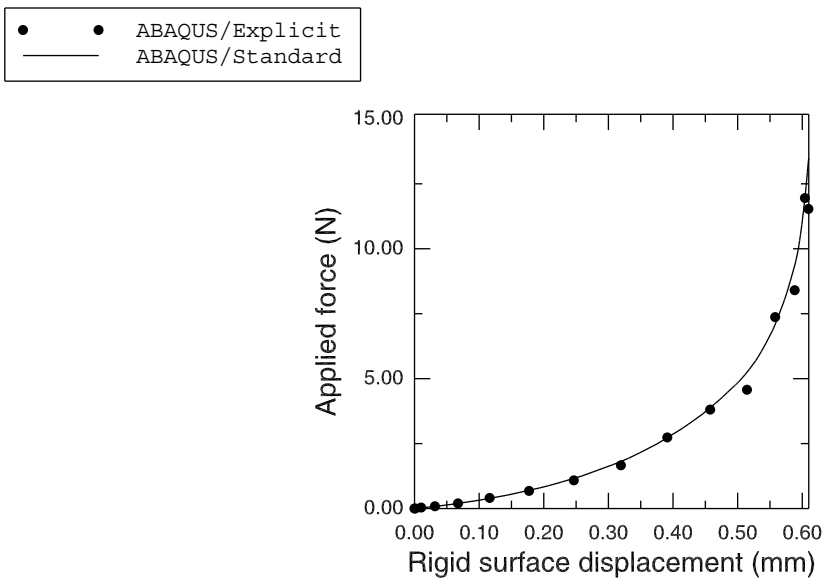


Figure 1.1.18–4 Rigid surface load-displacement curve.

1.1.19 SUBMODELING OF A STACKED SHEET METAL ASSEMBLY

Product: Abaqus/Standard

Sheet metal stampings stacked and fitted on top of each other and secured together via mechanical fasteners such as bolts or rivets are commonly used in the automotive industry. Examples include seat belt anchors and seating track assemblies. The submodeling capability in Abaqus facilitates economical, yet detailed, prediction of the ultimate strength and integrity of such jointed assemblies. A global model analysis of an assembly is first performed to capture the overall deformation of the system. Subsequently, the displacement results of this global analysis are used to drive the boundaries of a submodeled region of critical concern. The submodeling methodology provides accurate modeling that is more economical than using a globally refined mesh in a single analysis.

In a finite element analysis of such a structure, shell elements are commonly used to represent the sheet metal stampings. The nodes of each shell typically lie along the mid-plane of the shell thickness. The thickness of the shells is used in the structural calculations but is not taken into account in the contact calculations. Hence, a structure composed of a stack-up of several sheet stampings may have the nodes of each sheet all lying in the same spatial plane. This close proximity creates uncertainty in a submodel analysis since Abaqus will not be able to determine the correct correspondence between the sheets in the submodel and the global model. Therefore, Abaqus provides a capability that allows the user to specify particular elements of the global model that are used to drive a particular set of nodes in a submodel, which eliminates the uncertainty. This capability is demonstrated in this example problem.

Geometry and model

The global model consists of five separate metal stampings meshed with S4R and S3R shell elements. An exploded view of the global finite element model is shown in Figure 1.1.19–1. The stampings are stacked one upon the other by collapsing the configuration in the 3-direction. All the shell elements are 0.5 mm thick, with all nodes positioned at the mid-surface of each shell. The separate meshes are connected together with BEAM-type MPCs between corresponding perimeter nodes on the large bolt holes through each layer. The nodes on the edges of the two small holes at the bottom of Layer 1 are constrained in all six degrees of freedom, representing the attachment point to ground. The translational degrees of freedom of the nodes around the perimeter of Layer 2 are also constrained, representing the far-field boundary condition in that plate.

Several surface definitions are used to model the contact between the various adjacent layers. The contact definitions prevent unwanted penetration between shell element layers. The small-sliding contact formulation is employed. Most of the contact in this problem is between adjacent layers, but there is also direct contact between Layer 2 and Layer 4. To avoid overconstraints, it is important that no point on Layer 4 simultaneously contact Layer 3 and Layer 2; therefore, node-based surfaces are used for the slave surfaces. This precludes accurate calculation of contact stresses, but that is not important in this case since more accurate contact stresses are obtained in the submodel.

All five stampings are made of steel and are modeled as an elastic-plastic material. The elastic modulus is 207,000 MPa, Poisson's ratio is 0.3, and the yield stress is 250 MPa. The metal plasticity definition includes moderate strain hardening.

The submodel stampings are truncated versions of the global model, located in the same physical position as the global model. In this case these are the regions of concern for high stresses and potential failure of the joint. The submodel is discretized with a finer mesh than the global model to provide a higher level of accuracy. Figure 1.1.19–2 shows an exploded view of the submodel. Because the stampings in the submodel contain the large bolt holes, the submodel contains BEAM-type MPCs in a manner analogous to that in the global model.

The submodel has several surface definitions and contact pairs to avoid penetration of one stamping into another. The submodel contains no node-based surfaces, however. The contact is modeled as element-based surface-to-surface in each layer.

The material definition and shell thicknesses in the submodel are the same as those in the global model.

Results and discussion

The global model is loaded by enforcing prescribed boundary conditions on the protruding edge of Layer 3. This edge is displaced −5.0 mm in the 1-direction and −12.5 mm in the 3-direction. Figure 1.1.19–3 shows the deformed shape of the global model. The displacements at the nodes are saved to the results file for later use by the submodel analysis.

The submodel driven nodes are loaded using a submodel boundary condition. The perimeter nodes of each layer of the submodel that correspond to a "cut" out of the global geometry are driven by the interpolated nodal displacement results in the global results file. Each driven node set is in a separate shell layer. Therefore, the submodel analysis contains multiple submodels, which designate the global model element sets to be searched for the responses that drive the submodel driven node sets. For example, the driven nodes in submodel Layer 1 (node set **`L1BC`**) are driven by the results for the global element set which contains the elements of (global) Layer 1 (element set **`LAYER1`**). The driven nodes for Layers 2–4 are specified in a similar way. Because submodel Layer 5 has no driven nodes, only four submodels are required.

Figure 1.1.19–4 shows the deformed shape of the submodel. Figure 1.1.19–5 and Figure 1.1.19–6 show contour plots of the out-of-plane displacements in Layer 2 for the global model and submodel, respectively. In both cases the displacement patterns are similar; however, the maximum displacement predicted by the global model is about 7.8% larger than that predicted by the submodel.

Input files

stackedassembly_s4r_global.inp	S4R global model.
stackedassembly_s4r_global_mesh.inp	Key input data for the S4R global model.
stackedassembly_s4r_sub.inp	S4R submodel.
stackedassembly_s4r_sub_mesh.inp	Key input data for the S4R submodel.

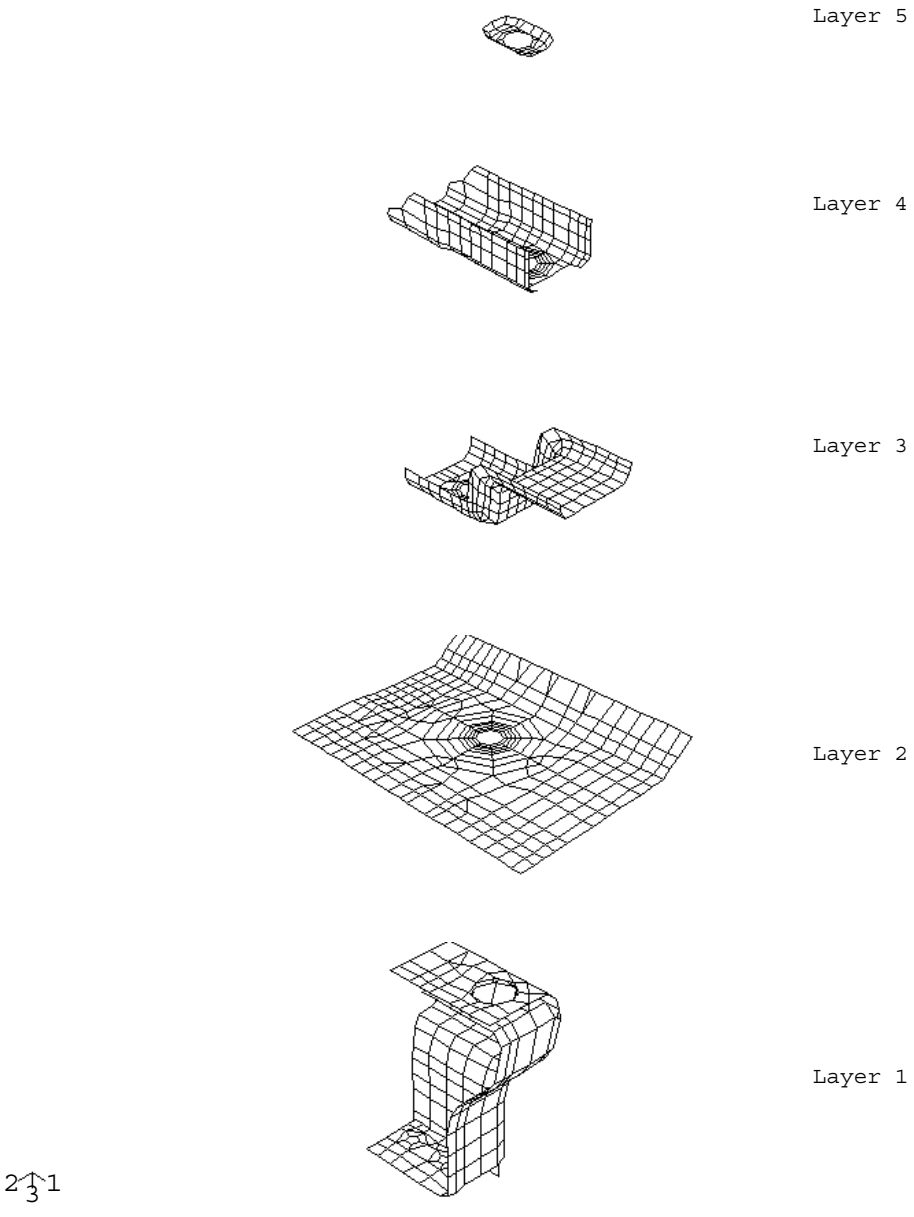


Figure 1.1.19–1 Exploded view of global model.

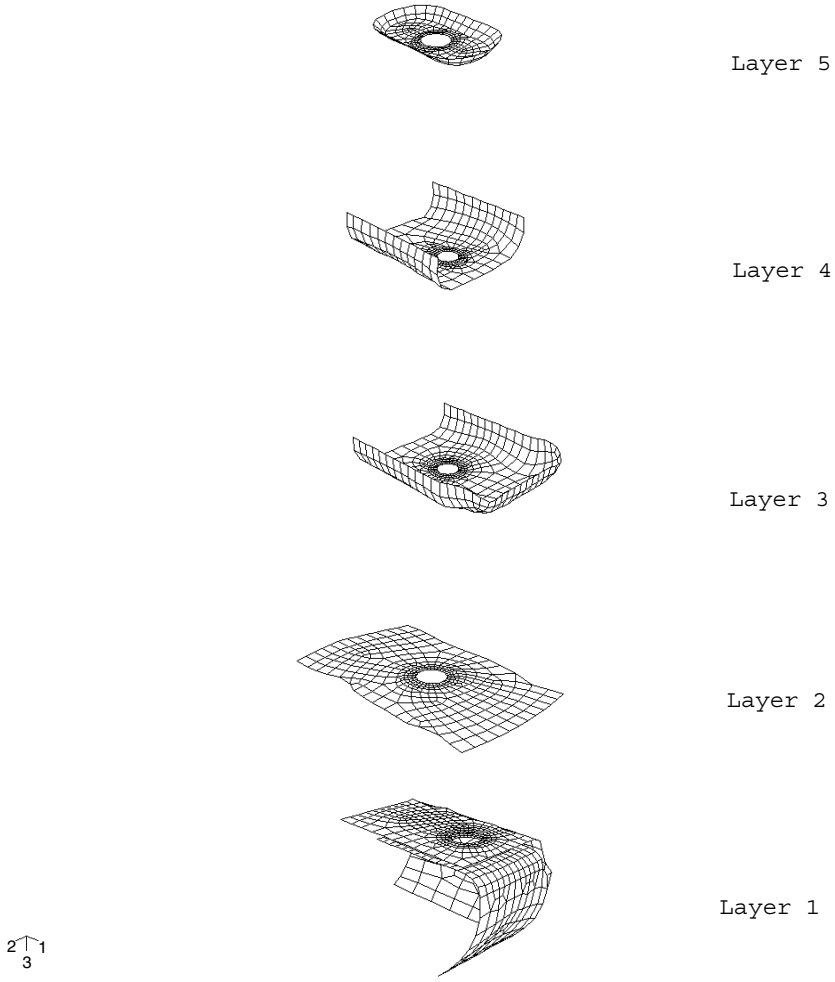


Figure 1.1.19–2 Exploded view of submodel.

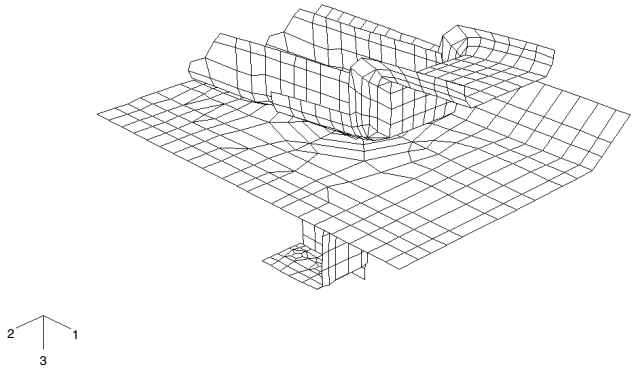


Figure 1.1.19–3 Deformed shape of global model.

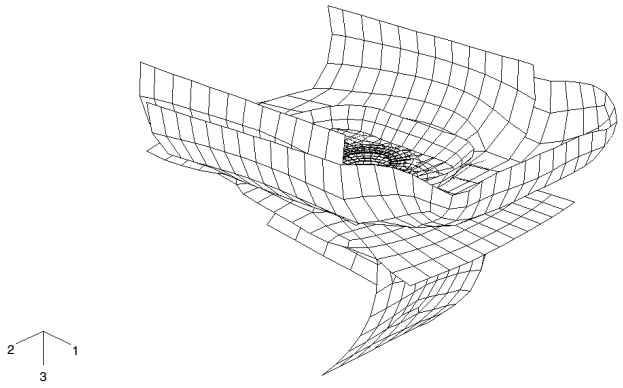


Figure 1.1.19–4 Deformed shape of submodel.

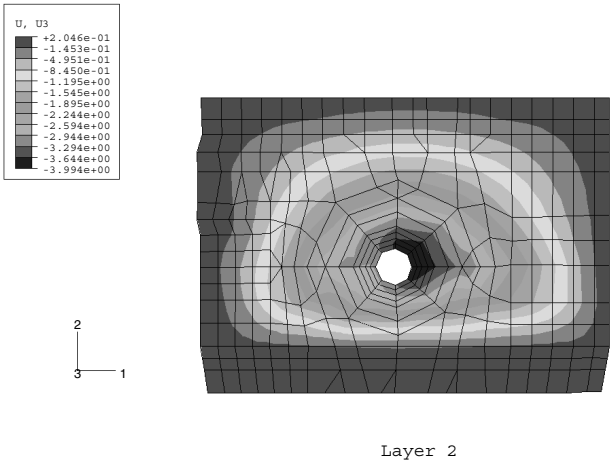


Figure 1.1.19–5 Out-of-plane displacement in Layer 2, global model.

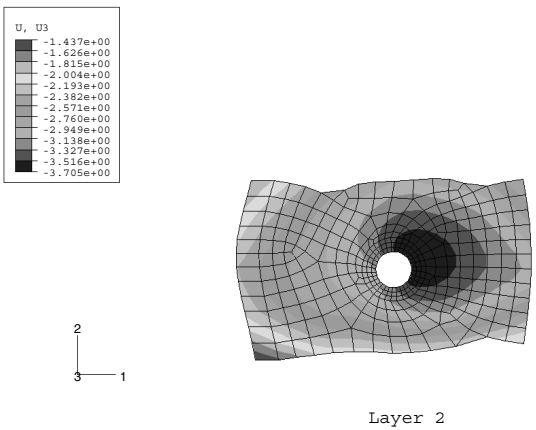


Figure 1.1.19–6 Out-of-plane displacement in Layer 2, submodel.

1.1.20 AXISYMMETRIC ANALYSIS OF A THREADED CONNECTION

Product: Abaqus/Standard

Threaded connectors are commonly used components in the piping and offshore industry. They must withstand a variety of loading conditions: thread engagement, torque, bending, axial pullout, internal pressure under operating and overload conditions, and potential fluid leakage through threaded connections. Abaqus provides a wide range of modeling, analysis, and output capabilities that can be used to assess the design of a connector under these and other loading conditions.

Two Abaqus methods that are particularly useful for analyzing threaded connectors are the specification of an allowable contact interference and of pressure penetration loads.

- The automatic "shrink" fit method can be used to automatically resolve the overclosure of two contacting surfaces. This method is applicable only during the first step of an analysis, and it cannot be used with self-contact. See "Modeling contact interference fits in Abaqus/Standard," Section 36.3.4 of the Abaqus Analysis User's Guide, for details.
- The surface-based pressure penetration capability described in "Pressure penetration loading," Section 37.1.7 of the Abaqus Analysis User's Guide, is used to simulate pressure penetration between contacting surfaces. This capability is provided for simulating cases where a joint between two deforming bodies (for example, between two components threaded onto each other) or between a deforming body and a rigid surface (such as a soft gasket used in a joint) is exposed at one or multiple ends to a fluid pressure. This pressure will penetrate into the joint and load the surfaces forming the joint until some area of the surfaces is reached where the contact pressure between the abutting surfaces exceeds the critical value specified in the pressure penetration load, cutting off further penetration.

The contact output variables in Abaqus can provide the designer a wealth of information about the performance of a connector during all steps of an analysis. When modeling surface-based contact with axisymmetric elements (CAX- and CGAX-type elements) an output quantity of particular use is the maximum torque that can be transmitted about the z-axis by a specified contact pair. The maximum torque, T, is a scalar value defined as

$$T = \iint r^2 p \, ds \, d\theta,$$

where p is the pressure transmitted across the interface, r is the radius to a point on the interface, and s is the current distance along the interface in the r–z plane. T is not a real torque; it is a computed limit of torque that a contact pair may transmit about the z-axis assuming that all the slave nodes on the contact surface are slipping and that the friction coefficient is set to 1. The actual maximum torque that can be transmitted about the z-axis by a specified contact pair can be estimated by scaling T by the friction coefficient specified for the contact pair. The value of T can be output by requesting the contact output variable CTRQ.

This example demonstrates the usefulness of the specification of allowable contact interference and of pressure penetration loads as well as the Abaqus contact output variables in an axisymmetric analysis of a particular threaded connector.

Geometry and model

A three-dimensional cut-away view of the threaded connection assembly analyzed in this example is shown in Figure 1.1.20–1. Although the actual threads are helical, they are represented with an axisymmetric geometry. Previous experience has shown this simplification to be appropriate for these types of problems. Both the "pin" and the "box" are made from steel with a Young's modulus of 207 GPa (30×10^6 psi) and a Poisson's ratio of 0.3, which is characterized by a von Mises plasticity model. The unthreaded section of the pin has inner and outer radii of 48.6 mm (1.913 in) and 57.2 mm (2.25 in), respectively. The major diameter of the threads on the pin (diameter measured at the crest of the threads) is slightly larger than the major diameter of the threads on the box (diameter measured at the roots of the threads); thus, there is an initial interference between the threads on the pin and on the box.

The deformed axisymmetric mesh (after the initial interference has been resolved) in the vicinity of the threads is illustrated in Figure 1.1.20–2. Contact is modeled by the interaction of contact surfaces defined by grouping specific faces of the elements in the contacting regions.

Loading and boundary conditions

Two analyses of the threaded connection are performed: an axisymmetric analysis using CAX4 elements and an axisymmetric analysis with twist using CGAX4 elements. The first four steps for the two analyses are identical. The CGAX4 model has an additional fifth step.

The initial interference fit of the threads on the pin and box is resolved in the first step using the automatic "shrink" fit method with a friction coefficient of 0. In the second step the assembly is held fixed while the friction coefficient is changed from 0 to 0.1 using changes to friction properties. An internal gauge pressure of 0.689 MPa (100 psi) is applied to the connector in the third step. The pressure on the contact surfaces is applied using pressure penetration loading. In the first three steps the displacements in the 2-direction are constrained to be zero at both ends of the assembly. To simulate an axial load in the fourth step, a displacement boundary condition of −0.254 cm (−0.1 in) is applied to the end of the box in the 2-direction. In the fifth step for the CGAX4 model the end of the pin is held fixed while the end of the box is rotated 0.1 radians about the 2-axis, simulating a torque being applied to the connector. The actual torques generated about the 2-axis by the frictional stresses in the fifth step are given by the output variable CMS2. This value is compared to the estimated value given by CTRQ for the fourth step.

Results and discussion

All analyses are performed as large-displacement analyses. The results from the first four steps for both models are identical. Figure 1.1.20–3 and Figure 1.1.20–4, respectively, show the Mises stress distributions in the threaded assembly after the overclosure has been resolved in Step 1 and after the displacement boundary condition has been applied in Step 4. As is illustrated in Figure 1.1.20–4, some of the threads on the pin are beginning to pull out at the end of Step 4. However, plots of the pressure penetration on the contact surface of the box in Figure 1.1.20–5 and Figure 1.1.20–6 show that the seal of the threads is maintained; thus, no leakage is indicated. If the seal had failed, the penetration pressure on the box surface in contact with the pin would be 0.689 MPa (100 psi) instead of 0. Other contact

output variables such as CPRESS and COPEN provide additional information about the contact state throughout the analysis.

The scaled values of CTRQ (scaled by the friction coefficient of 0.1) and the values of CMS2 for all five steps in the CGAX4 analysis are illustrated in Figure 1.1.20–7. The value of CTRQ at the end of Step 4 is 1.22×10^6 lb-in, which translates into an estimated maximum torque of 1.22×10^5 lb-in for a friction coefficient of 0.1. The value of CMS2 computed during Step 5 for the CGAX4 model is 1.18×10^5 lb-in. The 3.7% difference in this example between the predicted and actual torque values can be attributed to a slight change in the normal pressure distribution between the contact surfaces that occurs when the box is rotated. The value of CMS2 is zero for the first four steps since no frictional stresses are generated between the contact surfaces until the fifth step. The value of CTRQ increases in the first step as the overclosure is resolved and dips in the fourth step due to the change in the contact pressure as the box is pulled away from the pin (see Figure 1.1.20–7).

Acknowledgments

SIMULIA gratefully acknowledges the ExxonMobil Upstream Research Corporation for their cooperation in implementing the CTRQ output variable and for supplying the geometry, mesh, and material properties used in this example.

Input files

threadedconnector_cax4.inp	Axisymmetric analysis of the threaded connector using CAX4 elements.
threadedconnector_cax4_n.inp	Node definitions for the axisymmetric analysis of the threaded connector using CAX4 elements.
threadedconnector_cax4_e.inp	Element definitions for the axisymmetric analysis of the threaded connector using CAX4 elements.
threadedconnector_cgax4.inp	Axisymmetric analysis of the threaded connector using CGAX4 elements.
threadedconnector_cgax4_n.inp	Node definitions for the axisymmetric analysis of the threaded connector using CGAX4 elements.
threadedconnector_cgax4_e.inp	Element definitions for the axisymmetric analysis of the threaded connector using CGAX4 elements.

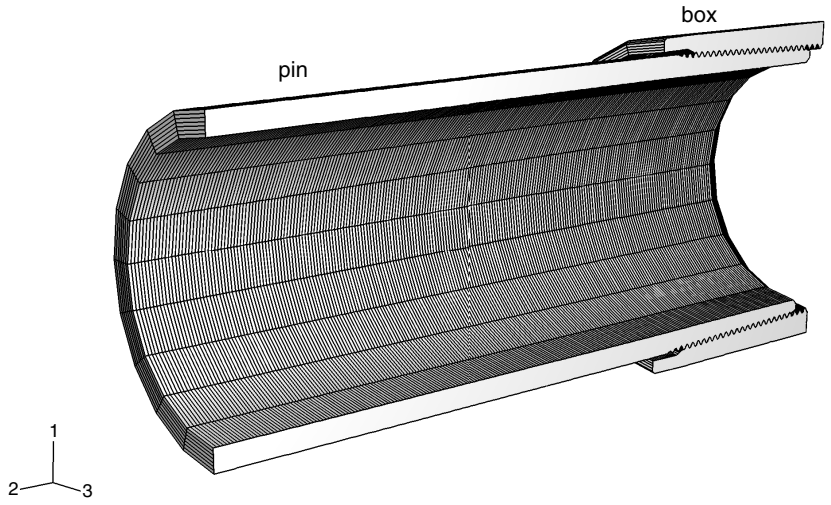


Figure 1.1.20–1 Three-dimensional cut-away view of the threaded connection.

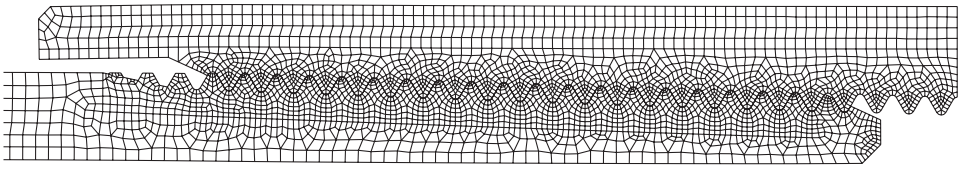


Figure 1.1.20–2 Axisymmetric mesh in the vicinity of the threads after the initial interference has been resolved.

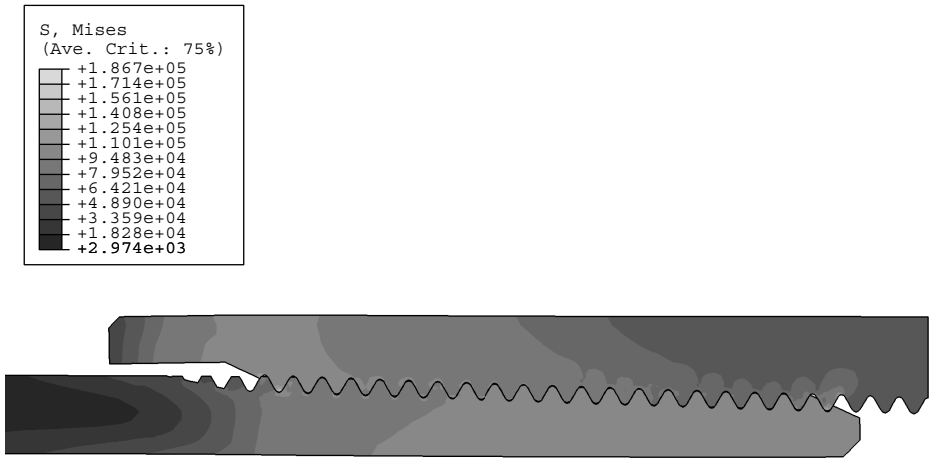


Figure 1.1.20–3 Stress distribution in the threads after the initial overclosure has been resolved.

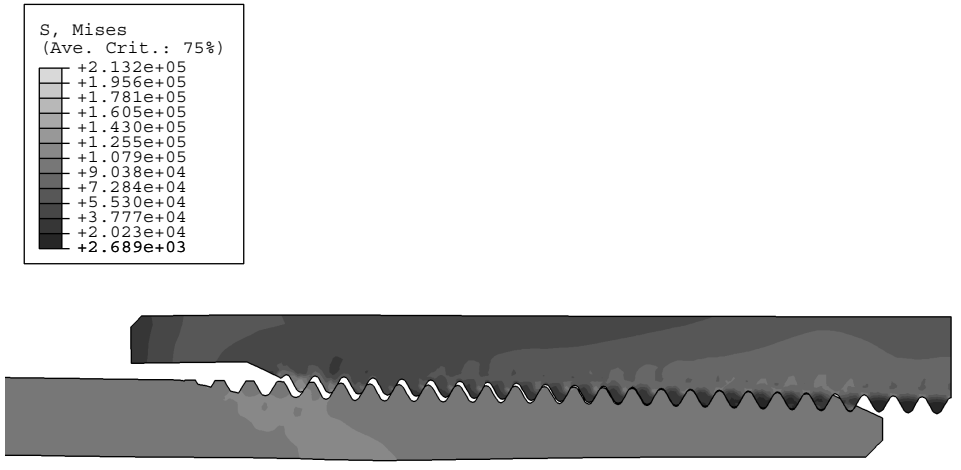


Figure 1.1.20–4 Stress distribution in the threads after axial loading.

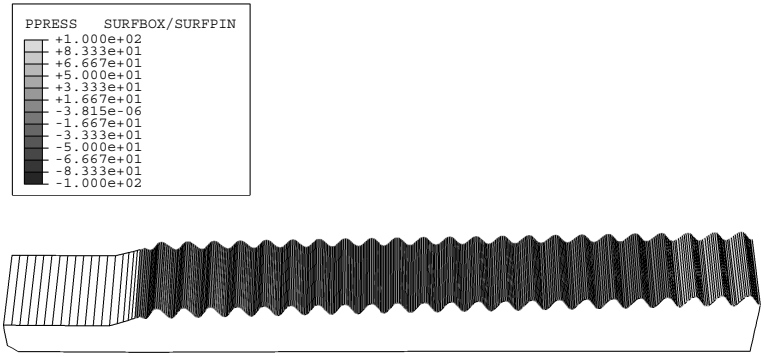


Figure 1.1.20–5 Pressure penetration on box contact surface after axial loading.

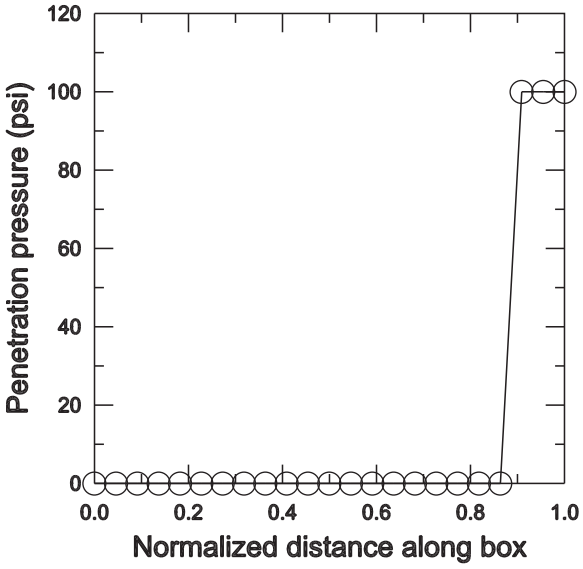


Figure 1.1.20–6 Plot of pressure penetration on box contact surface after axial loading.

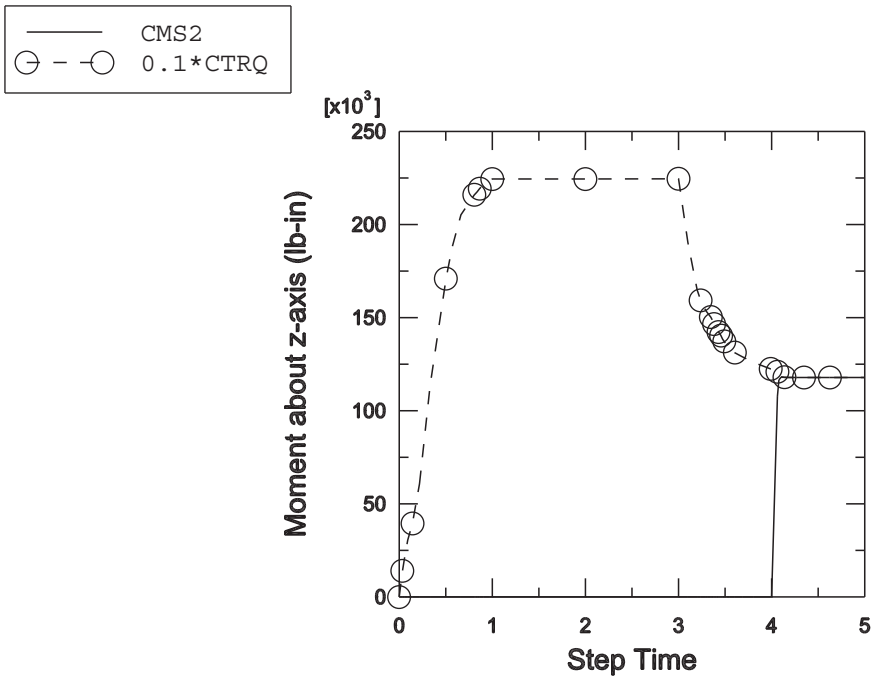


Figure 1.1.20–7 Comparison of 0.1*CTRQ to CMS2 for the CGAX4 model.

1.1.21 DIRECT CYCLIC ANALYSIS OF A CYLINDER HEAD UNDER CYCLIC THERMAL-MECHANICAL LOADINGS

Product: Abaqus/Standard

The prediction of fatigue and failure in structures is fundamental in assessing product performance. This example demonstrates the use of the direct cyclic analysis procedure to obtain results that can be used for fatigue life calculations.

It is well known that a highly loaded structure, such as a cylinder head in an engine subjected to large temperature fluctuations and clamping loads, can undergo plastic deformations. After a number of repetitive loading cycles there will be one of three possibilities: elastic shakedown, in which case there is no danger of low-cycle fatigue; plastic shakedown, leading to a stabilized plastic strain cycle, in which case energy dissipation criteria will be used to estimate the number of cycles to failure; and plastic ratchetting, in which case the design is rejected. The classical approach to obtaining the response of such a structure is to apply the periodic loading repetitively to the structure until a stabilized state is obtained or plastic ratchetting occurs. This approach can be quite expensive, since it may require application of many loading cycles to obtain the steady response. To avoid the considerable numerical expense associated with such a transient analysis, the direct cyclic analysis procedure, described in "Direct cyclic analysis," Section 6.2.6 of the Abaqus Analysis User's Guide, can be used to calculate the cyclic response of the structure directly.

Geometry and model

The cylinder head analyzed in this example is depicted in Figure 1.1.21–1. The cylinder head (which is a single cylinder) has three valve ports, each with an embedded valve seat; two valve guides; and four bolt holes used to secure the cylinder head to the engine block.

The body of the cylinder head is made from aluminum with a Young's modulus of 70 GPa, a yield stress of 62 MPa, a Poisson's ratio of 0.33, and a coefficient of thermal expansion of 22.6×10^{-6} per °C at room temperature. In this example the region in the vicinity of the valve ports, where the hot exhaust gases converge, is subjected to cyclic temperature fluctuations ranging from a minimum value of 35°C to a maximum value of 300°C. The temperature distribution when the cylinder head is heated to its peak value is shown in Figure 1.1.21–2. Under such operating conditions plastic deformation, as well as creep deformation, is observed. The two-layer viscoelastic-elastoplastic model, which is best suited for modeling the response of materials with significant time-dependent behavior as well as plasticity at elevated temperatures, is used to model the aluminum cylinder head (see "Two-layer viscoplasticity," Section 23.2.11 of the Abaqus Analysis User's Guide). This material model consists of an elastic-plastic network that is in parallel with an elastic-viscous network. The Mises metal plasticity model with kinematic hardening is used in the elastic-plastic network, and the power-law creep model with strain hardening is used in the elastic-viscous network. Since the elastic-viscoplastic response of aluminum varies greatly over this range of temperatures, temperature-dependent material properties are specified.

The two valve guides are made of steel, with a Young's modulus of 106 GPa and a Poisson's ratio of 0.35. The valve guides fit tightly into two of the cylinder head valve ports and are assumed to behave

elastically. The interface between the two components is modeled by using matched meshes that share nodes along the interface.

The three valve seats are made of steel, with a Young's modulus of 200 GPa and a Poisson's ratio of 0.3. The valve seats are press-fit into the cylinder head valve ports. This is accomplished by defining radial constraint equations of the form $u^r = u^p - u^s$ between the nodes on the valve seat surface and the nodes on the valve port surface, where u^p is the radial displacement on the valve port, u^s is the radial displacement on the valve seat, and u^r is a reference node. During the first step of the analysis a prescribed displacement is applied to the reference node, resulting in normal pressures developing between the two components. The valve seats are assumed to behave elastically.

All of the structural components (the cylinder head, the valve guides, and the valve seats) are modeled with three-dimensional continuum elements. The model consists of 19394 first-order brick elements (C3D8) and 1334 first-order prism elements (C3D6), resulting in a total of about 80,000 degrees of freedom. The C3D6 elements are used only where the complex geometry precludes the use of C3D8 elements.

Loading and boundary constraints

The loads are applied to the assembly in two analysis steps. In the first step the three valve seats are press-fit into the corresponding cylinder head valve port using linear multi-point equation constraints and prescribed displacement loadings as described above. A static analysis procedure is used for this purpose. The cyclic thermal loads are applied in the second analysis step. It is assumed that the cylinder head is securely fixed to the engine block through the four bolt holes, so the nodes along the base of the four bolt holes are secured in all directions during the entire simulation.

The cyclic thermal loads are obtained by performing an independent thermal analysis. In this analysis three thermal cycles are applied to obtain a steady-state thermal cycle. Each thermal cycle involves two steps: heating the cylinder head to the maximum operating temperature and cooling it to the minimum operating temperature using concentrated flux and film conditions. The nodal temperatures for the last two steps (one thermal cycle) are assumed to be a steady-state solution and are stored in a results (`.fil`) file for use in the subsequent thermal-mechanical analysis. The maximum value of the temperature occurs in the vicinity of the valve ports where the hot exhaust gases converge. The temperature in this region (node 50417) is shown in Figure 1.1.21–3 as a function of time for a steady-state cycle.

In the second step of the mechanical analysis cyclic nodal temperatures generated from the previous heat transfer analysis are applied. The direct cyclic procedure with a fixed time incrementation of 0.25 and a load cycle period of 30 is specified in this step, resulting in a total number of 120 increments for one iteration. The number of terms in the Fourier series and the maximum number of iterations are 40 and 100, respectively.

For comparison purposes the same model is also analyzed using the classical transient analysis, which requires 20 repetitive steps before the solution is stabilized. A cyclic temperature loading with a constant time incrementation of 0.25 and a load cycle period of 30 is applied in each step.

Results and discussion

One of the considerations in the design of a cylinder head is the stress distribution and deformation in the vicinity of the valve ports. Figure 1.1.21–4 shows the Mises stress distribution in the cylinder head at the end of a loading cycle (iteration 75, increment 120) in the direct cyclic analysis. The total strain distribution at the same time in the direct cyclic analysis is shown in Figure 1.1.21–5. The deformation and stress are most severe in the vicinity of the valve ports, making this region critical in the design. The results shown in Figure 1.1.21–6 through Figure 1.1.21–16 are measured in this region (element 50152, integration point 1). Figure 1.1.21–6, Figure 1.1.21–7, and Figure 1.1.21–8 show the evolution of the stress component, plastic strain component, and viscous strain component, respectively, in the global 1-direction throughout a complete load cycle during iterations 50, 75, and 100 in the direct cyclic analysis. The time evolution of the stress versus the plastic strain, shown in Figure 1.1.21–9, is obtained by combining Figure 1.1.21–6 with Figure 1.1.21–7. Similarly, the time evolution of the stress versus the viscous strain, shown in Figure 1.1.21–10, is obtained by combining Figure 1.1.21–6 with Figure 1.1.21–8. The shapes of the stress-strain curves remain unchanged after iteration 75, as do the peak and mean values of the stress over a cycle. However, the mean value of the plastic strain and the mean value of the viscous strain over a cycle continue to grow from one iteration to another iteration, indicating that the plastic ratchetting occurs in the vicinity of the valve ports.

Similar results for the evolution of stress versus plastic strain and the evolution of stress versus viscous strain during cycles 5, 10, and 20 obtained using the classical transient approach are shown in Figure 1.1.21–11 and Figure 1.1.21–12, respectively. The plastic ratchetting is observed to be consistent with that predicted using the direct cyclic approach. A comparison of the evolution of stress versus plastic strain obtained during iteration 100 in the direct cyclic analysis with that obtained during cycle 20 in the transient approach is shown in Figure 1.1.21–13. A similar comparison of the evolution of stress versus viscous strain obtained using both approaches is shown in Figure 1.1.21–14. The shapes of the stress-strain curves are similar in both cases.

One advantage of using the direct cyclic procedure, in which the global stiffness matrix is inverted only once, instead of the classical approach in Abaqus/Standard is the cost savings achieved. In this example the total computational time leading to the first occurrence of plastic ratchetting in the direct cyclic analysis (75 iterations) is approximately 70% of the computational time spent in the transient analysis (20 steps). The savings will be more significant as the problem size increases.

Additional cost savings for the solution can often be obtained by using a smaller number of terms in the Fourier series and/or a smaller number of increments in an iteration. In this example, if 20 rather than 40 Fourier terms are chosen, the total computational time leading to the first occurrence of plastic ratchetting in the direct cyclic analysis (75 iterations) is approximately 65% of the computational time spent in the transient analysis (20 steps). Furthermore, if a fixed time incrementation of 0.735 rather than 0.25 is specified, leading to a total number of 41 increments for one iteration, the total computational time in the direct cyclic analysis is reduced by a factor of three without compromising the accuracy of the results. A comparison of the evolution of stress versus plastic strain obtained using fewer Fourier terms during iteration 75 is shown in Figure 1.1.21–15. A similar comparison of the evolution of stress versus viscous strain obtained using fewer Fourier terms is shown in Figure 1.1.21–16. The shapes of

the stress-strain curves and the amount of energy dissipated during the cycle are similar in both cases, although the case with fewer Fourier terms provides less accurate stress results.

Another advantage of using the direct cyclic approach instead of the classical approach is that the likelihood of plastic ratchetting or stabilized cyclic response can be predicted automatically by comparing the displacement and residual coefficients with some internal control variables. There is no need to visualize the detailed results for the whole model throughout the loading history, which leads to a further reduction of the data storage and computational time associated with output. For this example examination of the displacement and the residual coefficients written to the message (`.msg`) file makes it clear that the constant term in the Fourier series does not stabilize and, thus, plastic ratchetting occurs.

Acknowledgments

SIMULIA gratefully acknowledges PSA Peugeot Citroën and the Laboratory of Solid Mechanics of the Ecole Polytechnique (France) for their cooperation in developing the direct cyclic analysis capability and for supplying the geometry and material properties used in this example.

Input files

dircyccylinderhead_heat.inp	Input data for the heat transfer analysis.
dircyccylinderhead_heat_mesh.inp	Node and element definitions for the heat transfer analysis.
dircyccylinderhead_heat_sets.inp	Node set, element set, and surface definitions for the heat transfer analysis.
dircyccylinderhead_heat_load1.inp	Loading definitions during the heating process for the heat transfer analysis.
dircyccylinderhead_heat_load2.inp	Loading definitions during the cooling process for the heat transfer analysis.
dircyccylinderhead_dcm.inp	Input data for the direct cyclic analysis.
dircyccylinderhead_dcm_mesh.inp	Node and element definitions for the direct cyclic analysis.
dircyccylinderhead_dcm_sets.inp	Node set and element set definitions for the direct cyclic analysis.
dircyccylinderhead_dcm_eqc.inp	Kinematic constraint definitions for the direct cyclic analysis.
dircyccylinderhead_dcm_ps.inp	Post output for the direct cyclic analysis.

References

- Maitournam, H., B. Pommier, and J. J. Thomas, “Détermination de la réponse asymptotique d’une structure anélastique sous chargement thermomécanique cyclique,” C. R. Mécanique, vol. 330, pp. 703–708, 2002.
- Maouche, N., H. Maitournam, and K. Dang Van, “On a new method of evaluation of the inelastic state due to moving contacts,” Wear, pp. 139–147, 1997.

- Nguyen-Tajan, T. M. L., B. Pommier, H. Maitournam, M. Houari, L. Verger, Z. Z. Du, and M. Snyman, "Determination of the stabilized response of a structure undergoing cyclic thermal-mechanical loads by a direct cyclic method," ABAQUS Users' Conference Proceedings, 2003.

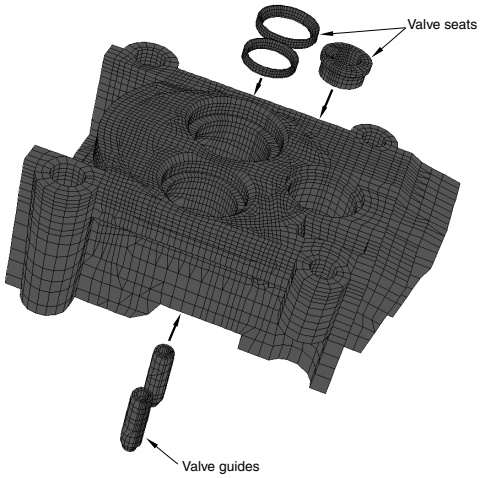


Figure 1.1.21–1 A cylinder head model.

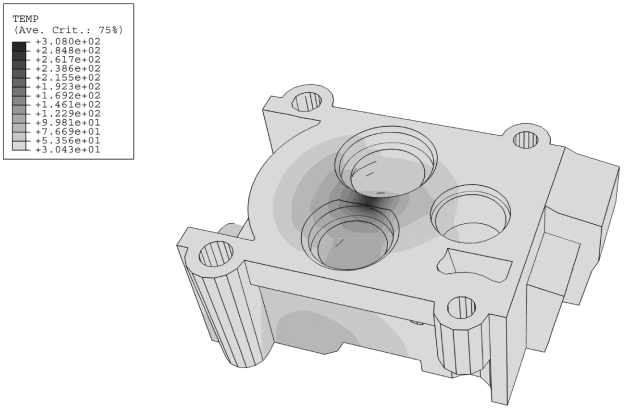


Figure 1.1.21–2 Temperature distribution when the cylinder head is heated to its peak value.

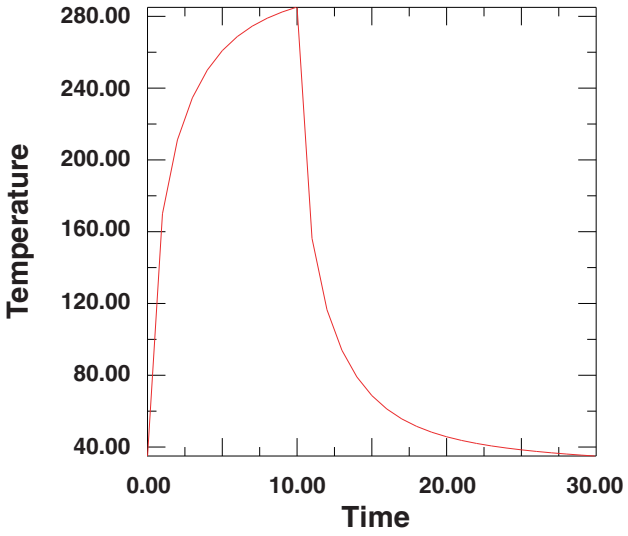


Figure 1.1.21–3 Temperature at node 50417 as a function of time for a steady-state cycle.

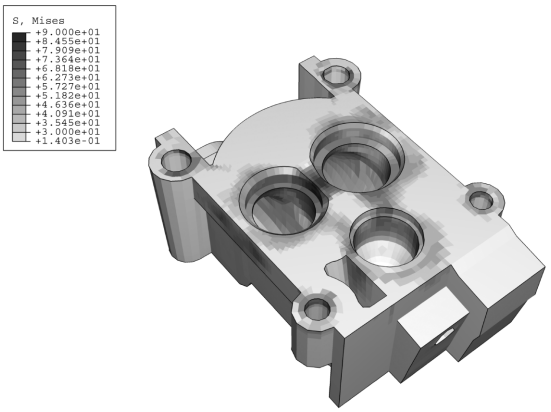


Figure 1.1.21–4 Mises stress distribution in the cylinder head at the end of a loading cycle (iteration 75, increment 120) in the direct cyclic analysis.

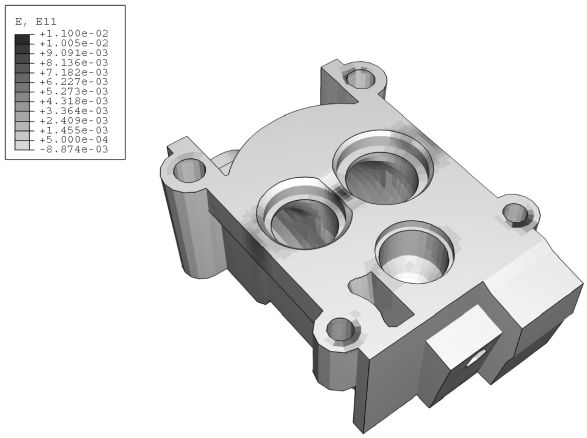


Figure 1.1.21–5 Total strain distribution in the cylinder head at the end of a loading cycle (iteration 75, increment 120) in the direct cyclic analysis.

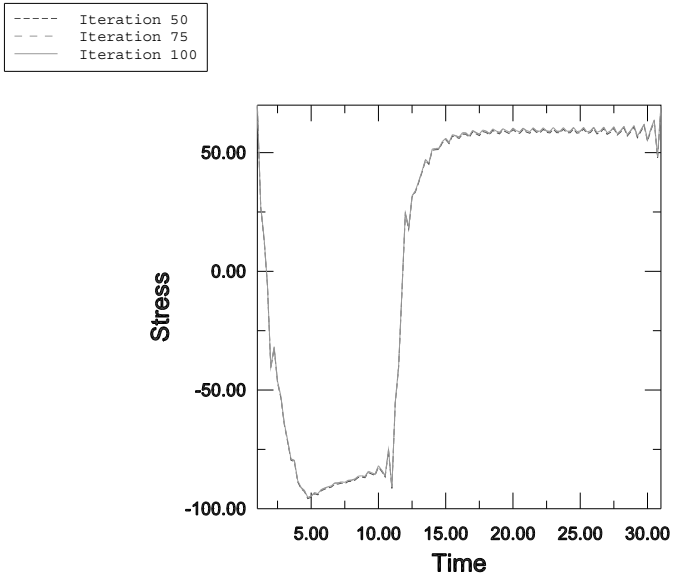


Figure 1.1.21–6 Evolution of the stress component in the global 1-direction during iterations 50, 75, and 100 in the direct cyclic analysis.

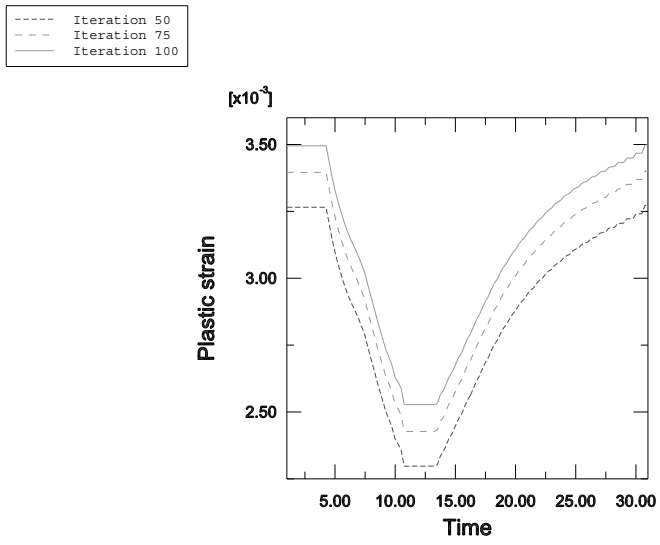


Figure 1.1.21–7 Evolution of the plastic strain component in the global 1-direction during iterations 50, 75, and 100 in the direct cyclic analysis.

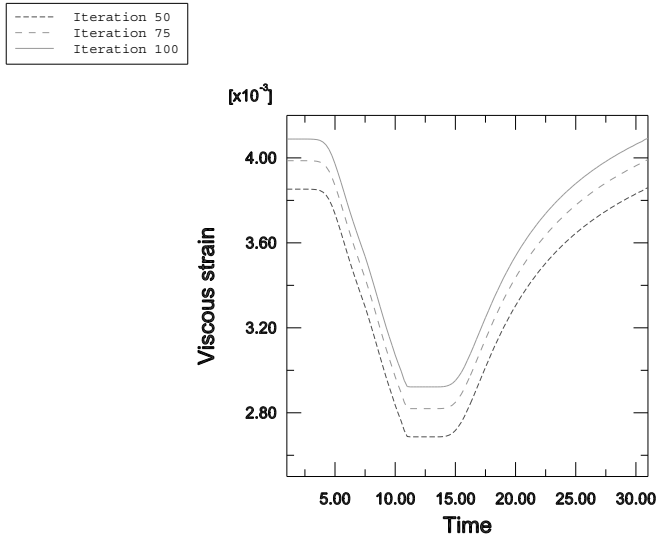


Figure 1.1.21–8 Evolution of the viscous strain component in the global 1-direction during iterations 50, 75, and 100 in the direct cyclic analysis.

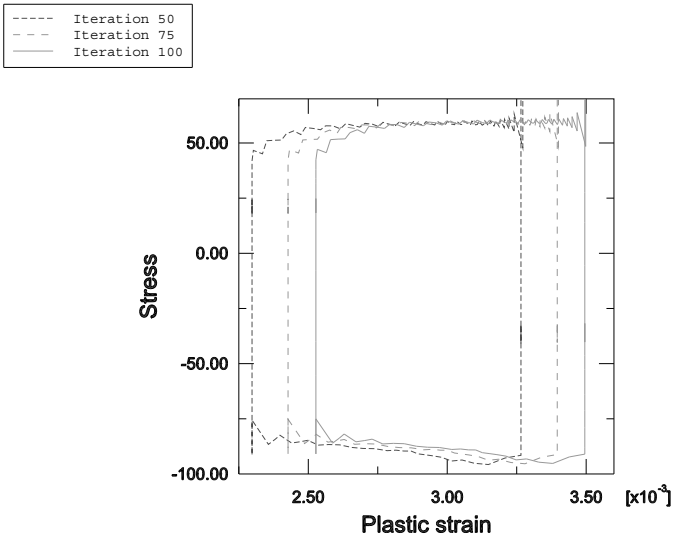


Figure 1.1.21–9 Evolution of the stress versus plastic strain during iterations 50, 75, and 100 in the direct cyclic analysis.

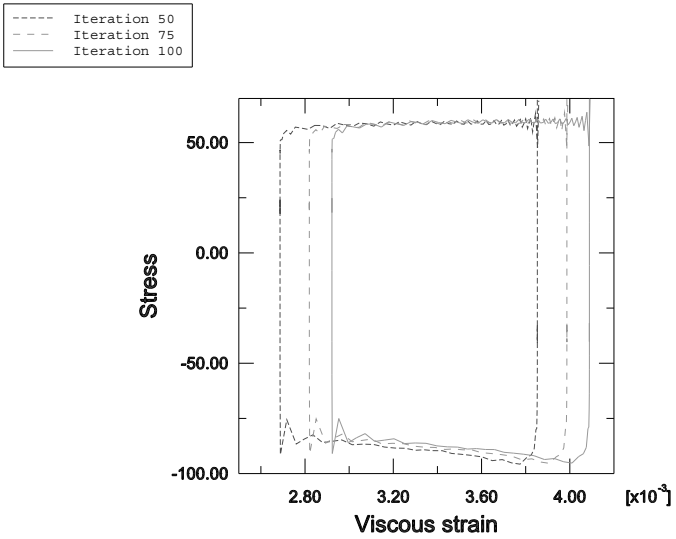


Figure 1.1.21–10 Evolution of the stress versus viscous strain during iterations 50, 75, and 100 in the direct cyclic analysis.

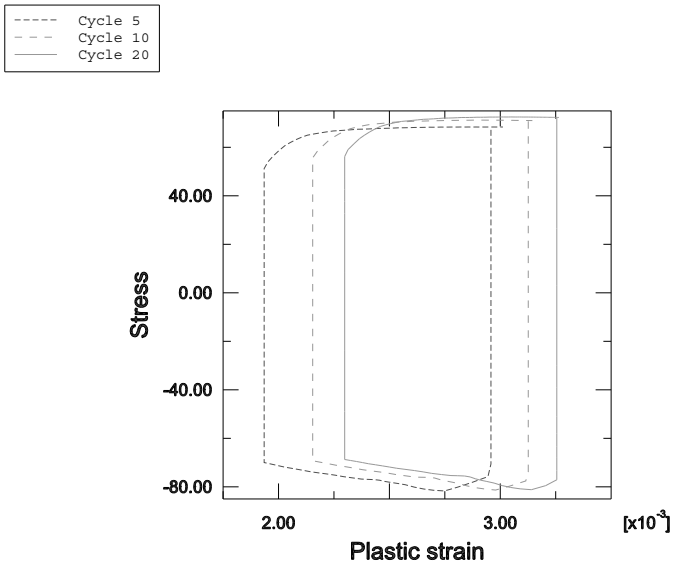


Figure 1.1.21–11 Evolution of the stress versus plastic strain during steps 5, 10, and 20 in the transient analysis.

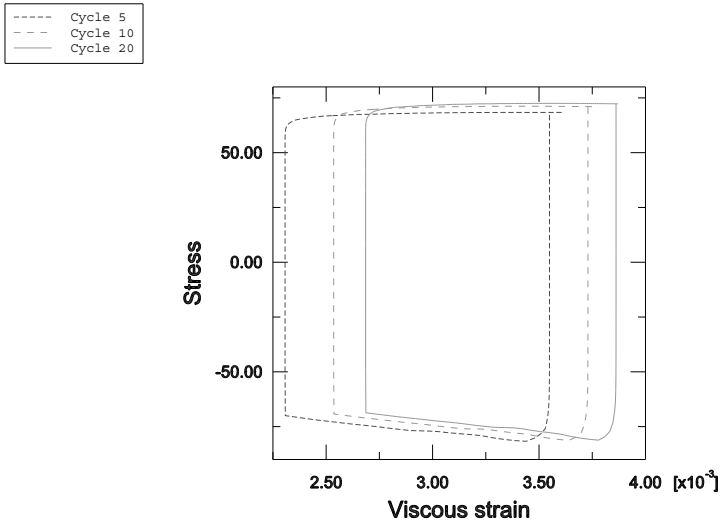


Figure 1.1.21–12 Evolution of the stress versus viscous strain during steps 5, 10, and 20 in the transient analysis.

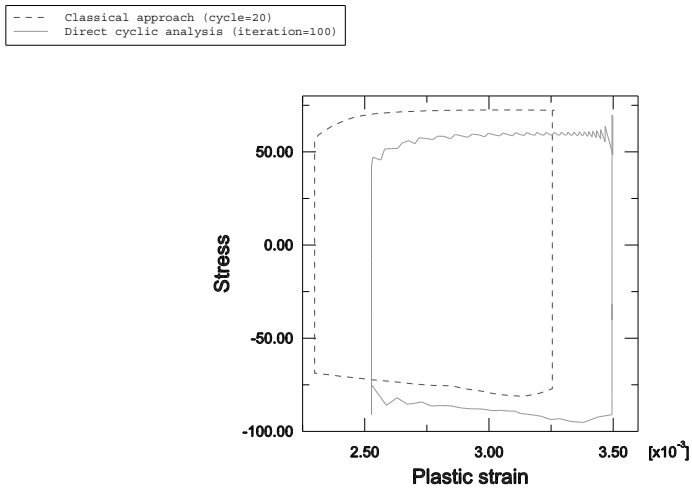


Figure 1.1.21–13 Comparison of the evolution of stress versus plastic strain obtained with the direct cyclic analysis and transient analysis approaches.

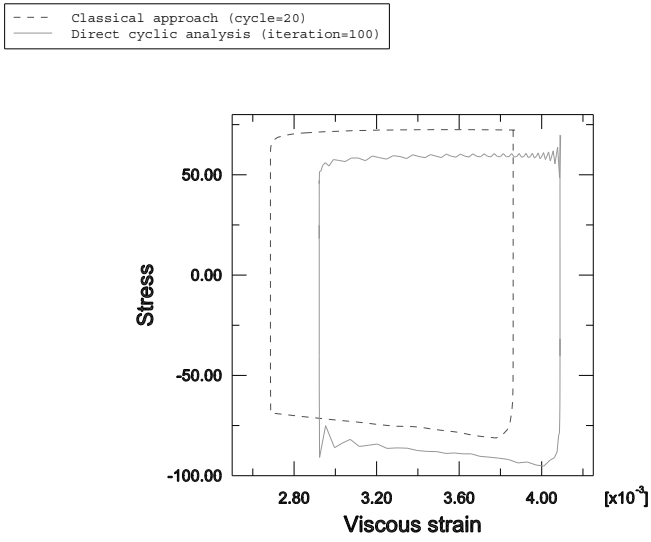


Figure 1.1.21–14 Comparison of the evolution of stress versus viscous strain obtained with the direct cyclic analysis and transient analysis approaches.

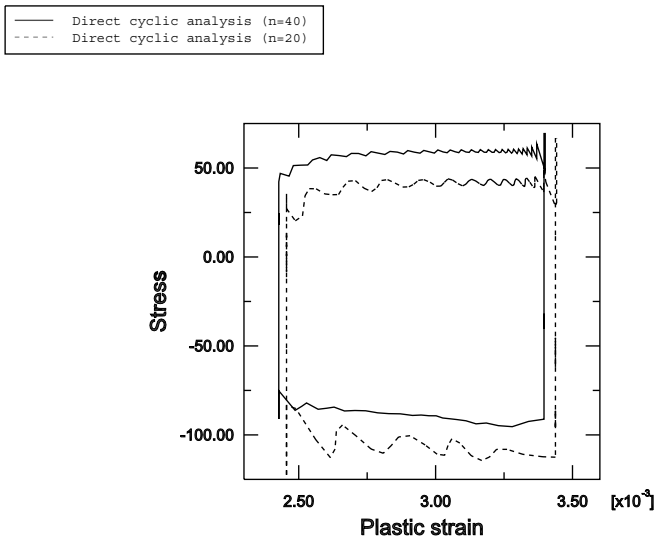


Figure 1.1.21–15 Comparison of the evolution of stress versus plastic strain obtained using different numbers of Fourier terms during iteration 75 in a direct cyclic analysis.

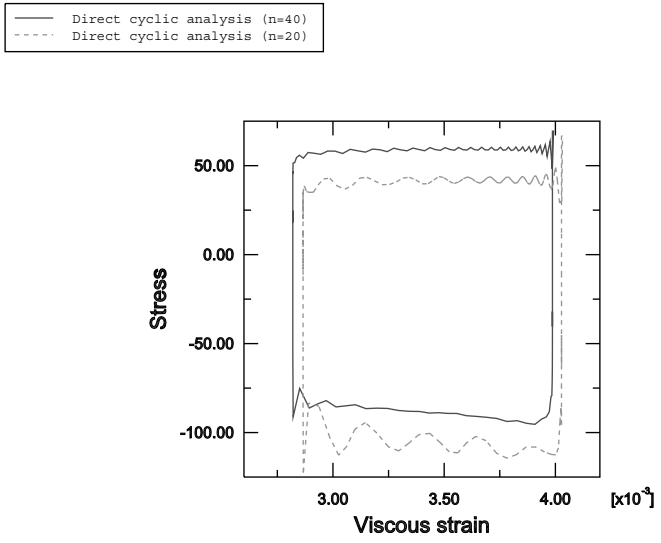


Figure 1.1.21–16 Comparison of the evolution of stress versus viscous strain obtained using different numbers of Fourier terms during iteration 75 in a direct cyclic analysis.

1.1.22 EROSION OF MATERIAL (SAND PRODUCTION) IN AN OIL WELLBORE

Product: Abaqus/Standard

This example demonstrates the use of adaptive meshing and adaptive mesh constraints in Abaqus/Standard to model the large-scale erosion of material such as sand production in an oil well as oil is extracted. In Abaqus/Standard the erosion of material at the external surface is modeled by declaring the surface to be part of an adaptive mesh domain and by prescribing surface mesh motions that recede into the material. Abaqus/Standard will then remesh the adaptive mesh domain using the same mesh topology but accounting for the new location of the surface. All the material point and node point quantities will be advected to their new locations. This example also demonstrates the use of mesh-to-mesh solution mapping in a case where a new mesh topology is desired to continue the analysis beyond a certain stage.

Problem description

The process of optimizing the production value of an oil well is complex but can be simplified as a balance between the oil recovery rate (as measured by the volumetric flow rate of oil), the sustainability of the recovery (as measured by the amount of oil ultimately recovered), and the cost of operating the well. In practice, achieving this balance requires consideration of the erosion of rock in the wellbore, a phenomenon that occurs when oil is extracted under a sufficiently high pressure gradient. This erosion phenomenon is generally referred to as "sand production." Depending on the flow velocities the sand may accumulate in the well, affecting the sustainability of recovery, or it may get carried to the surface along with the oil. The sand in the oil causes erosion in the piping system and its components such as chokes and pipe bends, increasing the costs of operating the well. Excessive sand production is, therefore, undesirable and will limit oil recovery rates. A typical measure of these limits on recovery rates is the so-called "sandfree rate." A sandfree rate might be based on direct damage caused by sand production or might be based on the cost of sand management systems that limit the damage of the sand. The former measure is called the maximum sandfree rate, or MSR. The latter measure is called the acceptable sandfree rate, or ASR. The ASR concept has become possible with the availability of many commercial sand management systems as well as new designs of piping components. The ASR concept has also engendered a need for predicting the sand production rates to properly choose and size the sand management systems and piping components. In this example we focus on measures that can be obtained from Abaqus to predict these rates.

Geometry and model

The geometry of an oil well has two main components. The first is the wellbore drilled through the rock. The second component is a series of perforation tunnels that project perpendicular to the wellbore axis. These tunnels, which are formed by explosive shape charges, effectively increase the surface area of the wellbore for oil extraction. The perforation tunnels are typically arranged to fan out in a helical fashion around the wellbore, uniformly offset in both vertical spacing and azimuthal angle.

Three-dimensional model

The domain of the problem considered in the three-dimensional example is a 203 mm (8 in) thick circular slice of oil-bearing rock with both the wellbore and perforation tunnels modeled. The domain has a diameter of 10.2 m (400 in). The perforation tunnels emanate radially from the wellbore and are spaced 90° apart. Each perforation tunnel is 43.2 mm (1.7 in) diameter and 508 mm (20 in) long. The wellbore has a radius of 158.8 mm (6.25 in). Due to symmetry only a quarter of the domain that contains one perforation tunnel is modeled. Figure 1.1.22–1 shows the finite element model. The rock is modeled with C3D8P elements, and the well's casing is modeled with M3D4 elements.

Planar model

The planar model is a simplified version of the three-dimensional model, where the perforation tunnels and wellbore casing are neglected. The rock is modeled with CPE4P elements. The model domain consists of a quarter-symmetry square domain of length 10.2 m (400 in), with a single wellbore with a radius of 158.8 mm (6.25 in). Figure 1.1.22–5 shows the finite element model. Figure 1.1.22–6 details the region near the wellbore.

Material

A linear Drucker-Prager model with hardening is chosen for the rock, and the casing in the three-dimensional model is linear elastic.

Loading

The loading sequence for a wellbore analysis generally includes

- establishing geostatic equilibrium, based on the overburden loading;
- simulation of material removal operations, including drilling the wellbore and forming the perforation tunnel; and
- applying a drawdown pressure in the wellbore to simulate pumping.

This sequence is modeled slightly differently in the three-dimensional and planar models.

Three-dimensional model

The analysis consists of five steps. First, a geostatic step is performed where equilibrium is achieved after applying the initial pore pressure, the initial stress, and the distributed load representing the soil above the perforation tunnel. The second step represents the drilling operation where the elements representing the wellbore and the perforation tunnel are removed using the element removal capability in Abaqus (see "Element and contact pair removal and reactivation," Section 11.2.1 of the Abaqus Analysis User's Guide). In the third step the boundary conditions are changed to apply the pore pressure on the face of the perforation tunnel. In the fourth step a steady-state soils analysis is carried out in which the pore pressure on the perforation tunnel surface is reduced to the desired drawdown pressure of interest. The fifth step consists of a soils consolidation analysis for four days during which the erosion occurs.

Planar model

The analysis consists of three steps. First, a geostatic step is performed where equilibrium is achieved after applying the initial pore pressure and the initial stress, representing an underbalanced state on the wellbore. The second step represents the drilling operation where the elements representing the wellbore are removed using the element removal capability, and the boundary conditions are changed to apply the pore pressure on the face of the perforation tunnel. The third step consists of a soils consolidation analysis for 32 hours during which the erosion occurs. As discussed below in "Rezoning the planar model," this third step is interrupted in order to rezone the model.

Erosion criterion

There are two main sources of eroded material in a well bore. One of the sources is volumetric and is due to the material that is broken up due to high stresses and transported by the fluid through the pores. The other source is surface based and is due to the material that is broken up by the hydrodynamic action of the flow on the surface. Depending on the properties of the oil-bearing strata and the flow velocities, one or the other may be the dominant source of eroded material. Development of equations describing the erosion in a well bore is an active research field. In this example we consider surface erosion only but choose a form for the erosion equation that has dependencies that are similar to those used by Papamichos and Stavropoulou for volumetric erosion. This approximation is reasonable because high stresses exist only in a very thin layer surrounding the well bore. The erosion equation is

$$\mathrm{V}_e = \lambda(1-n)\, c\, v_w,$$

where V_e is the erosion velocity, v_w is the pore fluid velocity, c is the transport concentration, n is the porosity, and λ is the so-called sand production coefficient. λ depends on the equivalent plastic strain ($\bar{\varepsilon}^{pl}$). It is zero below a cutoff equivalent plastic strain ($\bar{\varepsilon}^{pl}_{\mathrm{cutoff}}$), is equal to $\lambda_1(\bar{\varepsilon}^{pl} - \bar{\varepsilon}^{pl}_{\mathrm{cutoff}})$, and is limited by a constant λ_2.

Both λ_1 and λ_2 must be determined experimentally. In this example $\bar{\varepsilon}^{pl}_{\mathrm{cutoff}} = 0.028$, $\lambda_1 = 4$, and $\lambda_2 = 0.01$. We choose $c = 0.001$ as recommended by Papamichos and Stavropoulo. These values are chosen to show visible erosion in a reasonably short analysis time.

Adaptive mesh domain

Erosion is modeled during the final step of each analysis. The erosion equation describes the velocity of material recession as a local function of solution quantities. Abaqus/Standard provides functionality through adaptive meshing for imposing this surface velocity, maintaining its progression normal to the surface as the surface moves, and adjusting subsurface nodes to account for large amounts of erosive material loss.

Erosion itself is described through a spatial adaptive mesh constraint, which is applied to all the nodes on the surface of the perforation tunnel. Adaptive mesh constraints can be applied only on adaptive mesh domains; in this example a sufficiently large extent of the finite element mesh near the wellbore and perforation tunnel surfaces is declared as the adaptive mesh domain. A cut section of the

adaptive mesh domain for the three-dimensional model, including the perforation tunnel, is shown in Figure 1.1.22–2. The adaptive mesh domain for the planar model is the regular mesh near the wellbore (refer to Figure 1.1.22–6 and Figure 1.1.22–8). Identification of the adaptive mesh domain will result in smoothing of the near-surface mesh that is necessary to enable erosion to progress to arbitrary depths. All the nodes on the boundary of the adaptive mesh domain where it meets the regular mesh must be considered as Lagrangian to respect the adjacent nonadaptive elements.

A velocity adaptive mesh constraint is defined. The generality and solution dependence of the erosion equation are handled by describing the erosion equation in user subroutine **UMESHMOTION** (see "Defining ALE adaptive mesh domains in Abaqus/Standard," Section 12.2.6 of the Abaqus Analysis User's Guide). User subroutine **UMESHMOTION** is called at a given node for every mesh smoothing sweep. Mesh velocities computed by the Abaqus/Standard meshing algorithm for that node are passed into **UMESHMOTION**, which modifies them to account for the erosion velocities computed at that node. The modified velocities are determined according to the equation for V_e, where local results are needed for $\bar{\varepsilon}^{pl}$, $\boldsymbol{n}$, and v_w. To obtain these values, we request results for output variables PEEQ, VOIDR, and FLVEL respectively, noting that void ratio is related to porosity by $(1 - n) = 1/(1 + \mathrm{VOIDR})$. Since these output variables are all available at element material points, the utility routine **GETVRMAVGATNODE** is used to obtain results extrapolated to the surface nodes (see "Obtaining material point information averaged at a node," Section 2.1.8 of the Abaqus User Subroutines Reference Guide).

Rezoning the planar model

Rezoning, the process of creating a new mesh in the model's deformed configuration, is a useful technique in addressing element distortion in erosion problems.

The erosion model in this example is solution dependent in the sense that it defines an erosion velocity as a function of local values of equivalent plastic strain, PEEQ, and fluid velocity, FLVEL. Since these node-located results values are extrapolated from adjacent elements' material point results, using utility routine **GETVRMAVGATNODE**, the quality of the results used in the erosion model is dependent on local element quality. In practice, as elements near the surface deform, there is a tendency for instability in the erosion model.

This instability is mitigated by rezoning the model and providing a more regular mesh to continue the analysis. The rezoned model is created in Abaqus/CAE, and transfer of state variables occurs using mesh-to-mesh solution mapping in Abaqus/Standard. Rezoning occurs at 55,000 seconds (15.3 hours) into the 32-hour erosion analysis of the planar model.

Extracting two-dimensional profiles and remeshing using Abaqus/CAE

The rezoned model is created by extracting the two-dimensional profile of the deformed rock region from the output database for the original analysis. You perform this operation in Abaqus/CAE by entering commands into the command line interface at the bottom of the Abaqus/CAE main window. To extract the deformed geometry from the output database as an orphan mesh part, use the command **PartFromOdb**, which takes the following arguments:

name	The name of the orphan mesh part to be created.
odb	The output database object returned from the command **openOdb**.

instance The name of the part instance in the initial model in capital letters.

shape Determines whether to import the part in its UNDEFORMED or DEFORMED shape.

The command **`PartFromOdb`** returns a Part object that is passed to the command **`Part2DGeomFrom2DMesh`**. This command creates a geometric Part object from the orphan mesh imported earlier. It takes the following arguments:

name The name of the part to be created.

part The part object returned from the command **`PartFromOdb`**.

featureAngle A float specifying the angle (in degrees) between line segments that triggers a break in the geometry.

Once the profile of the deformed part has been created, you will prepare an input file for the subsequent period in the erosion analysis as follows:

- Reestablish all attributes that relate to the geometry of the deformed part. These attributes include load and boundary condition definitions, and set and surface definitions.
- Remesh the part.
- Create a single consolidation step that completes the duration of your intended erosion period.
- Write out the new input file.

Mesh-to-mesh solution mapping in Abaqus/Standard

The interpolation technique used in solution mapping is a two-step process. First, values of all solution variables are obtained at the nodes of the old mesh by extrapolating the values from the integration points to the nodes of each element and averaging those values over all elements abutting each node. The second step is to locate each integration point in the new mesh with respect to the old mesh. The variables are then interpolated from the nodes of the element in the old mesh to the location in the new mesh. All solution variables are interpolated automatically in this way so that the solution can proceed on the new mesh. Whenever a model is mapped, it can be expected that there will be some discontinuity in the solution because of the change in the mesh. To address this discontinuity the rezone analysis includes a geostatic step, which reestablishes equilibrium before the erosion process continues in a following step.

Results and discussion

The analysis scenarios and results differ between the three-dimensional and planar analyses.

Three-dimensional analysis

The consolidation analysis in which erosion takes place is run for a time period of four days to observe the initiation of sand production and predict its initial rate. Figure 1.1.22–3 shows the perforation tunnel at the end of four days where it is seen that the largest amount of material is eroded near the junction of the wellbore and perforation tunnel. Further away from the wellbore boundary the amount of erosion progressively decreases. This behavior is expected because there are high strains near the junction of

the wellbore and perforation tunnel, and the erosion criterion is active only for values of the equivalent plastic strain above a threshold value.

The amount of the volume change due to erosion in an adaptive domain is available using the history output variable VOLC. The actual amount of the solid material eroded depends on the porosity of the rock and is obtained by multiplying VOLC by $(1 - n)$. Figure 1.1.22–4 shows the volume change of the sand produced in cubic inches over the time period of the consolidation step. As the material consolidates, the erosion rate slows down. The stresses in the perforation hole reduce and stabilize over the time period. From Figure 1.1.22–4 it can be concluded that the stresses generated by the drawdown pressure, the fluid velocities, the wellbore and casing geometry, and the initial perforation tunnel geometry are such that this perforation tunnel will produce sand at a more stable rate as oil recovery continues; however, this rate could vary with further changes to the perforation tunnel caused by the erosion. At a design stage any of these parameters could be modified to limit the sand production rate. Many perforation tunnels emanate from a wellbore, and the total sand production from the wellbore will be the sum total of all the perforation tunnels.

Planar analysis

The underbalanced consolidation analysis in which erosion takes place is run for a time period of 32 hours to observe the initiation of sand production and predict its initial rate. Figure 1.1.22–7 shows the wellbore at 15 hours, the end of the first analysis job. Based on this configuration a new mesh is created, as shown in Figure 1.1.22–8. The final configuration of this model, representing 32 hours of erosion, is shown in Figure 1.1.22–9. As expected, the results show that erosion continues outward from the location of the maximum equivalent plastic straining. Figure 1.1.22–10 shows the total amount of sand produced, in cubic inches per inch of depth, over the time period of the consolidation step.

Input files

exa_erosion.inp	Three-dimensional model of the oil wellbore perforation tunnel.
exa_erosion_planar.inp	Planar model of the oil wellbore.
exa_erosion_planar_rezone.inp	Rezone analysis of the planar model.
exa_erosion.f	**UMESHMOTION** user subroutine.

Reference

- Papamichos, E., and M. Stavropoulou, "An Erosion-Mechanical Model for Sand Production Rate Prediction," International Journal of Rock Mechanics and Mining Sciences, no. 35, pp. 4–5, Paper No 090, 1998.

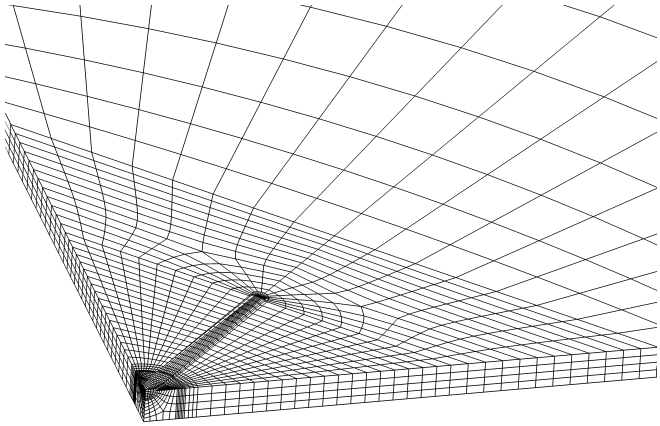


Figure 1.1.22–1 A cut section of the model showing the wellbore and perforation tunnel.

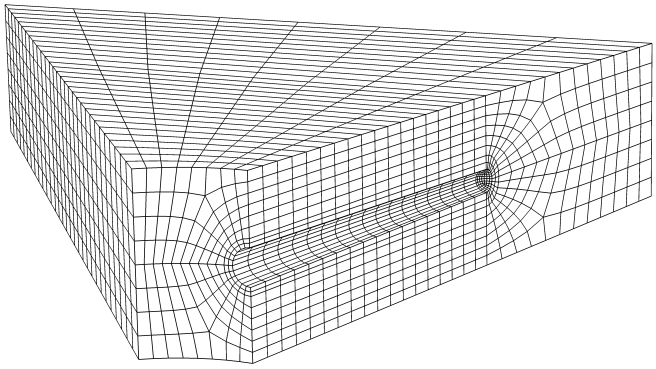


Figure 1.1.22–2 Half-section of the adaptive mesh domain showing the wellbore face and the perforation tunnel.

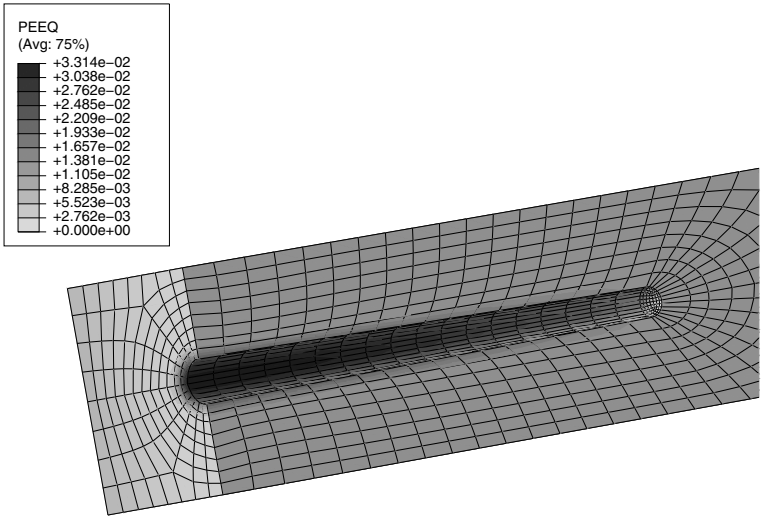


Figure 1.1.22–3 Shape of the perforation tunnel after four days of erosion.

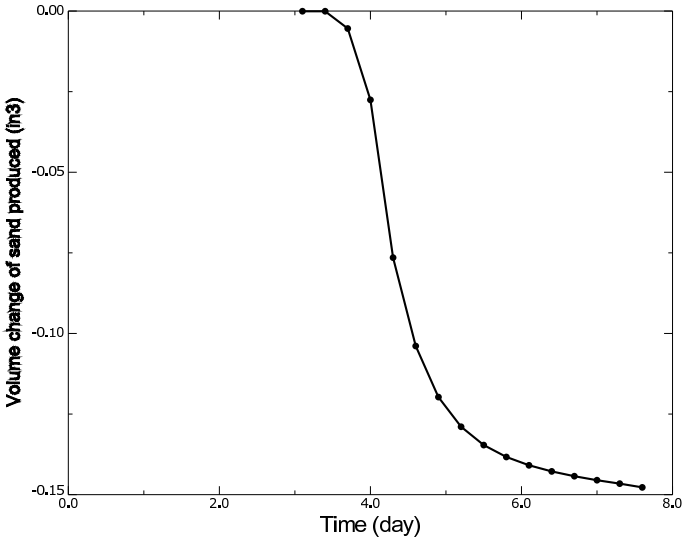


Figure 1.1.22–4 Total sand production volume change in a single perforation tunnel indicating a stabilized rate as the consolidation continues.

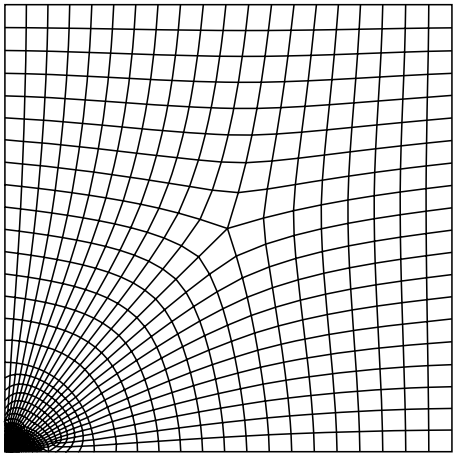


Figure 1.1.22–5 Planar model mesh.

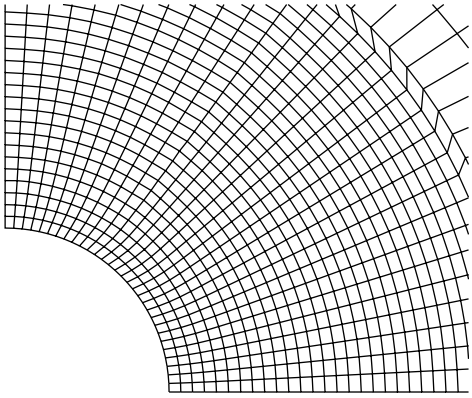


Figure 1.1.22–6 Planar model mesh: wellbore region detail.

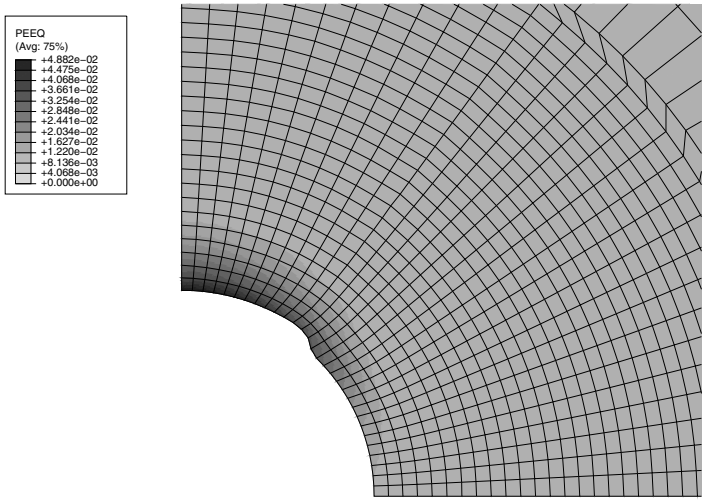


Figure 1.1.22–7 Equivalent plastic strain distribution and eroded shape of the wellbore.

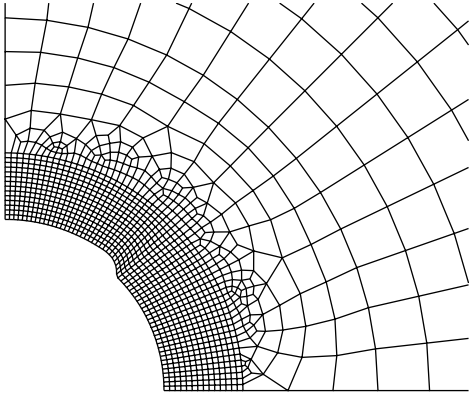


Figure 1.1.22–8 Planar model mesh: wellbore region detail: rezone analysis.

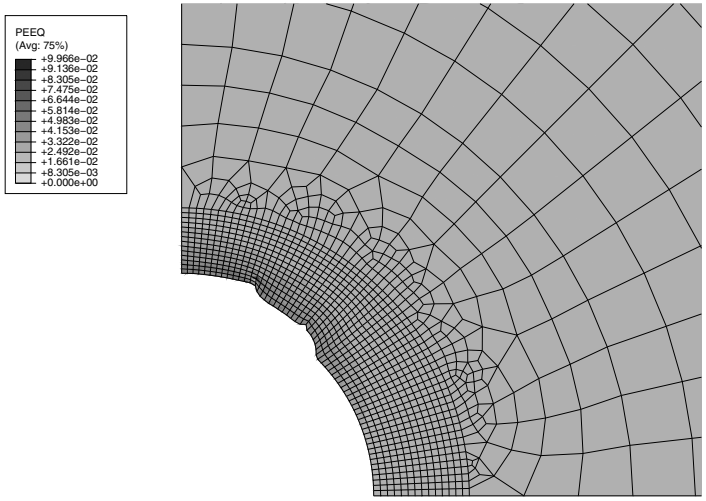


Figure 1.1.22–9 Equivalent plastic strain distribution and eroded shape of the wellbore: rezone analysis.

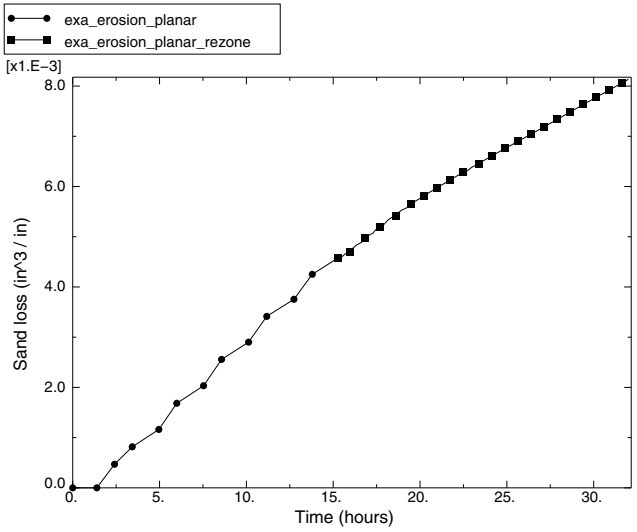


Figure 1.1.22–10 Total sand production volume per unit of depth.

1.1.23 SUBMODEL STRESS ANALYSIS OF PRESSURE VESSEL CLOSURE HARDWARE

Product: Abaqus/Standard

Objectives

This example demonstrates the use of surface-based submodeling as a technique to obtain solutions that are more accurate than those obtained using node-based submodeling in cases where:

- the submodel displacement field is expected to differ from the global model displacement field by a rigid translation and
- the geometry of the submodel differs from the global model in a region whose response is primarily load controlled.

This section details scenarios for each of these cases.

Application description

This example examines the stress behavior of closure head standpipe structures in a nuclear reactor vessel closure assembly. The vessel assembly forms the pressure boundary surrounding the fuel core. This example considers the following loading conditions:

- pre-tension load in the stud bolts,
- constant internal pressure, and
- loading due to the control rod drive mechanism (CDM) plug.

The loading conditions cover the most basic structural operation of a reactor vessel. The International System of units (SI) is used in the following sections to describe the model. The analysis itself is performed in English units. The model and analysis are derived from details of the Shippingport pressurized water reactor (1958).

Geometry

The problem domain comprises a cylindrical vessel shell, a hemispherical bottom head, a dome-shaped closure head, and the closure and seal assembly, as shown in Figure 1.1.23–1. The overall height of the vessel shell including the bottom head is 7650 mm (301 in). The bottom head has an inner radius of 1410 mm (55.5 in) and a thickness of 157 mm (6.18 in). The inlet nozzles on the bottom head are not considered in this example. The inner radius of the vessel shell is 1380 mm (54.5 in), and the thickness is 213 mm (8.40 in). The closure head has a height of 2330 mm (91.8 in), an inner radius of 1310 mm (51.5 in), and a thickness of 210 mm (8.25 in).

The closure head includes eight standpipes with CDM plugs inserted in each. The standpipes have an inside diameter of 472 mm (12 in) and an outside diameter of 630 mm (16 in) and extend roughly 1000 mm (25 in) above the closure head. The CDM plugs have an outside diameter of 465 mm (11.8 in) and a flange diameter of 630 mm (16 in) and are 748 mm (19 in) tall. Each CDM plug sits in a closure head standpipe on a 404 mm (10.25 in) diameter ledge.

The closure head is attached to the vessel shell by a seal and closure assembly. The assembly includes 40 stud bolts passing through the bolting flanges of the closure head and the vessel shell, each of which is restrained by two cap nuts (one on each end). Each stud bolt is 2290 mm (90 in) in length and has a diameter of 146 mm (5.75 in). The closure nuts are 304 mm (12 in) long with a thickness of 28.6 mm (1.13 in). To complete the closure assembly, an Omega seal is welded to the under surface of the closure head and top surface of the vessel shell.

Materials

All components are constructed of high-strength steel.

Boundary conditions and loading

A pre-tension load of 2200 kN (5×10^6 lbf) is applied to each bolt. The inner surfaces of the head and the vessel shell are subject to a constant pressure of 1.38×10^7 Pa (2000 psi) from the water.

Interactions

Contact occurs between

- the reactor vessel and closure head,
- the lower nuts and the reactor vessel bolting flange,
- the upper nuts and the closure head bolting flange, and
- the closure head standpipe and CDM plug.

Model terminology

This example illustrates the use of the submodeling technique in ways that are generalizations of the Abaqus user interface concept of a global model driving the response of a submodel. Specifically, some submodel analyses described in this section represent material domains that are adjacent to, rather than lying within, the domain considered in the initial analysis. To more clearly describe the models in this example, the term “source model” is used instead of “global model.” A source model is a model that provides solution results to a subsequent submodel analysis.

Abaqus modeling approaches and simulation techniques

The objective of the analyses in this example is an understanding of stresses in the region of the closure head standpipe.

Summary of analysis cases

Case 1 Reactor closure analysis: reference solution	The stress distribution in the closure head is determined from a single, finely meshed model that includes closure head standpipe and CDM plug details.

Case 2 Submodeling of the closure head standpipe region	A defeatured source model of the vessel assembly is analyzed first. Features excluded are the closure head standpipes and the CDM plugs. This model then drives a submodel with a more detailed representation of the closure head standpipe region.
Case 3 Submodel application of CDM hardware loading	The CDM plug is analyzed separately as a source model. Boundary conditions are introduced on the CDM plug where it interacts with the closure head standpipe seating ledge to determine the surface traction characteristics at this interface. This model is then used to drive a submodel of the remaining vessel assembly.

In the two submodel analysis cases the node-based submodeling technique, in which the submodel is driven with displacements, is compared to the surface-based submodeling technique, in which the submodel is driven with stresses. The three cases are discussed in more detail below.

Case 1 Reactor closure analysis: reference solution

This reference case determines the stress response of the reactor vessel assembly when subjected to boltup and pressure loading using a single analysis. By comparison, the other modeling cases make use of submodeling. The reactor vessel assembly is cyclic-symmetric with respect to the axis of the cylindrical vessel body and only one-quarter of the whole assembly is modeled. The global geometry is shown in Figure 1.1.23–2.

Analysis types

A static stress analysis is performed.

Mesh design

The vessel is meshed with C3D20R elements, and the closure head is meshed with C3D10M elements. All other parts including the head, bolts, and Omega seal are meshed with C3D8R elements. The mesh is shown in Figure 1.1.23–3.

Material model

The elastic model is used in all models with a Young's modulus of 2.07×10^{11} N/m^2 (3.0×10^7 lbf/in^2) and a Poisson's ratio of 0.29.

Boundary conditions

Symmetric boundary conditions are applied to the two side surfaces of the vessel quarter. The nodes on the centerline are constrained separately and are free to move only along the vessel central axis. The node located at the center of the vessel bottom surface is pinned to give the model a statically determinant condition.

Loading

A pre-tension load of 2200 kN (5×10^6 lbf) is applied to each stud bolt in the model. The inner surfaces of the head, the vessel body, and the nozzle are subject to a constant pressure of 1.38×10^7 Pa (2000 psi) from the water.

Bolting of the CDM plug to the closure head standpipe, defined as the CDM assembly, is simulated in this analysis by the application of a pair of concentrated loads acting through distributing coupling constraints. Refer to Figure 1.1.23–4 for identification of the loaded regions of each CDM assembly. For each CDM plug one coupling constraint acts on the top surface of the plug (region B). For each standpipe one coupling constraint acts across the top surface (region C). The reference node for each coupling constraint is positioned along the center axis of the CDM assembly so that vertical concentrated loads can be applied without generating overturning moments. A bolting force of 106 kN (2.4×10^5 lbf) for each CDM assembly is chosen as adequate to overcome the liftoff force due to the vessel internal pressure. This force is applied vertically in the up direction to the standpipe coupling constraint. A downward force is applied to the accompanying coupling constraint on the CDM plug, but this force is lessened by the amount of the pressure-generating liftoff force due to the operating pressure acting on region A (shown in Figure 1.1.23–4), since the capping of this region is not considered explicitly in the model. Based on the diameter of region A, this pressure liftoff force equals 99 kN (2.26×10^5 lbf).

Constraints

The Omega seal is tied to the flange surfaces of the vessel head and the vessel body. As mentioned above, distributing coupling constraints are applied to the CDM plugs and the closure head standpipes.

Interactions

Small-sliding contact definitions are prescribed between

- the reactor vessel and closure head,
- the lower nuts and the reactor vessel bolting flange,
- the upper nuts and the closure head bolting flange, and
- the CDM plug and the accompanying seating ledge in the closure head.

Analysis steps

The analysis is performed in a single static step with automatic stabilization to help establish the contact between the stud bolts, head, seal, and vessel. Results show that the static dissipation energy is minimal compared to the strain energy; therefore, its effect on the response can be neglected.

Output requests

Default field and history output requests are specified in the step.

Results and discussion

This case is provided as a reference. Submodel analysis results are compared to these results in the discussion of Case 2 and Case 3 below.

Case 2 Submodel analysis of the closure head standpipe region

This case is representative of the most common submodel analysis approach: a global analysis of a coarse source model followed by a detailed submodel analysis representing a smaller region of the source model. Here, the coarse source model excludes details of the CDM plug and closure head standpipe as an illustration of a source model with significant defeaturing—a common motivation for subsequent submodel analysis. The submodel comprises a portion of the closure head and two CDM plugs, using a finer mesh and with feature details included. The relation between the source model and submodel is shown in Figure 1.1.23–5.

Analysis types

A static stress analysis is performed.

Mesh design

In the source model the vessel and closure head are meshed with C3D20R elements; all the other parts including the head, bolts, and Omega seal are meshed with C3D8R elements. The source model mesh is shown in Figure 1.1.23–6.

For the submodel the CDM plug is meshed with C3D8R elements and the closure head is meshed with C3D10M elements. The submodel mesh is shown in Figure 1.1.23–7.

Material model

The material model is the same as in Case 1.

Boundary conditions

The source model boundary conditions reflect those applied in Case 1. Similarly, the submodel has symmetric boundary conditions applied to the two side surfaces of the closure head. In the node-based submodel analysis, submodel boundary conditions are applied to the submodel boundary. In the surface-based submodel analysis, a boundary condition is applied in the 2-direction on the coupling constraint for each CDM plug in the closure head to suppress the rigid body mode.

Loading

The source model loads reflect those applied in Case 1 except for the bolting of the CDM plugs to the closure head, which is introduced in the submodel.

In the surface-based submodel analysis, submodel distributed loads are applied to the submodel boundary surface.

Constraints

Distributing coupling constraints are applied to the CDM plugs and the closure head standpipes, as in Case 1.

Interactions

Contact interactions are the same as in Case 1.

Analysis steps

The global analysis of the source model is performed in a single static step with automatic stabilization to help establish the contact between the stud bolts, head, seal, and vessel. The submodel analysis is performed in a single static step.

Output requests

Default field and history output requests are specified in the step.

Results and discussion

Mises stress results are compared along paths in two regions in the closure head:

- the first comparison is made along a ligament through the closure head shell, as shown in Figure 1.1.23–8; and
- the second comparison is made along a circular path in the vicinity of the CDM hardware seating ledge, as shown in Figure 1.1.23–9.

By reviewing the stress distribution comparisons (discussed below), you can see that in this case, surface-based submodeling is superior to node-based submodeling for results lying within the main closure head shell. In the upper region of the standpipe, neither method provides adequate results indicating that the level of defeaturing in the source model is too great for an accurate submodel analysis of the standpipe region.

Closure head shell ligament

Figure 1.1.23–10 compares the Mises stress distribution on the path shown in Figure 1.1.23–8 for the reference model, the node-based submodel solution, and the surface-based submodel solution. These results show that the surface-based submodel solution provides a more accurate stress distribution than the node-based submodel technique in this region. This result is consistent with the guidelines documented in "Surface-based submodeling" in "Submodeling: overview," Section 10.2.1 of the Abaqus Analysis User's Guide, namely that a surface-based solution is more accurate in cases where the environment is load controlled—the vessel pressurization dominates the closure head response in the shell region—and the submodel geometry differs from the source model geometry—the source model does not include the standpipe detail.

In practice, the classification of an analysis according to these guidelines, particularly the classification of load-controlled vs. displacement-controlled, is often not obvious nor is the reference solution available for comparison. Therefore, you should always compare measures of interest between

the source model and the submodel on or near the submodel driven boundary and confirm that they show reasonable agreement. In this case the Mises stress is compared along a path, shown in Figure 1.1.23–7, cutting across the submodel driven boundary. In Figure 1.1.23–11 the results comparison shows that the surface-based submodel solution provides a stress distribution on the submodel boundary that more closely matches that for the global solution of the source model. This plot also shows the reference solution, which shows better agreement with the surface-based submodel solution at the outer edge of the shell and better agreement with the node-based submodel solution at the inner edge of the shell. Hence, the agreement between global model and submodel stress distributions, while necessary, is not sufficient to confirm an adequate submodel solution at all locations. You must also use judgment as to whether geometric differences are too great between the source model and submodel.

Standpipe seating ledge

Figure 1.1.23–12 compares the Mises stress distribution on the path shown in Figure 1.1.23–9 for the reference model, the node-based submodel solution, and the surface-based submodel solution. The stress results near the seating ledge show that neither submodeling technique is clearly superior or provides adequate accuracy. This follows from the fact that the standpipe and seating ledge region did not appear at all in the source model analysis; the defeaturing was too severe in this case for an adequate submodel solution in this region.

This seating ledge stress comparison makes it clear that although a favorable comparison of results on the submodel boundary, as was done in Figure 1.1.23–11, is necessary, it is not sufficient to ensure adequate submodel results in all locations in the model. In this case the seating ledge region was absent entirely from the defeatured source model, and you should not expect accurate results in this region.

Case 3 Submodel application of CDM plug loading

This case represents an atypical use of submodeling in which the source model is associated with a small part of the structure and the submodel comprises most of the overall structure. Here, the source model focuses on the CDM plugs to predict how each of the CDM plugs loads the closure head.

The subsequent submodel analysis uses results from the source analysis for loading the remainder of the structure. The regions considered for the source model and submodel are shown in Figure 1.1.23–13.

Mesh design

In the source model the CDM plug hardware is meshed with C3D8R elements. The source model mesh is shown in Figure 1.1.23–13. The geometry for the remaining structure is also shown in this figure to illustrate the plug positioning relative to the overall reactor assembly.

The submodel mesh is nearly identical to that shown in Figure 1.1.23–3 for Case 1. The only difference is that the CDM plugs are excluded from the model in Case 3.

Boundary conditions

The submodel analysis of the CDM plug source model simulates contact with the seating ledge with a boundary condition constraint on the plug seating surface.

Loading

The loading follows that for Case 1 with the application of the loads split between the source model and submodel.

Source model analysis

The bolt load applied to the CDM plug is simulated through a downward concentrated force applied to a distributing coupling constraint reference node in each of the CDM plugs. The magnitude of this force is the bolting force of 106 kN (2.4×10^5 lbf) less the pressure generating liftoff force of 99 kN (2.26×10^5 lbf), for the reasons detailed in the Case 1 description.

Submodel analysis

A pre-tension load of 2200 kN (5×10^6 lbf) is applied to each stud bolt in the model. The inner surfaces of the head, the vessel body, and the nozzle are subject to a constant pressure of 1.38×10^7 Pa (2000 psi) from the water.

The bolt load applied to the standpipe is simulated through an upward concentrated force applied to a distributing coupling constraint reference node in each of the standpipes. The magnitude of this force is the bolting force of 106 kN (2.4×10^5 lbf) less the pressure generating liftoff force of 99 kN (2.26×10^5 lbf).

Constraints

All constraint definitions are the same as in Case 1.

Interactions

All interaction definitions are the same as in Case 1, except that the contact interaction between the CDM plug and the standpipe seating ledge is effected through submodel loads and boundary conditions.

Run procedure

Run the analyses with the input files listed for Case 3 below.

Results and discussion

Stress results are considered on the same paths defined for comparison of reference and submodel results in Case 2. The Mises stress distribution on these paths is compared in Figure 1.1.23–14 and Figure 1.1.23–15 for the two forms of submodeling and the reference solution.

These results show that in both the high-stressed region, shown in the ligament stress plot, and in the vicinity of the seating ledge, the surface-based submodeling approach provides a more accurate solution. The poor results for node-based submodeling follow from the fact that the assembly model—the submodel in this case—elongates along the vessel main axis. The CDM assembly region experiences this elongation as a rigid body translation. The standpipe seating ledge, however, is constrained in its movement by the submodel boundary conditions. These boundary conditions follow from the separate

analysis of the CDM plug source model that does not consider the solution-dependent elongation of the vessel assembly.

Discussion of results and comparison of cases

Case 2 and Case 3 illustrate situations where you may see improved accuracy when using the surface-based submodeling approach.

The effect of stiffness change on submodel analysis

In cases where the submodel stiffness matches that of the source model, you can expect, using reasonable modeling practices, that the submodel analysis will provide adequate results. In cases where the submodel stiffness differs, such as in Case 2, you must exercise caution in evaluating your submodel solution. Comparison of stress contours on the common boundary of the source model and submodel can aid you in determining if your solution is adequate. In the case of significant defeaturing, you should not rely on the submodeling analysis technique in any form for detailed stress response in areas absent from the source model, such as the closure head standpipe.

The effect of displacement discrepancies on submodel analysis

In cases where you expect that the submodel displacement solution will differ from the corresponding source model solution by only a rigid body motion, such as in Case 3, you can expect that a node-based submodeling approach will give incorrect results. In this case you can use the alternative surface-based submodeling of stresses and obtain improved solution accuracy.

Files

Case 1 Reactor closure analysis: reference solution

ReactorHead_reference.inp	Input file to analyze the reactor vessel closure assembly.

Case 2 Submodeling of the closure head standpipe region

ReactorHead_global.inp	Global analysis of the reactor vessel closure assembly source model with a defeatured closure head.
ReactorHead_submodel_node.inp	Closure head submodel analysis using node-based submodeling.
ReactorHead_submodel_surface.inp	Closure head submodel analysis using surface-based submodeling.

Case 3 Submodel application of CDM plug loading

ReactorHead_CDMdetail.inp	Global analysis of the CDM plug source model.
ReactorHead_assembly_node.inp	Reactor vessel closure assembly submodel analysis with CDM loading effected through node-based submodeling.

ReactorHead_assembly_surface.inp	Reactor vessel closure assembly submodel analysis with CDM loading effected through surface-based submodeling.

References

Abaqus Analysis User's Guide

- "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

Abaqus Keywords Reference Guide

- *BOUNDARY
- *DSLOAD
- *SUBMODEL

Other

- Naval Reactors Branch, Division of Reactor Development, United States Atomic Energy Commission, *The Shippingport Pressurized Water Reactor,* Reading, Massachusetts: Addison Wesley Publishing Company, 1958.

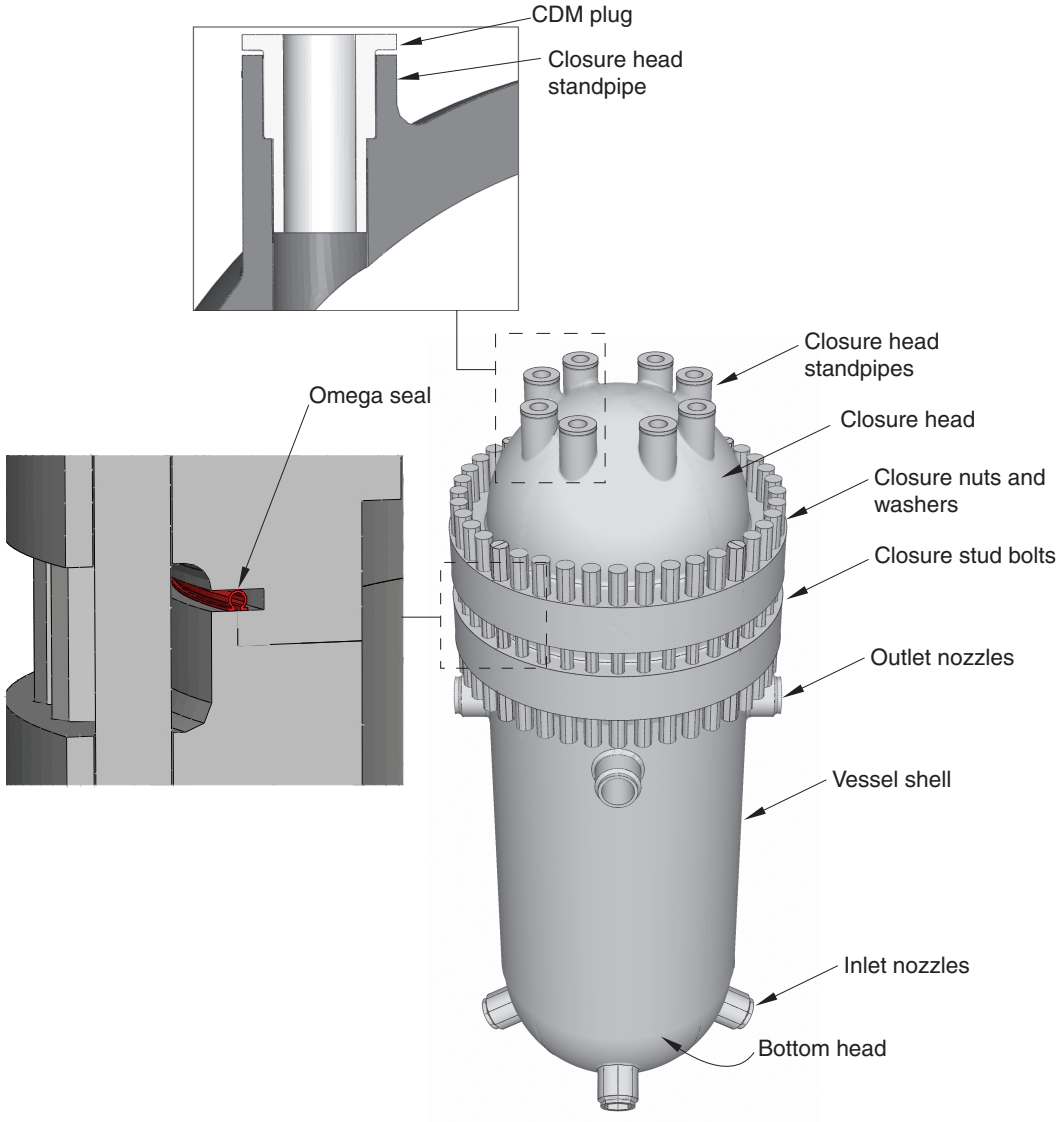


Figure 1.1.23–1 Reactor vessel assembly.

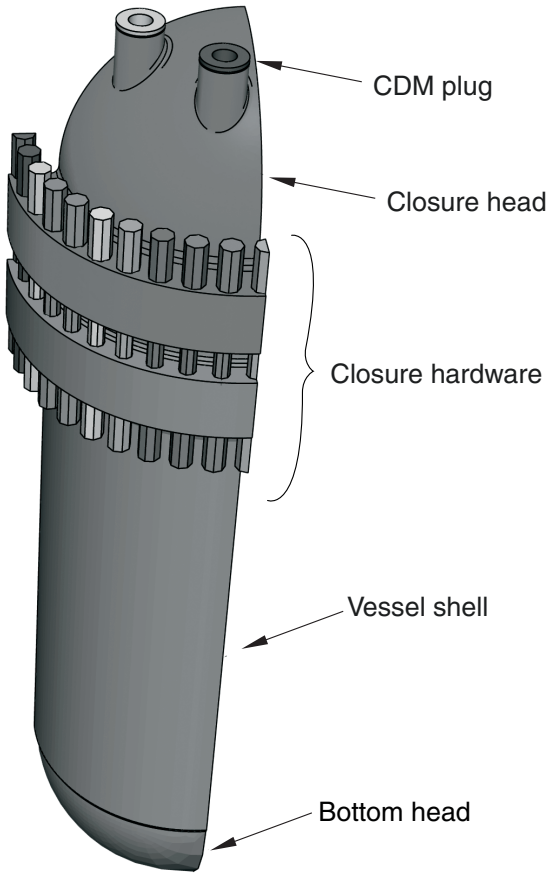


Figure 1.1.23–2 Reactor vessel assembly model.

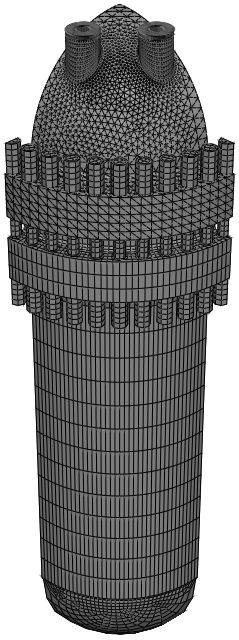


Figure 1.1.23–3 Reference analysis mesh.

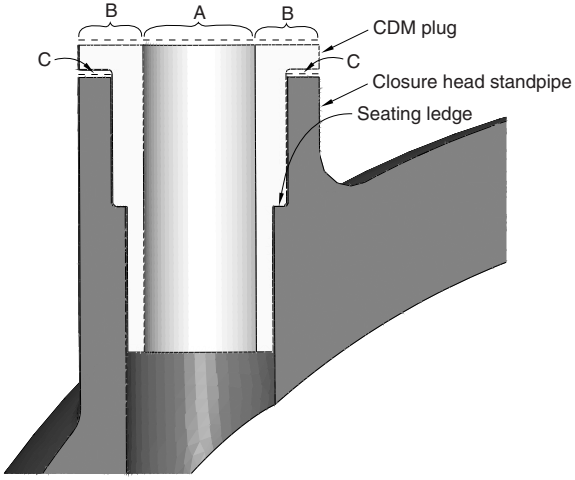


Figure 1.1.23–4 Load application areas on the CDM plug and closure head standpipe.

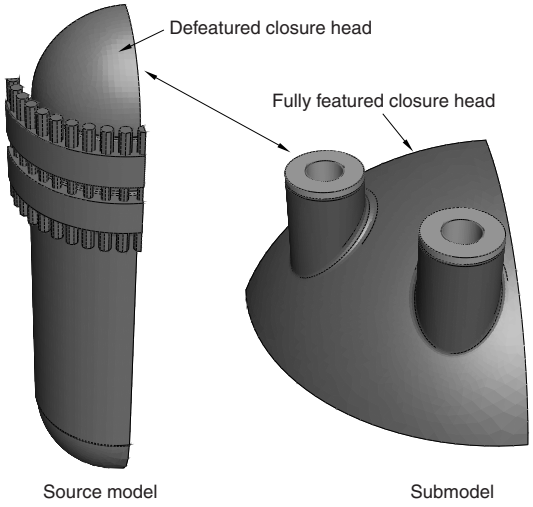


Figure 1.1.23–5 Case 2 closure head submodel relation to the source model.

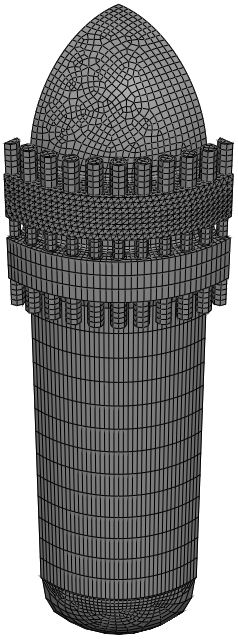


Figure 1.1.23–6 Case 2 global analysis mesh with defeatured closure head.

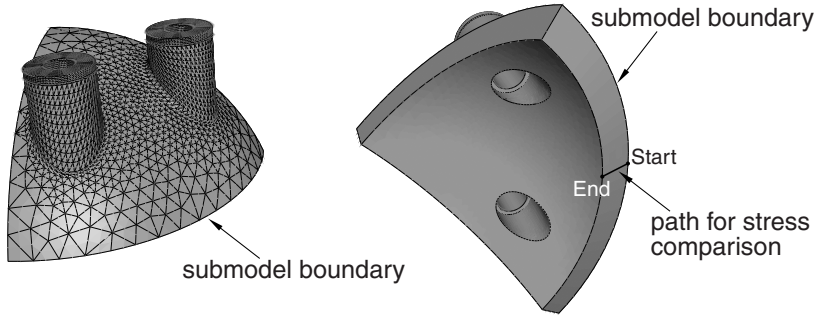


Figure 1.1.23–7 Case 2 submodel analysis showing the mesh (left) and the path definition for stress comparison to the source model (right).

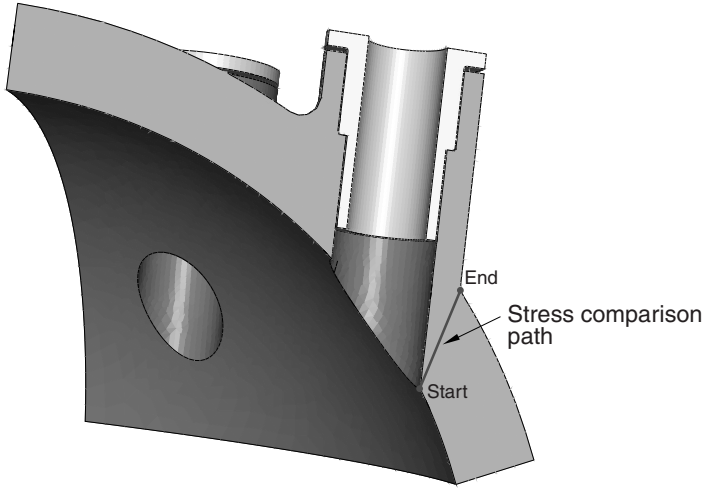


Figure 1.1.23–8 Through-ligament path definition.

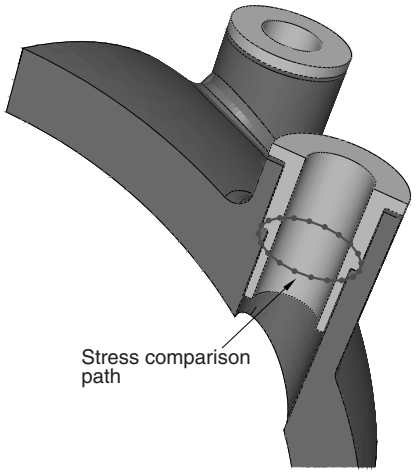


Figure 1.1.23–9 Seating ledge path definition.

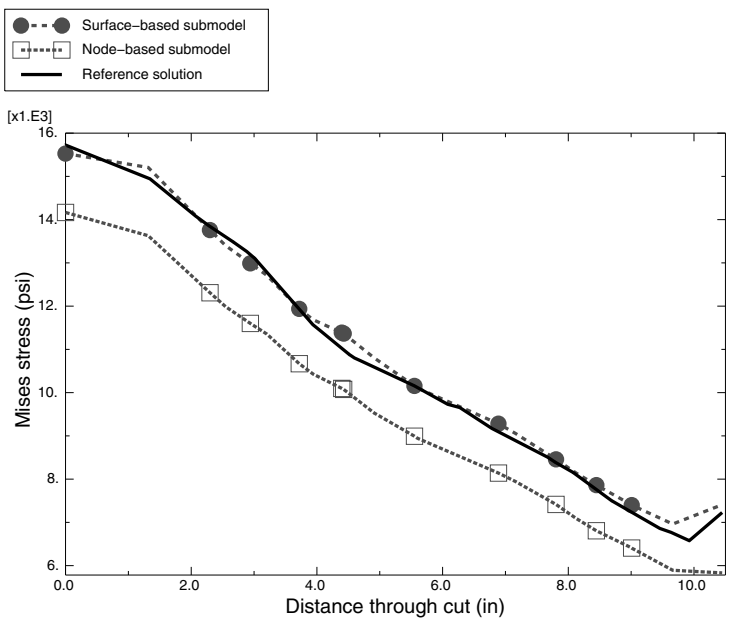


Figure 1.1.23–10 Case 2 stress distribution comparison through the closure head ligament.

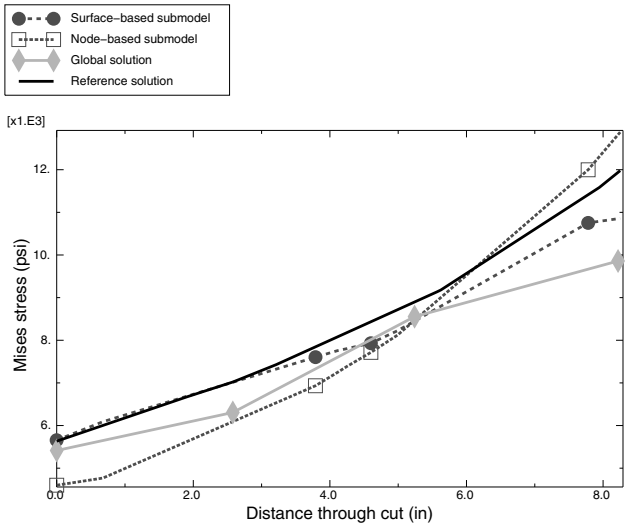


Figure 1.1.23–11 Comparison of the closure head through-thickness Mises stress distribution at the location of the submodel boundary.

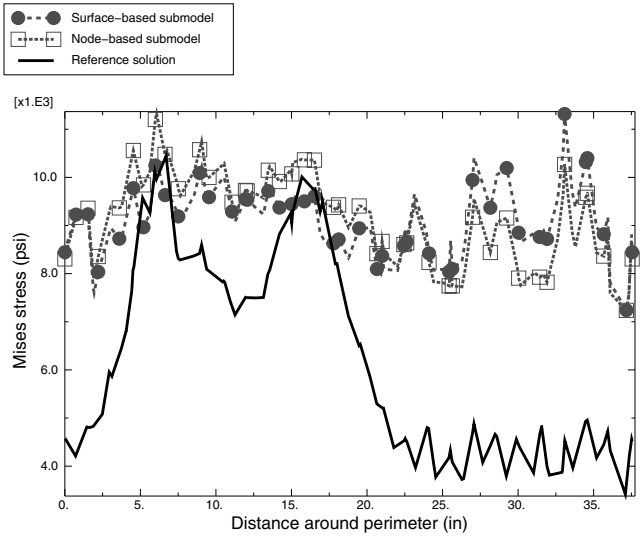


Figure 1.1.23–12 Case 2 stress distribution comparison around the seating ledge.

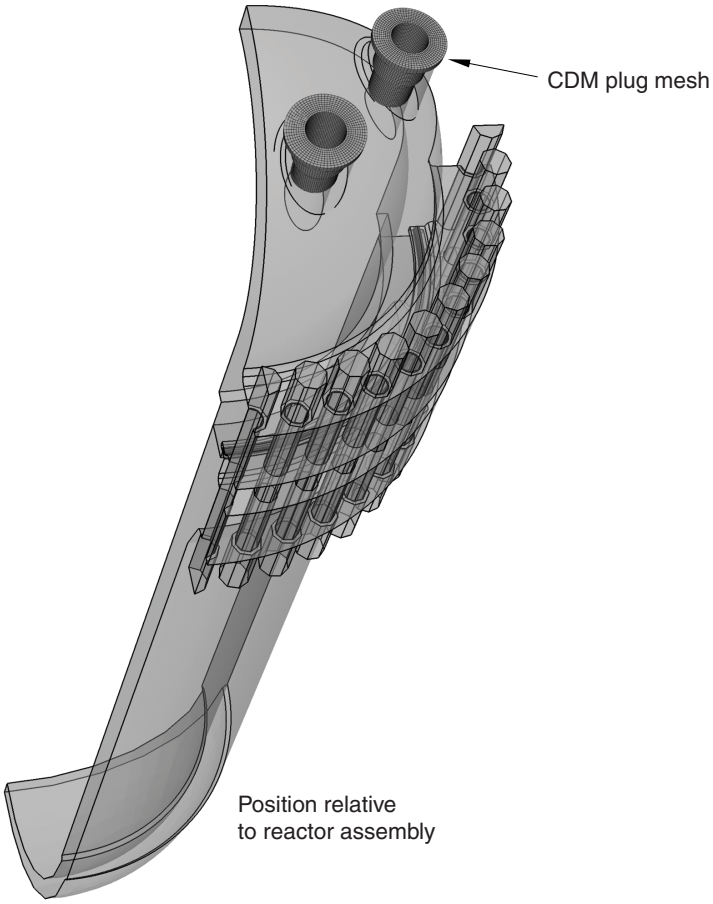


Figure 1.1.23–13 Case 3 CDM plug analysis mesh.

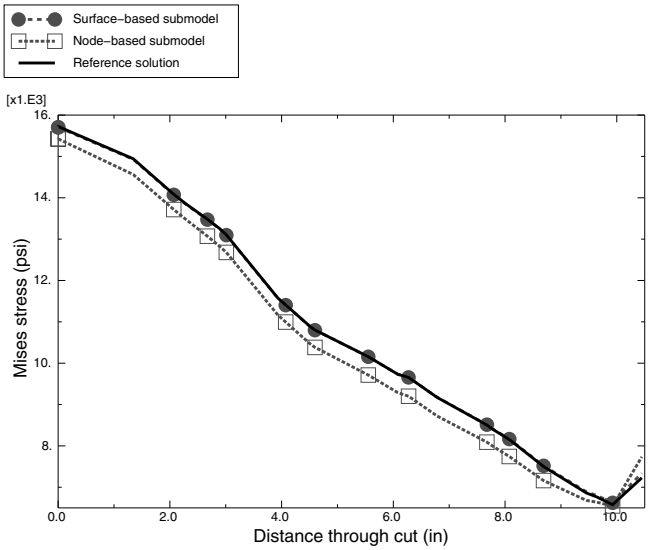


Figure 1.1.23–14 Case 3 stress distribution comparison through the closure head ligament.

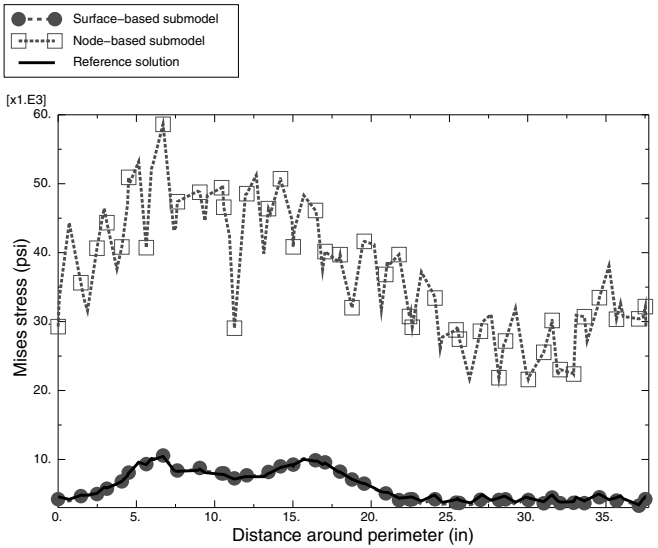


Figure 1.1.23–15 Case 3 stress distribution comparison around the seating ledge.

1.1.24 USING A COMPOSITE LAYUP TO MODEL A YACHT HULL

Products: Abaqus/Standard Abaqus/CAE Abaqus/Viewer

Objectives

This example problem demonstrates the following Abaqus features and techniques:

- importing the shell geometry of a yacht hull from an ACIS (`.sat`) file,
- creating a composite layup using Abaqus/CAE,
- applying plies in the layup to regions of the model,
- viewing a ply stack plot from a region of the model,
- viewing an envelope plot that shows the critical plies in each region of the model, and
- viewing an *X*–*Y* plot through the thickness of an element.

Application description

Composite hulls are used routinely in the yacht industry. Composite materials allow manufacturers to create high-performance marine vessels that incorporate the complex hull shapes that engineers have derived from computational fluid dynamics analyses and from experimental testing. Composites also provide the strength, rigidity, and low mass that high-performance yachts require. However, incorporating many layers of material with varying orientations in a complex three-dimensional finite element model can be time consuming. The addition of local reinforcements complicates the process. These issues are described by Bosauder et al. (2006).

The composite layup capability in Abaqus/CAE simplifies the process of composites modeling by mirroring the procedure that manufacturers follow on the shop floor—stacking sheets of composite material in a region of a mold and aligning the material in a specified direction. The Abaqus/CAE composite layup editor allows you to easily add a ply, choose the region to which it is applied, specify its material properties, and define its orientation. You can also read the definition of the plies in a layup from data in a text file, which is convenient when the data are stored in a spreadsheet or are generated by a third-party tool.

Geometry

Figure 1.1.24–1 shows the hull, mast, rigging, and keel of the yacht model. The geometry of the model is imported as a single part from an ACIS (`.sat`) file, as shown in Figure 1.1.24–2. The part models one half of the hull, and symmetric boundary conditions are applied. The hull represents a high-performance 20-meter yacht with reinforced bulkheads that stiffen the structure. The infrastructure above the deck does not play a role in modeling the performance of the hull and is not included in the model.

Sets are created that correspond to the regions of the composite layup to which plies are applied.

Materials

The model is partitioned into 27 regions. Each region contains plies of glass-epoxy cloth surrounding a Nomex core. Most regions contain nine plies—four glass-epoxy plies on either side of the Nomex core. However, additional plies are added to reinforce regions of high strain. Some bulkheads are reinforced with stringers made of glass-epoxy cloth with an effective Young's modulus of 128000 N/mm^2. Table 1.1.24–1 shows the material properties of the glass-epoxy cloth, and Table 1.1.24–2 shows the material properties of the Nomex core.

Figure 1.1.24–3 shows several rows of the composite layup table and illustrates how plies and material orientations are assigned to a region of the model. Figure 1.1.24–4 shows a ply stack plot of the same region.

Boundary conditions and loading

The center of the model is constrained to be symmetric about the *y*-axis, as shown in Figure 1.1.24–2. The following loads are applied:

- A hydrostatic pressure is applied to the hull. The pressure is modeled with an analytical field that increases the pressure linearly along the *z*-axis.
- Concentrated forces that model the tension from the sail rigging are applied to the front, rear, and side of the deck. The forces are applied along the *x*-axis of a datum coordinate system. Each coordinate system has an origin at the location of the load, and the *x*-axis orients the load toward the location of the top of the mast. The concentrated forces are transferred to the deck through distributing couplings.
- The load from the mast is applied at the base of the hull in the *z*-direction.
- The keel is modeled with a lumped mass attached to the hull through a kinematic coupling.
- An inertia relief load is applied at the center of the hull to bring the model into equilibrium after the loads are applied.

Abaqus modeling approaches and simulation techniques

A single loading case is considered that uses a static analysis to study the effect of the loading on the composite layup.

Mesh design

The model is meshed by Abaqus/CAE using the free meshing technique and quadrilateral-dominated elements.

Loading

- The tension load from the sail rigging is 5500 N at the front of the hull, 4000 N at the rear of the hull, and 7500 N at the side of the hull.
- The load from the mast is 17500 N.
- A lumped mass of 10 metric tons models the keel.

Constraints

A kinematic coupling transfers the weight of the keel to the base of the hull, and three distributing couplings transfer the load from the rigging to the hull.

Analysis steps

A single static load step is defined for the analysis; nonlinear effects are not included.

Output requests

By default, Abaqus/CAE writes field output data from only the top and bottom section points of a composite layup, and no data are generated from the other plies. In this model, output is requested for all section points in all plies. This allows you to create an envelope plot of the entire model that indicates which plies in each region are carrying the highest strain.

Results and discussion

Figure 1.1.24–5 shows an envelope plot of the in-plane shear strain (E12) in the middle of the hull. Figure 1.1.24–6 shows the through-thickness variation of this strain component.

Files

You can use the Abaqus/CAE Python scripts to create the model and to run the analysis. You can also use the Abaqus/Standard input file to run the analysis.

compositehull_model.py	Script to create the model using the geometry from compositehull_geometry.sat and the composite layup from compositehull_layup.txt.
compositehull_geometry.sat	ACIS file containing the geometry of the model.
compositehull_layup.txt	A comma-separated text file defining the plies in the composite layup.
compositehull_job.py	Script to analyze the model.
compositehull_job.inp	Input file to analyze the model.

References

Abaqus Analysis User's Guide

- "Shell elements," Section 29.6 of the Abaqus Analysis User's Guide

Abaqus/CAE User's Guide

- "Creating composite layups," Section 12.4.4 of the Abaqus/CAE User's Guide
- Chapter 23, "Composite layups," of the Abaqus/CAE User's Guide

Other

- Bosauder, P., D. Campbell, and B. Jones, "Improvements in the Commercial Viability of Finite Element Analysis (FEA) for Accurate Engineering of Marine Structures," JEC conference, Paris, March 2006.

Table 1.1.24–1 Material properties of the glass-epoxy cloth.

Variable	Value
E_1	35000 N/mm^2
E_2	7500 N/mm^2
ν_{12}	0.3
G_{12}	3600 N/mm^2
G_{13}	3000 N/mm^2
G_{23}	3000 N/mm^2
ρ	1.5×10^{-9} metric tons/mm^3

Table 1.1.24–2 Material properties of the Nomex core.

Variable	Value
E_1	10 N/mm^2
E_2	10 N/mm^2
ν_{12}	0.3
G_{12}	1 N/mm^2
G_{13}	30 N/mm^2
G_{23}	30 N/mm^2
ρ	8.0×10^{-11} metric tons/mm^3

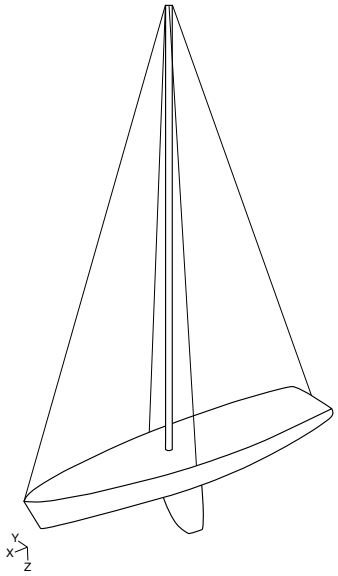


Figure 1.1.24–1 The yacht model.

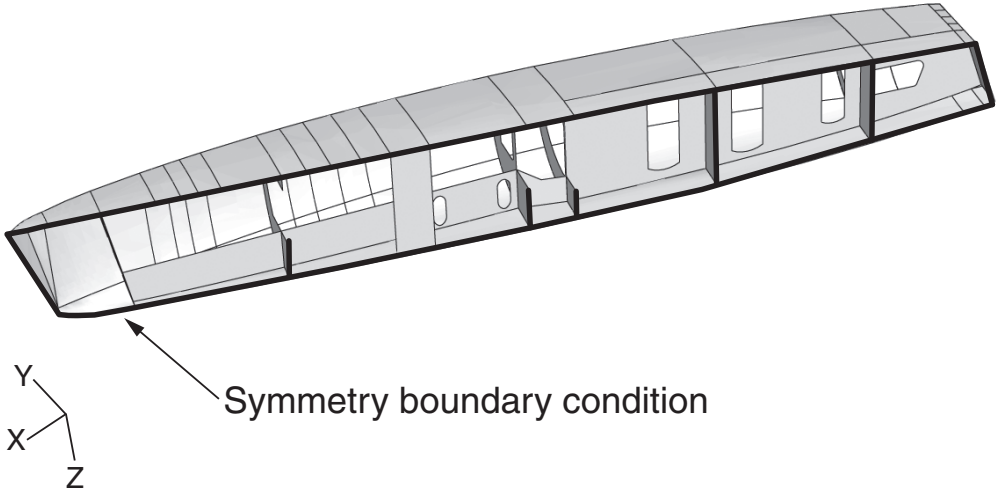
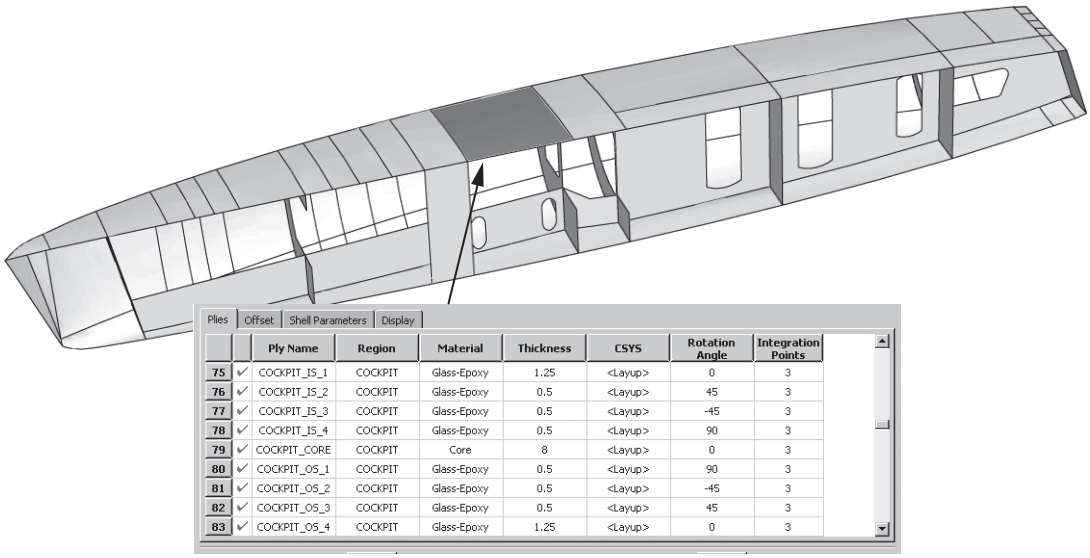


Figure 1.1.24–2 The symmetric model.



Plies | Offset | Shell Parameters | Display

		Ply Name	Region	Material	Thickness	CSYS	Rotation Angle	Integration Points
75	✓	COCKPIT_IS_1	COCKPIT	Glass-Epoxy	1.25	<Layup>	0	3
76	✓	COCKPIT_IS_2	COCKPIT	Glass-Epoxy	0.5	<Layup>	45	3
77	✓	COCKPIT_IS_3	COCKPIT	Glass-Epoxy	0.5	<Layup>	-45	3
78	✓	COCKPIT_IS_4	COCKPIT	Glass-Epoxy	0.5	<Layup>	90	3
79	✓	COCKPIT_CORE	COCKPIT	Core	8	<Layup>	0	3
80	✓	COCKPIT_OS_1	COCKPIT	Glass-Epoxy	0.5	<Layup>	90	3
81	✓	COCKPIT_OS_2	COCKPIT	Glass-Epoxy	0.5	<Layup>	-45	3
82	✓	COCKPIT_OS_3	COCKPIT	Glass-Epoxy	0.5	<Layup>	45	3
83	✓	COCKPIT_OS_4	COCKPIT	Glass-Epoxy	1.25	<Layup>	0	3

Figure 1.1.24–3 Assigning plies in the layup table to a region of the model.

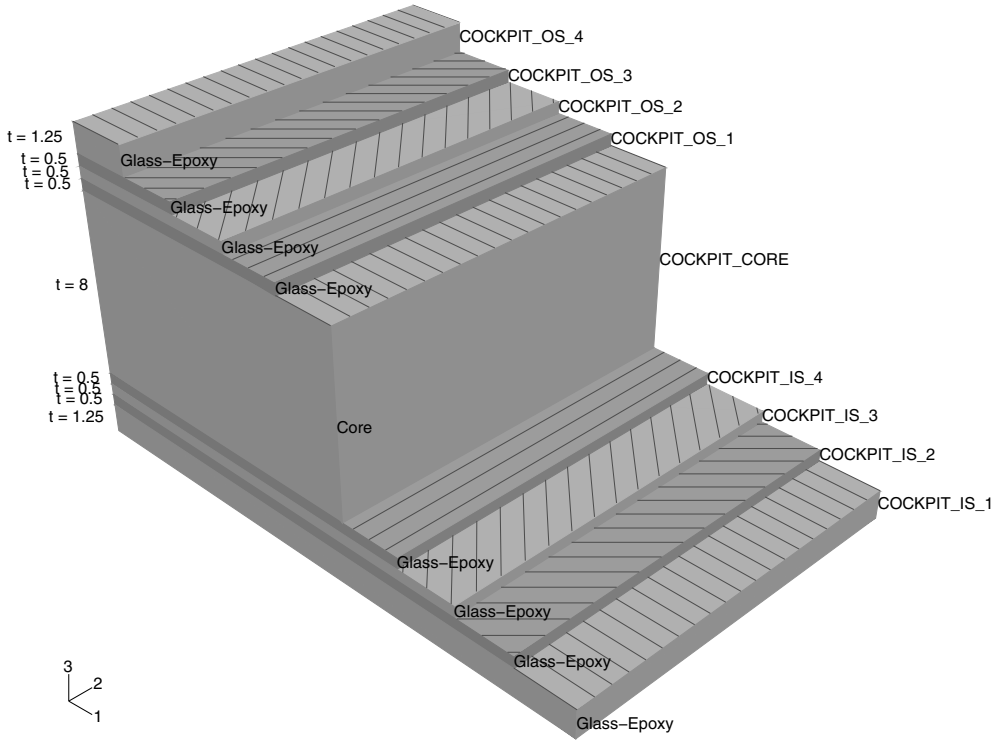


Figure 1.1.24–4 A ply stack plot from the cockpit.

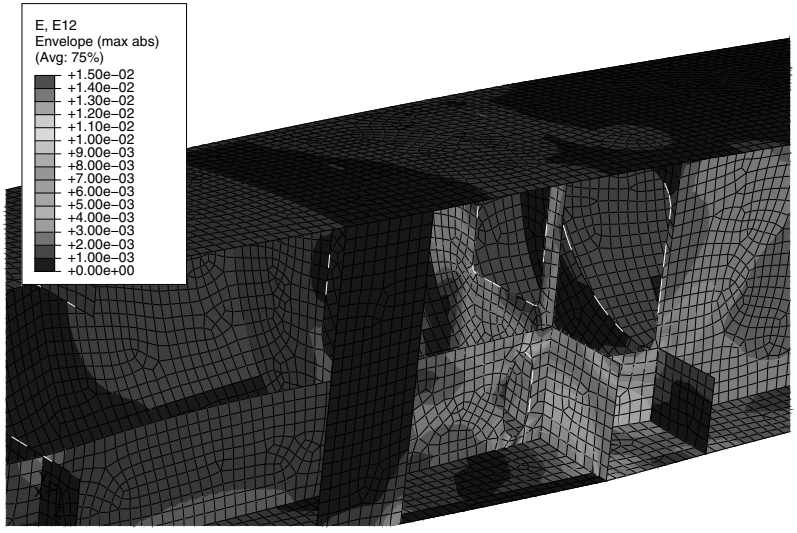


Figure 1.1.24–5 Envelope plot of strain in the critical plies in the center of the hull.

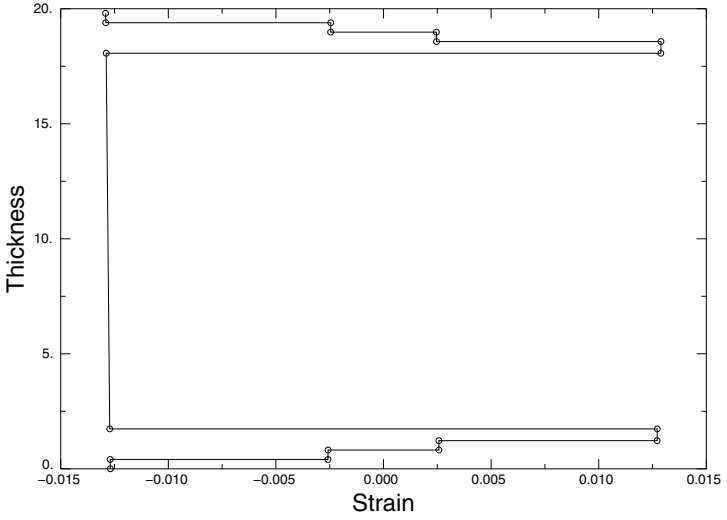


Figure 1.1.24–6 Strain (E12) across the thickness of an element.

1.1.25 ENERGY COMPUTATIONS IN A CONTACT ANALYSIS

Product: Abaqus/Standard

Objectives

This example illustrates using energy output in a static nonlinear elastic, finite-sliding, frictional contact analysis to account for the following output:

- frictional dissipation,
- elastic contact energy, which represents energies stored in penalty springs of contact constraints and "softened" contact constraints; and
- the remaining work done by contact forces that is not accounted for by other energy output variables.

Application description

Energy output can provide valuable insight into model behavior and increase confidence in numerical accuracy. Large relative displacements on a contact interface within a time increment can lead to significant energy contributions that are not always intuitive. In this problem contact-related energy output variables are introduced and explained along with other meaningful energy output variables.

Problem setup

The model represents a clamped snapping arrowhead that passes through an opening under a wall when pushed by a prescribed x-displacement. The geometry and mesh of C3D8R elements are shown in Figure 1.1.25–1. The perimeter of the right wall is fixed. Figure 1.1.25–2 shows an intermediate position with the structure visibly bent when the arrowhead is in contact while passing through the opening. Figure 1.1.25–3 shows that some stress remains at the end of simulation, which is related to some inaccuracy in the integration of the constitutive behavior. Ideally, the final position should be totally stress and strain free.

Abaqus modeling approaches and simulation techniques

General contact uses penalty enforcement of contact constraints in the normal contact direction by default. Frictional constraints in the tangential directions are also enforced by default with the penalty method. This penalty method can be thought of as inserting normal and tangential springs in the model while contact is active. Such springs store elastic energy that is released when contact opens up. The contact constraint elastic energy in the normal direction (ALLCCEN) and the contact constraint elastic energy in tangential directions (ALLCCET) represent nonnegative sums of the energies in all such active "contact springs" (or "penalty springs") in the model. The energy output variable ALLCCE is their sum: ALLCCE = ALLCCEN + ALLCCET. The use of "softened" contact relationships (see "Contact pressure-overclosure relationships," Section 37.1.2 of the Abaqus Analysis User's Guide)

can also contribute energy to ALLCCEN. There is no contact constraint elastic energy associated with directly enforced hard contact and Lagrange friction.

The incremental frictional dissipation (ALLFD) together with incremental ALLCCET represents the physical incremental work of the frictional contact forces. Both ALLFD and ALLCCET remain zero in frictionless problems.

The elastic contact energy output variables (ALLCCEN, ALLCCET, and ALLCCE) are computed in total form every time increment. Other energies are computed incrementally, with the incremental contribution added to the running totals each increment.

To understand what the contact constraint discontinuity work (ALLCCDW) represents, first consider the full incremental work of contact forces on incremental displacements, which may produce incremental contributions to the following three contact energy variables:

- Frictional dissipation (ALLFD)
- Recoverable energy stored in contact penalty springs and "softened" contact constraints (ALLCCE)
- Stabilization dissipation associated with contact (ALLCCSD), not present in this example

Output variable ALLCCDW accounts for the remaining work done by contact forces; i.e., "full contact work" is equal to ALLFD + ALLCCE + ALLCCSD – ALLCCDW. ALLCCDW can be positive or negative, and it may be counterintuitive. For example, frictionless hard contact cannot physically store energy on the interface because normal contact forces are always orthogonal to incremental tangential displacements. Numerically, such contact constraints may have energy associated with them as demonstrated by nonzero ALLCCDW in this example, as discussed below. The total energy balance ETOTAL = ALLIE + ALLFD + ALLCCE – ALLWK – ALLCCDW.

Results and discussion

Figure 1.1.25–4 shows a complete set of nonzero energy output variables for this example. The most significant energies are external work (ALLWK) and contact constraint discontinuity work (ALLCCDW). In this example these output variables have different signs. If the external work were modified to be ALLWK + ALLCCDW, the major energies would be as shown in Figure 1.1.25–5. This combination of ALLWK and ALLCCDW is approximately in balance with the combination of frictional dissipation and elastic energy throughout the simulation.

The modified external work (ALLWK + ALLCCDW) is often representative of the physical external work in contact problems in terms of being equal to the sum of the stored and dissipated energies. Consider a particular contact constraint having a gap distance, g_i, in one increment and becoming closed with contact force, f_{i+1}, in the next increment (see Figure 1.1.25–6). A trapezoidal rule for integrating the work done by the contact force multiplies the average force by the relative incremental motion. In this case the resulting contribution to ALLCCDW is negative $1/2{*}g_i{*}f_{i+1}$. This energy contribution is nonphysical and would disappear in the numerics as the time increment tends to zero. When contact opens up, similar behavior happens with sign reversals. Numerical integration for ALLWK is also limited with respect to accounting accurately for sudden changes in external forces. Summing ALLWK and ALLCCDW often cancels the respective nonphysical energy contributions, and the net effect on the total energy balance (ETOTAL) is zero.

A nonzero incremental ALLCCDW contribution can also occur if the normal direction changes during a time increment that also has relative motion between contact surfaces. The first significant drop in ALLCCDW for this example (see Figure 1.1.25–4) occurs from increment 4 to increment 5, when the leading wall edge slips from the inclined arrowhead surface to the top surface, as illustrated in Figure 1.1.25–7, causing a significant change in the contact constraint normal direction.

With finer time incrementation, ALLCCDW generally becomes smaller, as demonstrated in Figure 1.1.25–8. The maximum time increment allowed with so-called large time increments is five times larger than with small time increments. Convergence of global Newton iterations is more difficult to reach with small time increments because of the model behavior near the last stage of the analysis when the arrowhead snaps to the stress-free state. The "back of the head" touches the rear edge of the wall along the way, whereas with large time increments, this touching is avoided completely. The difference is reflected by the reaction force X becoming negative for small time increments (see Figure 1.1.25–9).

Elastic contact energies are shown in Figure 1.1.25–10. They are small compared to the major energies in the model. If ALLCCEN is not small, contact penalty stiffness may need to be increased to improve solution accuracy. If ALLCCET is not small, the frictional elastic or slip tolerance may need to be decreased to increase the frictional penalty stiffness.

Ideally, ETOTAL should be exactly constant (in this example zero), but actually it is constant only approximately. The approximation is due to the fact that the recoverable strain energy (ALLIE) is computed by a modified trapezoidal rule, rather than an exact trapezoidal rule (which is not implemented due to dramatic computational performance implications). Such behavior is common in geometrically nonlinear problems, and one can only assess that the variation in ETOTAL is small compared to meaningful energies.

Nonzero initial ETOTAL will (appropriately) occur, for example, when initial stress is specified, such that the initial value of ALLIE is positive while other energies are initially zero. Nonzero initial ETOTAL also occurs when initial velocities are defined for a dynamic analysis; in this case initial kinetic energy is positive.

Convergence tolerances can also influence energy results because the energy balance depends on the balance of internal and external forces and moments in the model.

Input files

snaparrow_gc.inp	Input data for the analysis with large time increments.
snaparrow_gc2.inp	Input data for the analysis with small time increments.

Reference

Abaqus Analysis User's Guide

- "Defining general contact interactions in Abaqus/Standard," Section 36.2.1 of the Abaqus Analysis User's Guide

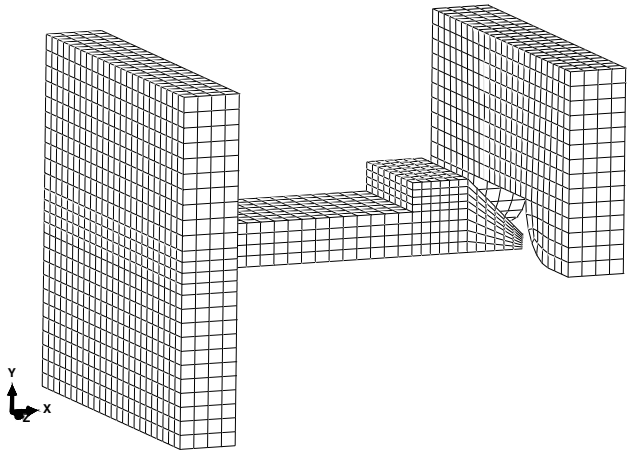


Figure 1.1.25–1 Initial position.

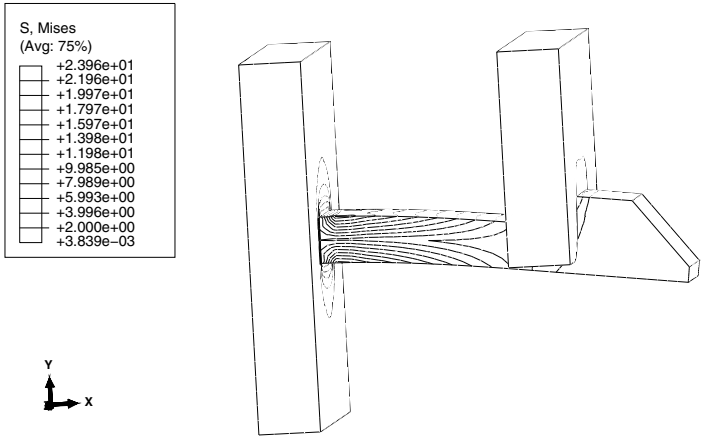


Figure 1.1.25–2 Intermediate position.

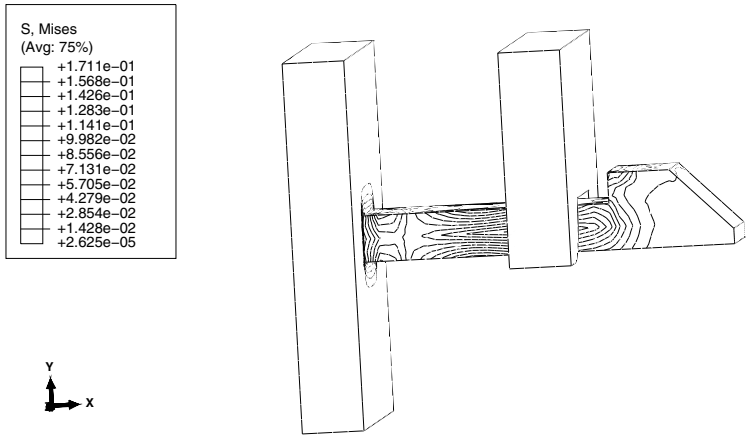


Figure 1.1.25–3 Final position.

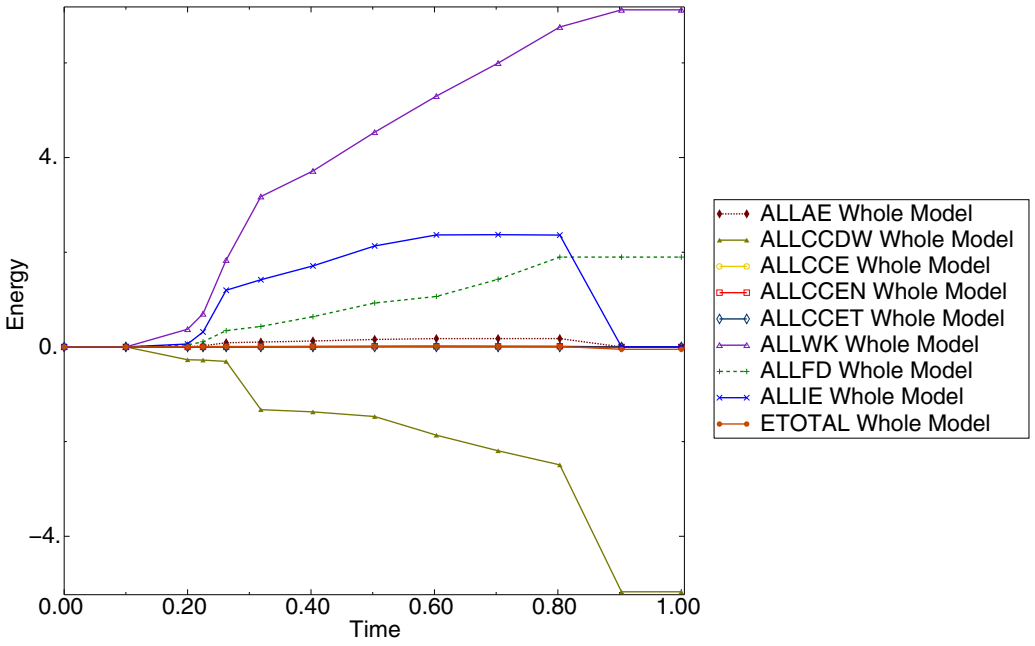


Figure 1.1.25–4 Nonzero energies in the model.

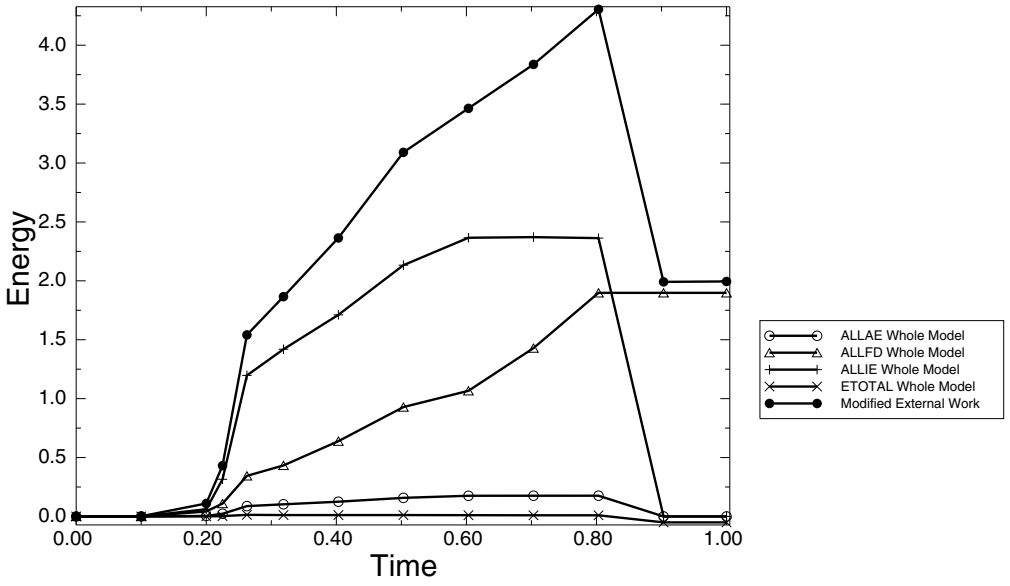


Figure 1.1.25–5 Major energies with modified external work.

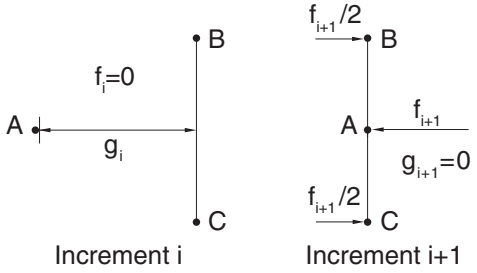


Figure 1.1.25–6 One contact point example to illustrate contributions to ALLCCDW.

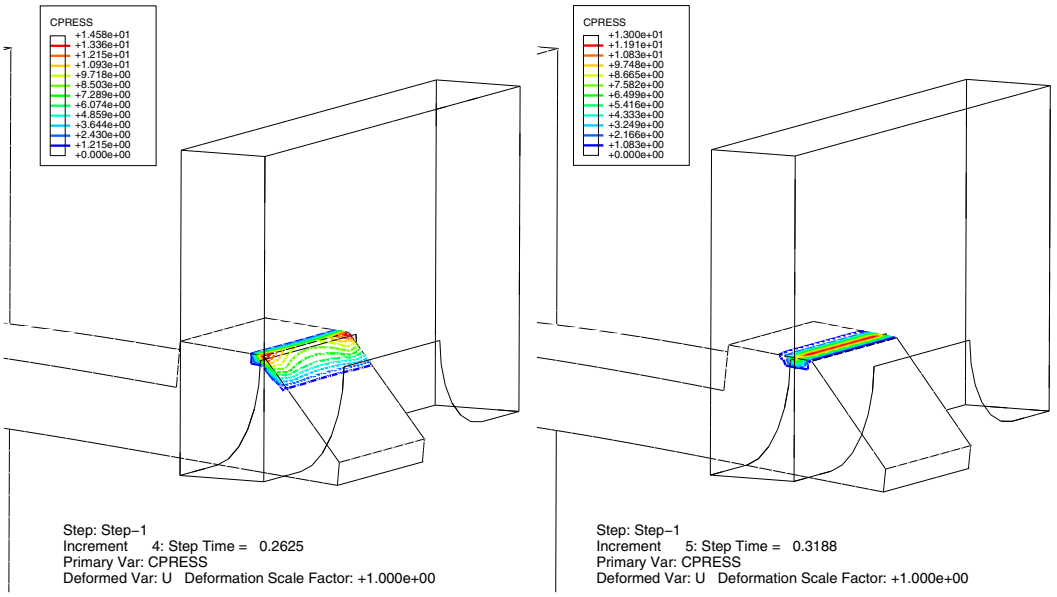


Figure 1.1.25–7 Contact normal direction change causes significant incremental change in ALLCCDW; large time increment.

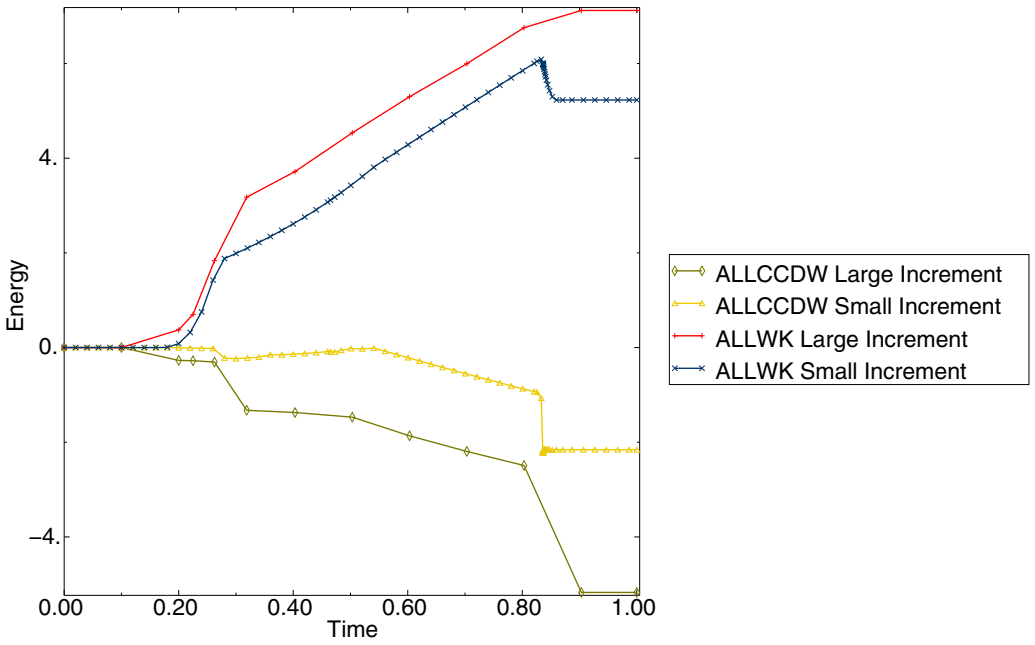


Figure 1.1.25–8 ALLCCDW and ALLWK with large and small time increments.

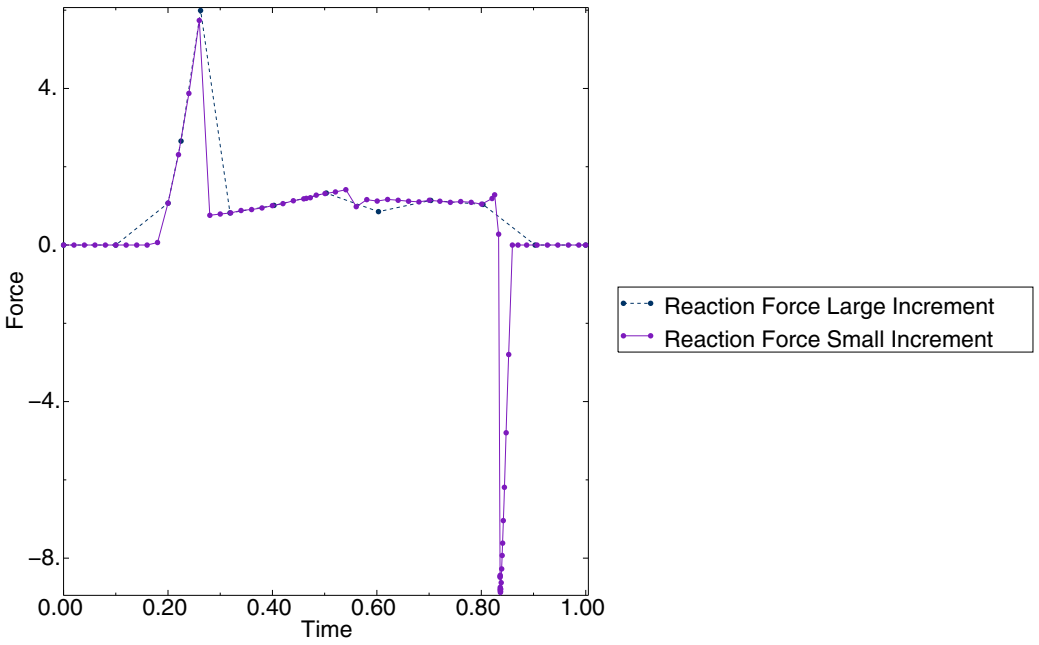


Figure 1.1.25–9 Reaction force X with large and small time increments.

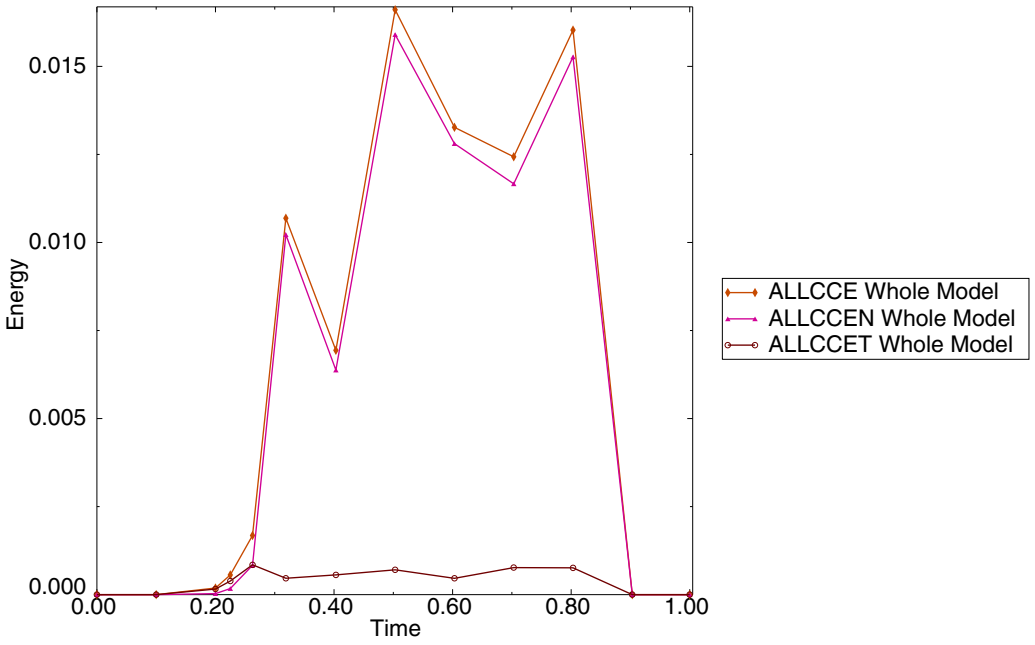


Figure 1.1.25–10 Contact elastic energies; large time increments.

1.2 Buckling and collapse analyses

- "Snap-through buckling analysis of circular arches," Section 1.2.1
- "Laminated composite shells: buckling of a cylindrical panel with a circular hole," Section 1.2.2
- "Buckling of a column with spot welds," Section 1.2.3
- "Elastic-plastic K-frame structure," Section 1.2.4
- "Unstable static problem: reinforced plate under compressive loads," Section 1.2.5
- "Buckling of an imperfection-sensitive cylindrical shell," Section 1.2.6

1.2.1 SNAP-THROUGH BUCKLING ANALYSIS OF CIRCULAR ARCHES

Product: Abaqus/Standard

It is often necessary to study the postbuckling behavior of a structure whose response is unstable during part of its loading history. Two of the models in this example illustrate the use of the modified Riks method, which is provided to handle such cases. The method is based on moving with fixed increments along the static equilibrium path in a space defined by the displacements and a proportional loading parameter. The actual load value may increase or decrease as the solution progresses. The modified Riks method implemented in Abaqus is described in "Modified Riks algorithm," Section 2.3.2 of the Abaqus Theory Guide.

The other two models illustrate the use of viscous damping. One example applies viscous damping as a feature of surface contact, which allows for the definition of a "viscous" pressure that is proportional to the relative velocity between the surfaces. The implementation of this option in Abaqus is described in "Contact pressure definition," Section 5.2.1 of the Abaqus Theory Guide. The other example applies volume proportional damping to the model. The implementation of this option is described in the automatic stabilization section of "Solving nonlinear problems," Section 7.1.1 of the Abaqus Analysis User's Guide.

Three separate cases are considered here. The first is a clamped shallow arch subjected to a pressure load. Reference solutions for this case are given by Ramm (1981) and Sharafi and Popov (1971). The second case is the instability analysis of a clamped-hinged circular arch subjected to a point load. The exact analytical solution for this problem is given by DaDeppo and Schmidt (1975). The third case is a modification of the shallow arch problem in which the ends are pinned rather than clamped and the arch is depressed with a rigid punch.

Model and solution control

The shallow circular arch is shown in Figure 1.2.1–1. Since the deformation is symmetric, one-half of the arch is modeled. Ten elements of type B21 (linear interpolation beams) are used. A uniform pressure is first applied to snap the arch through. The loading is then reversed so that the behavior is also found as the pressure is removed.

The deep circular arch is shown in Figure 1.2.1–2. One end of the arch is clamped, and the other is hinged. A concentrated load is applied at the apex of the arch. The arch undergoes extremely large deflections but small strains. Because of the asymmetric boundary conditions, the arch will sway toward the hinged end and then collapse. The arch is almost inextensible for most of the response history. Sixty elements of type B31H are used. Hybrid elements are used because they are most suitable for problems such as this.

Solution controls are used to set a very tight convergence tolerance because the problem contains more than one equilibrium path. If tight tolerances are not used, the response might follow a path that is different from the one shown.

In the Riks procedure actual values of load magnitudes cannot be specified. Instead, they are computed as part of the solution, as the "load proportionality factor" multiplying the load magnitudes given on the loading data lines. User-prescribed load magnitudes serve only to define the direction and to estimate the magnitude of the initial increment of the load for a step. This initial load increment is

the product of the ratio of the initial time increment to the time period and the load magnitudes given in the loading options. The user can terminate a Riks analysis by specifying either a maximum load proportionality factor or a maximum displacement at a node, or both. When a solution point is computed at which either of these limits is crossed, the analysis will stop. In any event, or if neither option is used, the analysis ends when the maximum number of increments for the step is exceeded.

In snap-through studies such as these, the structure can carry increasing load after a complete snap. Therefore, the analysis is terminated conveniently by specifying a maximum load proportionality factor.

For the clamped shallow arch the initial snap occurs at a pressure of about −1000 (force/length^2 units). Thus, −250 (force/length^2 units) seems to be a reasonable estimate for the first increment of load to be applied. Accordingly, an initial time increment of 0.05 is specified for a time period of 1.0 and a pressure load of −5000 (force/length^2 units). The solution will have been sufficiently developed at a pressure of about −2000 (force/length^2 units). Therefore, the analysis is terminated when the load proportionality factor exceeds 0.4.

To illustrate the use of Riks in several steps, a second step is included in which the pressure is taken off the arch so that it will snap back toward its initial configuration. At any point in a Riks analysis, the actual load is given by $P = P_0 + \lambda(P_{ref} - P_0)$, where P_0 is the load at the end of the previous step, P_{ref} is the load magnitude prescribed in the current step, and λ is the load proportionality factor. The arch is unloaded so that in the initial time increment, a pressure of approximately $0.15P_0$ is removed. Using an initial time increment of 0.05 in a time period of 1.0, a load of $P_{ref} = -2P_0$ is prescribed for this restarted step. Furthermore, we want the analysis to end when all the load is removed and the arch has returned to its initial configuration. Therefore, a displacement threshold of 0.0 is set for the center of the arch. The analysis terminates when this limit is crossed. Because Abaqus must pick up the load magnitude at the end of the initial Riks step to start the next step, any step following a Riks step can be done only as a restart job from the previous step.

For the deep clamped-hinged arch, the initial snap occurs at a load of about 900 (force units). The load magnitude specified is 100 (force units), and the maximum load proportionality factor is specified as 9.5.

The shallow arch depressed with a rigid punch is shown in Figure 1.2.1–3. The analysis uses the same model of the arch as the first problem. However, the end is pinned rather than clamped, and load is applied through the displacement of the punch. The pinned boundary condition makes the problem more unstable than the clamped-end case. A preliminary analysis in which the arch is depressed with a prescribed displacement of the midpoint of the arch shows that the force will become negative during snap-through. Thus, if the arch is depressed with a rigid punch, the Riks method will not help convergence because, at the moment of snap-through, the arch separates from the punch, and the movement of the punch no longer controls the displacement of the arch. Therefore, contact damping is introduced to aid in convergence. Viscous damping with surface contact adds a pressure that is proportional to the relative velocity to slow down the separation of the arch from the punch.

The viscous damping clearance is set to 10.0, and the fraction of the clearance interval is set to 0.9; the damping is constant for a clearance of up to 9.0. Since the arch is 4.0 units high, the distance traveled by the top of the arch from the initial position to the final snap-through position is 8.0 units. This distance is clearly larger than the clearance between the middle of the arch and the tip of the punch at

any time during the analysis. Thus, the viscous damping is in effect for the whole period when the arch has separated from the punch.

To choose the viscous damping coefficient, note that it is given as pressure per relative velocity. The relevant pressure is obtained by dividing the approximate peak force (10000.0) by the contact area (1.0). The relevant velocity is obtained by dividing the distance over which the top of the arch travels (8.0 from initial to snapped position, which can be rounded to 10.0) by the time (approximately 1.0, the total time of the step). A small percentage (0.1%) of this value is used for the viscous damping coefficient:

$$\mu = 0.001\frac{Ft}{Al} = 0.001\frac{(10000.0)(1.0)}{(1.0)(10.)} = 1.0.$$

With $\mu = 1.0$, the analysis runs to completion. Another analysis was run with a smaller value of $\mu = 0.1$, but the viscous damping was not sufficient to enable the analysis to pass the point of snap-through. Thus, a damping coefficient of 1.0 was determined to be an appropriate value.

Automatic stabilization based on volume proportional damping is also considered for the shallow arch compressed with a rigid punch, as an alternative to contact damping. Two forms of automatic stabilization are considered: one with a constant damping factor that is chosen by default (see "Automatic stabilization of static problems with a constant damping factor" in "Solving nonlinear problems," Section 7.1.1 of the Abaqus Analysis User's Guide), and one with an adaptive damping factor (see "Adaptive automatic stabilization scheme" in "Solving nonlinear problems," Section 7.1.1 of the Abaqus Analysis User's Guide).

Results and discussion

The results for the clamped shallow arch are shown in Figure 1.2.1–4, where the downward displacement of the top of the arch is plotted as a function of the pressure. The algorithm obtains this solution in 12 increments, with a maximum of three iterations in an increment. At the end of 12 increments the displacement of the top of the arch is about 7.5 length units. This represents a complete snap through, as the original rise of the arch was 4 length units. Figure 1.2.1–5 and Figure 1.2.1–6 show a series of deformed configuration plots for this problem. Several other authors have examined this same case and have obtained essentially the same solution (see Ramm, 1981, and Sharafi and Popov, 1971).

The results for the deep clamped-hinged arch are shown in Figure 1.2.1–7, where the displacement of the top of the arch is plotted as a function of the applied load. Figure 1.2.1–8 shows a series of deformed configuration plots for this problem. The arch collapses unstably at the peak load. Following this, the beam stiffens rapidly as the load increases. The ability of the Riks method to handle unstable response is well-illustrated by this example.

The results of the preliminary analysis of the prescribed displacement of a pinned shallow arch are shown in Figure 1.2.1–9, with the displacement of the top of the arch plotted as a function of the reaction force at that point. This plot shows the negative force that develops during snap-through. A series of deformed configuration plots for the pinned shallow arch depressed with a punch and with viscous damping introduced is shown in Figure 1.2.1–10, with one plot showing the arch separated from the punch. Figure 1.2.1–11 is a plot of the force between the punch and the top of the arch. The force is positive until snap-through, when the arch separates from the punch and a negative viscous force

develops. Once the snap-through is complete, the force drops to zero as the punch continues to move down while separated from the arch. When the punch contacts the arch, a positive force develops again.

Similar results are produced when the contact viscous damping is replaced by volume proportional damping (with either constant or adaptive damping coefficients). A sequence of configurations like Figure 1.2.1–10 is obtained, in which separation of the arch from the punch occurs during snap-through. At the end of the analysis the amount of energy dissipated is similar to the amount dissipated with the viscous damping option.

You can use the **abaqus restartjoin** execution procedure to extract data from the output database created by a restart analysis and append the data to a second output database. For more information, see "Joining output database (`.odb`) files from restarted analyses," Section 3.2.21 of the Abaqus Analysis User's Guide.

Input files

snapbuckling_shallow_step1.inp	Initial analysis step for the shallow arch.
snapbuckling_shallow_unload.inp	Restart run to obtain the unloading response of the shallow arch.
snapbuckling_deep.inp	Deep arch.
snapbuckling_shallow_midpoint.inp	Shallow arch loaded by a fixed displacement of the midpoint.
snapbuckling_shallow_punch.inp	Shallow arch loaded by the displacement of a rigid punch.
snapbuckling_b21h_deep.inp	60 elements of type B21H used for the deep clamped-hinged arch analysis.
snapbuckling_b32h_deep.inp	30 elements of type B32H used for the deep clamped-hinged arch analysis.
snapbuckling_restart1.inp	Restart analysis of snapbuckling_shallow_step1.inp during the RIKS step.
snapbuckling_restart2.inp	Restart analysis of snapbuckling_restart1.inp during the RIKS step. This illustrates restarting an existing RIKS restart analysis.
snapbuckling_shallow_stabilize.inp	Same as snapbuckling_shallow_punch.inp with the surface contact viscous damping replaced by the volume proportional damping of *STATIC, STABILIZE.
snapbuckling_shallow_stabilize_adap.inp	Same as snapbuckling_shallow_punch.inp with the surface contact viscous damping replaced by adaptive stabilization of *STATIC, STABILIZE, ALLSDTOL.

References

- DaDeppo, D. A., and R. Schmidt, "Instability of Clamped-Hinged Circular Arches Subjected to a Point Load," *Transactions of the American Society of Mechanical Engineers*, Journal of Applied Mechanics, pp. 894–896, Dec. 1975.

- Ramm, E., "Strategies for Tracing the Nonlinear Response Near Limit Points," in *Nonlinear Finite Element Analysis in Structural Mechanics*, edited by W. Wunderlich, E. Stein and K. J. Bathe, Springer Verlag, Berlin, 1981.
- Sharifi, P., and E. P. Popov, "Nonlinear Buckling Analysis of Sandwich Arches," Proc. ASCE, Journal of the Engineering Mechanics Division, vol. 97, pp. 1397–1412, 1971.

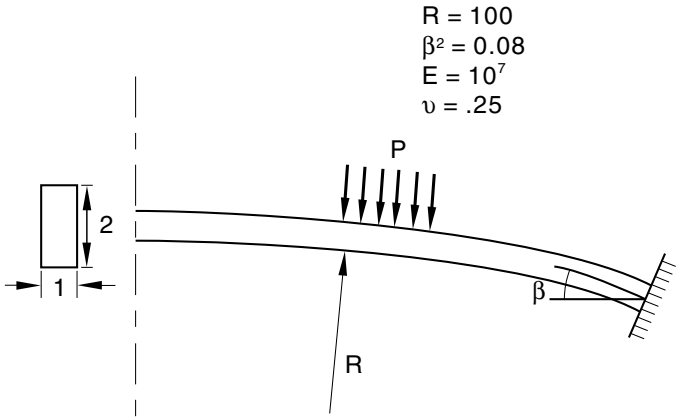


Figure 1.2.1–1 Clamped shallow circular arch.

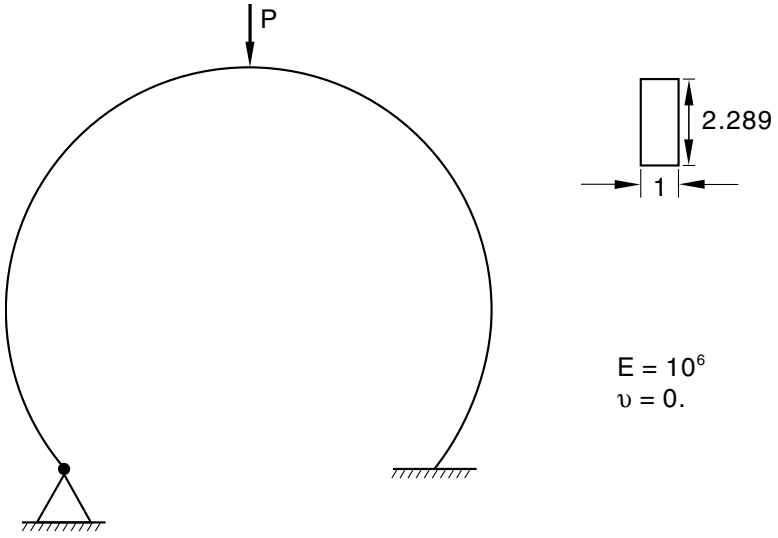


Figure 1.2.1–2 Deep clamped-hinged arch.

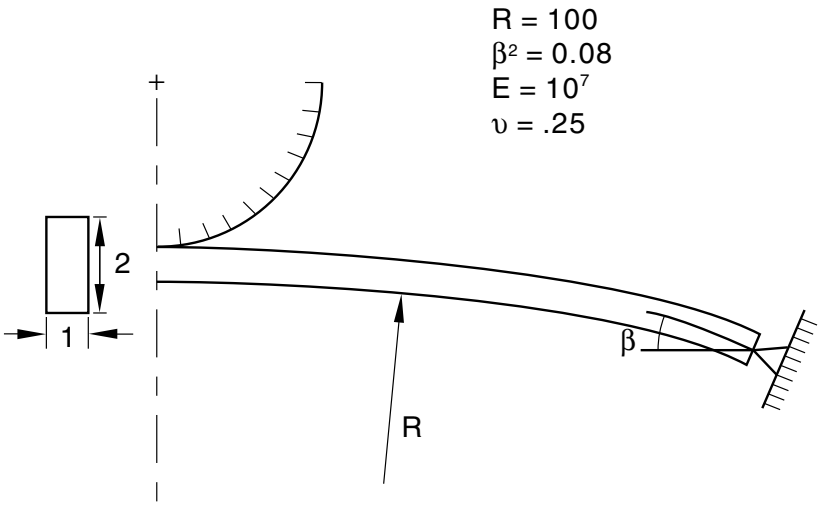


Figure 1.2.1–3 Pinned shallow arch with rigid punch.

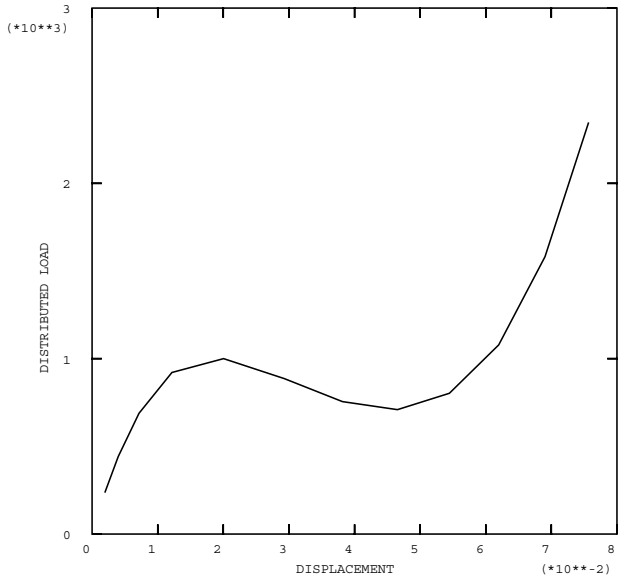


Figure 1.2.1–4 Load versus displacement curve for clamped shallow arch.

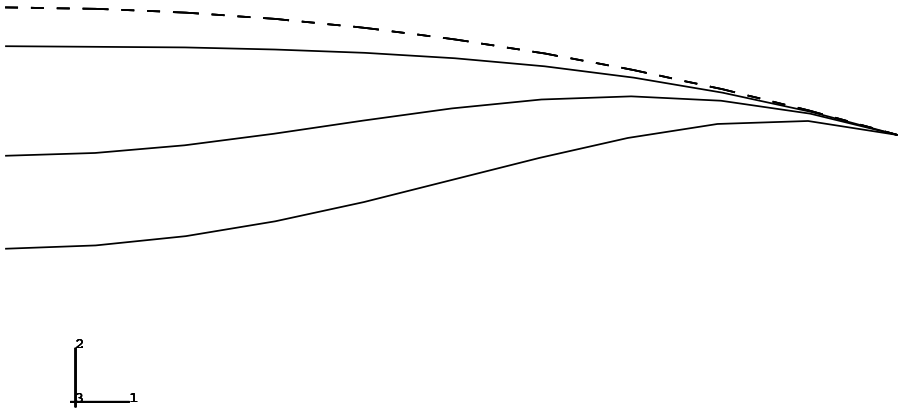


Figure 1.2.1–5 Deformed configuration plots for clamped shallow arch–Step 1.

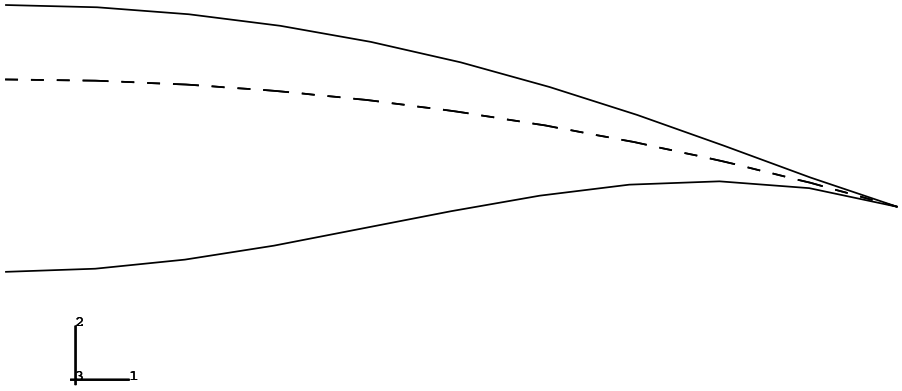


Figure 1.2.1–6 Deformed configuration plots for clamped shallow arch–Step 2.

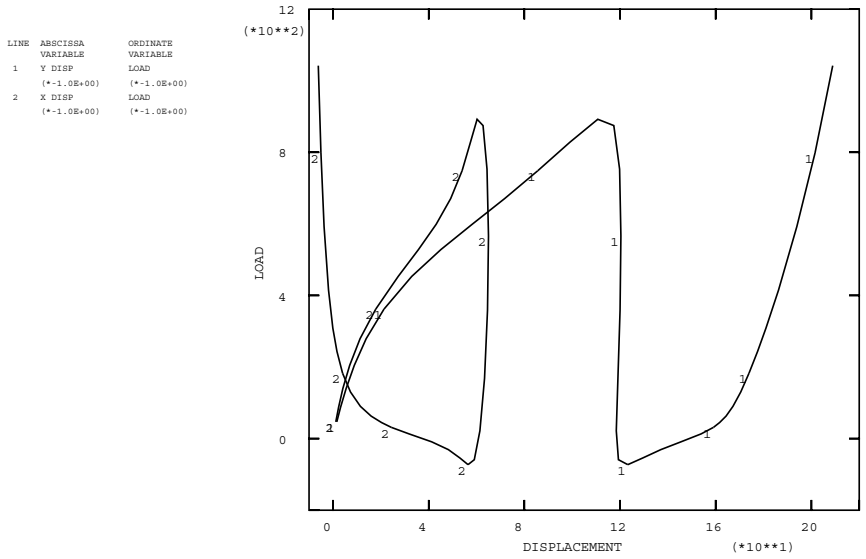


Figure 1.2.1–7 Load versus displacement curves for deep clamped-hinged arch.

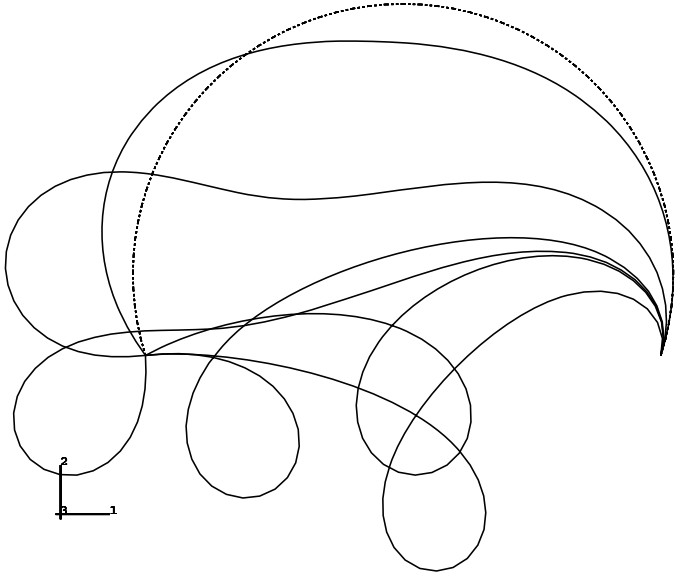


Figure 1.2.1–8 Deformed configuration plots for deep clamped-hinged arch.

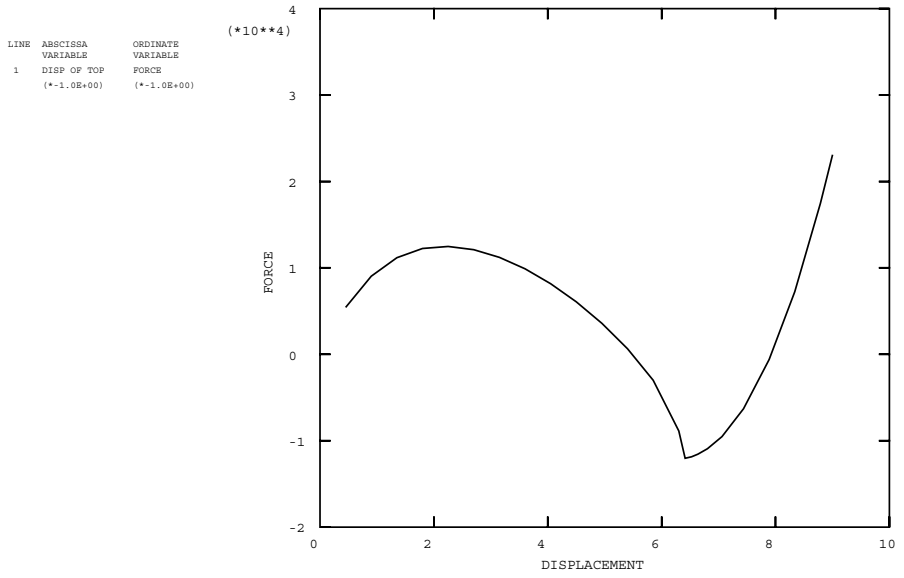


Figure 1.2.1–9 Force versus displacement curve for fixed displacement of pinned shallow arch.

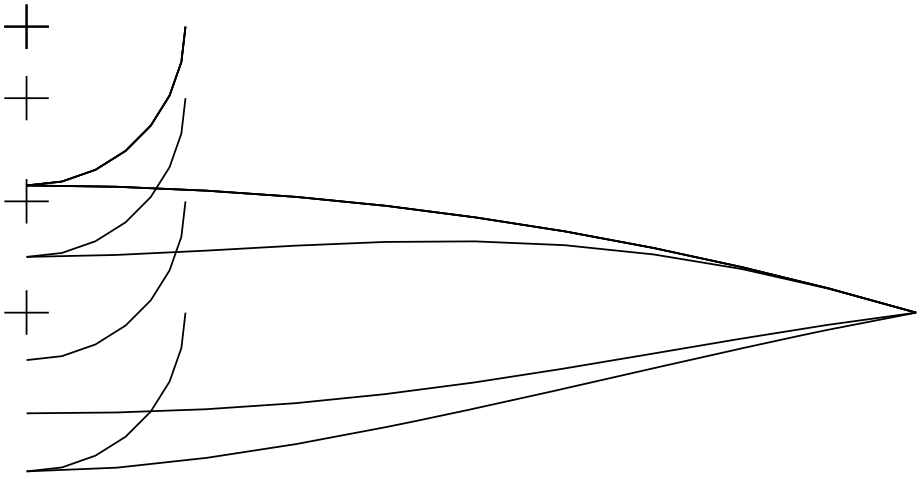


Figure 1.2.1–10 Deformed configuration plots for pinned arch depressed with rigid punch.

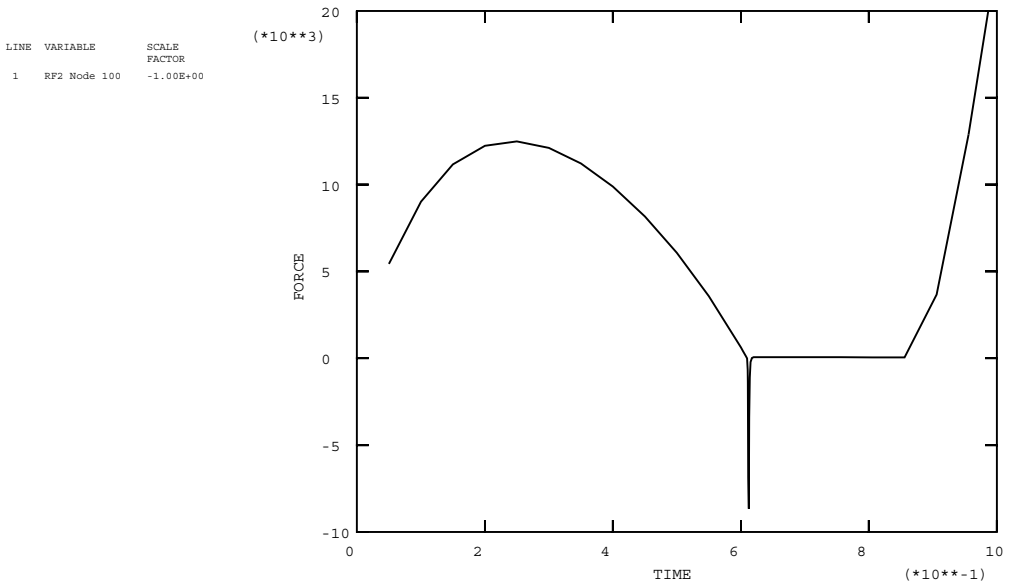


Figure 1.2.1–11 Force between the punch and the top of the pinned arch.

1.2.2 LAMINATED COMPOSITE SHELLS: BUCKLING OF A CYLINDRICAL PANEL WITH A CIRCULAR HOLE

Product: Abaqus/Standard

This example illustrates a type of analysis that is of interest in the aerospace industry. The objective is to determine the strength of a thin, laminated composite shell, typical of shells used to form the outer surfaces of aircraft fuselages and rocket motors. Such analyses are complicated by the fact that these shells typically include local discontinuities—stiffeners and cutouts—which can induce substantial stress concentrations that can delaminate the composite material. In the presence of buckling this delamination can propagate through the structure to cause failure. In this example we study only the geometrically nonlinear behavior of the shell: delamination or other section failures are not considered. Some estimate of the possibility of material failure could presumably be made from the stresses predicted in the analyses reported here, but no such assessment is included in this example.

The example makes extensive use of material orientation in a general shell section to define the multilayered, anisotropic, laminated section. The various orientation options for shells are discussed in "Analysis of an anisotropic layered plate," Section 1.1.2 of the Abaqus Benchmarks Guide.

General shell sections offer two methods of defining laminated sections: defining the thickness, material, and orientation of each layer or defining the equivalent section properties directly. The last method is particularly useful if the laminate properties are obtained directly from experiments or a separate preprocessor. This example uses both methods with a general shell section definition. Alternatively, you could use a shell section to analyze the model; however, because the material behavior is linear, no difference in solution would be obtained and the computational costs would be greater.

Geometry and model

The structure analyzed is shown in Figure 1.2.2–1 and was originally studied experimentally by Knight and Starnes (1984). The test specimen is a cylindrical panel with a 355.6 mm (14 in) square platform and a 381 mm (15 in) radius of curvature, so that the panel covers a 55.6° arc of the cylinder. The panel contains a centrally located hole of 50.8 mm (2 in) diameter. The shell consists of 16 layers of unidirectional graphite fibers in an epoxy resin. Each layer is 0.142 mm (.0056 in) thick. The layers are arranged in the symmetric stacking sequence $\{\pm 45/90/0/0/90/\mp 45\}$ degrees repeated twice. The nominal orthotropic elastic material properties as defined by Stanley (1985) are

$E_{11} = 135$ kN/mm^2 $(19.6 \times 10^6$ lb/in$^2)$,
$E_{22} = 13$ kN/mm^2 $(1.89 \times 10^6$ lb/in$^2)$,
$G_{12} = G_{13} = 6.4$ kN/mm^2 $(.93 \times 10^6$ lb/in$^2)$,
$G_{23} = 4.3$ kN/mm^2 $(0.63 \times 10^6$ lb/in$^2)$,
$\nu_{12} = 0.38$,

where the 1-direction is along the fibers, the 2-direction is transverse to the fibers in the surface of the lamina, and the 3-direction is normal to the lamina.

The panel is fully clamped on the bottom edge, clamped except for axial motion on the top edge and simply supported along its vertical edges. Three analyses are considered. The first is a linear (prebuckling) analysis in which the panel is subjected to a uniform end shortening of 0.8 mm (.0316 in). The total axial force and the distribution of axial force along the midsection are used to compare the results with those obtained by Stanley (1985). The second analysis consists of an eigenvalue extraction of the first five buckling modes. The buckling loads and mode shapes are also compared with those presented by Stanley (1985). Finally, a nonlinear load-deflection analysis is done to predict the postbuckling behavior, using the modified Riks algorithm. For this analysis an initial imperfection is introduced. The imperfection is based on the fourth buckling mode extracted during the second analysis. These results are compared with those of Stanley (1985) and with the experimental measurements of Knight and Starnes (1984).

The mesh used in Abaqus is shown in Figure 1.2.2–2. The anisotropic material behavior precludes any symmetry assumptions, hence the entire panel is modeled. The same mesh is used with the 4-node shell element (type S4R5) and also with the 9-node shell element (type S9R5); the 9-node element mesh, thus, has about four times the number of degrees of freedom as the 4-node element mesh. The 6-node triangular shell element STRI65 is also used; it employs two triangles for each quadrilateral element of the second-order mesh. Mesh generation is facilitated by specifying node fill and node mapping, as shown in the input data. In this model specification of the relative angle of orientation to define the material orientation within each layer, along with orthotropic elasticity in plane stress, makes the definition of the laminae properties straightforward.

The shell elements used in this example use an approximation to thin shell theory, based on a numerical penalty applied to the transverse shear strain along the element edges. These elements are not universally applicable to the analysis of composites since transverse shear effects can be significant in such cases and these elements are not designed to model them accurately. Here, however, the geometry of the panel is that of a thin shell; and the symmetrical lay-up, along with the relatively large number of laminae, tends to diminish the importance of transverse shear deformation on the response.

Relation between stress resultants and generalized strains

The shell section is most easily defined by giving the layer thickness, material, and orientation, in which case Abaqus preintegrates to obtain the section stiffness properties. However, the user can choose to input the section stiffness properties directly instead, as follows.

In Abaqus a lamina is considered as an orthotropic sheet in plane stress. The principal material axes of the lamina (see Figure 1.2.2–3) are longitudinal, denoted by L; transverse to the fiber direction in the surface of the lamina, denoted by T; and normal to the lamina surface, denoted by N. The constitutive relations for a general orthotropic material in the principal directions (L, T, N) are

$$\begin{Bmatrix} \sigma_L \\ \sigma_T \\ \tau_{LT} \\ \sigma_N \\ \tau_{LN} \\ \tau_{TN} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & 0 & C_{14} & 0 & 0 \\ C_{12} & C_{22} & 0 & C_{24} & 0 & 0 \\ 0 & 0 & C_{33} & 0 & 0 & 0 \\ C_{14} & C_{24} & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_L \\ \epsilon_T \\ \gamma_{LT} \\ \epsilon_N \\ \gamma_{LN} \\ \gamma_{TN} \end{Bmatrix}.$$

In terms of the data required to define orthotropic elasticity by specifying terms in the elastic stiffness matrix in Abaqus these are

$$
\begin{aligned}
&C_{11} = D_{1111}, \quad C_{12} = D_{1122}, \quad C_{14} = D_{1133}, \quad C_{22} = D_{2222} \\
&C_{24} = D_{2233}, \quad C_{33} = D_{1212}, \quad C_{44} = D_{3333}, \quad C_{55} = D_{1313} \\
&C_{66} = D_{2323}
\end{aligned}
$$

This matrix is symmetric and has nine independent constants. If we assume a state of plane stress, then σ_N is taken to be zero. This yields

$$
\begin{Bmatrix} \sigma_L \\ \sigma_T \\ \tau_{LT} \\ \tau_{LN} \\ \tau_{TN} \end{Bmatrix} =
\begin{bmatrix} Q_{11} & Q_{12} & 0 & 0 & 0 \\ Q_{12} & Q_{22} & 0 & 0 & 0 \\ 0 & 0 & Q_{33} & 0 & 0 \\ 0 & 0 & 0 & Q_{55} & 0 \\ 0 & 0 & 0 & 0 & Q_{66} \end{bmatrix}
\begin{Bmatrix} \epsilon_L \\ \epsilon_T \\ \gamma_{LT} \\ \gamma_{LN} \\ \gamma_{TN} \end{Bmatrix},
$$

where

$$
\begin{aligned}
Q_{11} &= C_{11} - \frac{C_{14}C_{14}}{C_{44}}, \\
Q_{12} &= C_{12} - \frac{C_{24}C_{14}}{C_{44}}, \\
Q_{22} &= C_{22} - \frac{C_{24}C_{24}}{C_{44}}, \\
Q_{33} &= C_{33}, \\
Q_{55} &= C_{55}, \\
Q_{66} &= C_{66}.
\end{aligned}
$$

The correspondence between these terms and the usual engineering constants that might be given for a simple orthotropic layer in a laminate is

$$
\begin{aligned}
Q_{11} &= \frac{E_1}{1 - \nu_{12}\nu_{21}}, \\
Q_{12} &= \frac{\nu_{12}E_2}{1 - \nu_{12}\nu_{21}} = \frac{\nu_{21}E_1}{1 - \nu_{12}\nu_{21}}, \\
Q_{22} &= \frac{E_2}{1 - \nu_{12}\nu_{21}}, \\
Q_{33} &= G_{12}, \\
Q_{55} &= G_{13}, \\
Q_{66} &= G_{23}.
\end{aligned}
$$

The parameters used on the right-hand side of the above equation are those that must be provided as part of the definition of orthotropic elasticity in plane stress.

If the $(1, 2, N)$ system denotes the standard shell basis directions that Abaqus chooses by default, the local stiffness components must be rotated to this system to construct the lamina's contribution to the general shell section stiffness. Since Q_{ij} represent fourth-order tensors, in the case of a lamina they are oriented at an angle θ to the standard shell basis directions used in Abaqus. Hence, the transformation is

$$
\begin{aligned}
\bar{Q}_{11} &= Q_{11}\cos^4\theta + 2(Q_{12} + 2Q_{33})\sin^2\theta\cos^2\theta + Q_{22}\sin^4\theta,\\
\bar{Q}_{12} &= (Q_{11} + Q_{22} - 4Q_{33})\sin^2\theta\cos^2\theta + Q_{12}(\sin^4\theta + \cos^4\theta),\\
\bar{Q}_{22} &= Q_{11}\sin^4\theta + 2(Q_{12} + 2Q_{33})\sin^2\theta\cos^2\theta + Q_{22}\cos^4\theta,\\
\bar{Q}_{13} &= (Q_{11} - Q_{12} - 2Q_{33})\sin\theta\cos^3\theta + (Q_{12} - Q_{22} + 2Q_{33})\sin^3\theta\cos\theta,\\
\bar{Q}_{23} &= (Q_{11} - Q_{12} - 2Q_{33})\sin^3\theta\cos\theta + (Q_{12} - Q_{22} + 2Q_{33})\sin\theta\cos^3\theta,\\
\bar{Q}_{33} &= (Q_{11} + Q_{22} - 2Q_{12} - 2Q_{33})\sin^2\theta\cos^2\theta + Q_{33}(\sin^4\theta + \cos^4\theta),\\
\bar{Q}_{55} &= Q_{55}\cos^2\theta - Q_{66}\sin^2\theta,\\
\bar{Q}_{56} &= Q_{55}\sin\theta\cos\theta - Q_{66}\sin\theta\cos\theta,\\
\bar{Q}_{66} &= Q_{55}\sin^2\theta - Q_{66}\cos^2\theta,
\end{aligned}
$$

where $\bar{Q}_{ij}$ are the stiffness coefficients in the standard shell basis directions used by Abaqus.

Abaqus assumes that a laminate is a stack of laminae arranged with the principal directions of each layer in different orientations. The various layers are assumed to be rigidly bonded together. The section force and moment resultants per unit length in the normal basis directions in a given layer can be defined on this basis as

$$
(N_1, N_2, N_{12}) = \int_{-h/2}^{h/2} (\sigma_1, \sigma_2, \tau_{12})\, dz,
$$

$$
\begin{aligned}
(M_1, M_2, M_{12}) &= \int_{-h/2}^{h/2} (\sigma_1, \sigma_2, \tau_{12}) z\, dz,\\
(V_1, V_2) &= \int_{-h/2}^{h/2} (\tau_{13}, \tau_{23})\, dz,
\end{aligned}
$$

where $\boldsymbol{h}$ is the thickness of the layer.

This leads to the relations

$$\begin{Bmatrix} N_1 \\ N_2 \\ N_{12} \\ M_1 \\ M_2 \\ M_{12} \\ V_1 \\ V_2 \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & B_{11} & B_{12} & B_{13} & 0 & 0 \\ A_{12} & A_{22} & A_{23} & B_{12} & B_{22} & B_{23} & 0 & 0 \\ A_{13} & A_{23} & A_{33} & B_{13} & B_{23} & B_{33} & 0 & 0 \\ B_{11} & B_{12} & B_{13} & D_{11} & D_{12} & D_{13} & 0 & 0 \\ B_{12} & B_{22} & B_{23} & D_{12} & D_{22} & D_{23} & 0 & 0 \\ B_{13} & B_{23} & B_{33} & D_{13} & D_{23} & D_{33} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & E_{11} & E_{12} \\ 0 & 0 & 0 & 0 & 0 & 0 & E_{12} & E_{22} \end{bmatrix} \begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \gamma_{12} \\ \kappa_1 \\ \kappa_2 \\ \kappa_{12} \\ \gamma_{13} \\ \gamma_{23} \end{Bmatrix},$$

where the components of this section stiffness matrix are given by

$$(A_{ij}, B_{ij}, D_{ij}) = \int_{-h/2}^{h/2} \bar{Q}_{ij}^m (1, z, z^2)\, dz,\ (i, j = 1, 2, 3)$$

$$E_{ij} = \int_{-h/2}^{h/2} \bar{Q}_{\alpha\beta}^m k_i k_j\, dz,\ (i, j = 1, 2, \text{ and } \alpha,\ \beta = i + 4, j + 4).$$

Here m indicates a particular layer. Thus, the $\bar{Q}_{ij}^m$ depend on the material properties and fiber orientation of the mth layer. The $k_i,\ i = 1{,}2$ parameters are the shear correction coefficients as defined by Whitney (1973). If there are n layers in the lay-up, we can rewrite the above equations as a summation of integrals over the n laminae. The material coefficients will then take the form

$$A_{ij} = \sum_{m=1}^{n} \bar{Q}_{ij}^m (h_m - h_{m-1}),$$

$$B_{ij} = \frac{1}{2} \sum_{m=1}^{n} \bar{Q}_{ij}^m (h_m^2 - h_{m-1}^2),$$

$$D_{ij} = \frac{1}{3} \sum_{m=1}^{n} \bar{Q}_{ij}^m (h_m^3 - h_{m-1}^3),$$

$$E_{ij} = \sum_{m=1}^{n} \bar{Q}_{\alpha\beta}^m (h_m - h_{m-1}) k_i k_j,$$

where the h_m and h_{m-1} in these equations indicate that the mth lamina is bounded by surfaces $z = h_m$ and $z = h_{m-1}$. See Figure 1.2.2–4 for the nomenclature.

These equations define the coefficients required for the direct input of the section stiffness matrix method in the general shell section. Only the $[A]$, $[B]$, and $[D]$ submatrices are needed for that option. The three terms in $[E]$, if required, are defined as part of the transverse shear stiffness. The section forces as defined above are in the normal shell basis directions.

Applying these equations to the laminate defined for this example leads to the following overall section stiffness:

$$[A] = \begin{bmatrix} 138.385 & 44.0189 & 0 \\ 44.0189 & 138.385 & 0 \\ 0 & 0 & 47.1831 \end{bmatrix} \text{ kN/mm}; \quad [B] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix};$$

$$[D] = \begin{bmatrix} 55.670 & 21.638 & 2.138 \\ 21.638 & 58.521 & 2.138 \\ 2.138 & 2.138 & 23.004 \end{bmatrix} \text{ kN-mm}; \quad [E] = \begin{bmatrix} 12.2387 & 0 \\ 0 & 12.2387 \end{bmatrix} \text{ kN/mm},$$

or

$$[A] = \begin{bmatrix} 790.239 & 251.367 & 0 \\ 251.367 & 790.239 & 0 \\ 0 & 0 & 269.436 \end{bmatrix} \times 10^3 \text{ lb/in}; \quad [B] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix};$$

$$[D] = \begin{bmatrix} 492.719 & 191.513 & 18.9245 \\ 191.513 & 517.951 & 18.9245 \\ 18.9245 & 18.9245 & 203.602 \end{bmatrix} \text{ lb-in}; \quad [E] = \begin{bmatrix} 49.573 & 0.002 \\ 0.002 & 52.967 \end{bmatrix} \times 10^3 \text{ lb/in}.$$

Results and discussion

The total axial force necessary to compress the panel 0.803 mm (0.0316 in) is 100.2 kN (22529 lb) for the mesh of S9R5 elements, 99.5 kN (22359 lb) for the mesh of S4R5 elements, and 100.3 kN (22547 lb) for the mesh of STRI65 elements. These values match closely with the result of 100 kN (22480 lb) reported by Stanley (1985). Figure 1.2.2–5 shows the displaced configuration and a profile of axial force along the midsection of the panel (at $z = L/2$). It is interesting to note that the axial load is distributed almost evenly across the entire panel, with only a very localized area near the hole subjected to an amplified stress level. This suggests that adequate results for this linear analysis could also be obtained with a coarser mesh that has a bias toward the hole.

The second stage of the analysis is the eigenvalue buckling prediction. To obtain the buckling predictions with Abaqus, an eigenvalue buckling prediction step is run. In this step nominal values of load are applied. The magnitude that is used is not of any significance, since eigenvalue buckling is a linear perturbation procedure: the stiffness matrix and the stress stiffening matrix are evaluated at the beginning of the step without any of this load applied. The eigenvalue buckling prediction step calculates the eigenvalues that, multiplied with the applied load and added to any "base state" loading, are the predicted buckling loads. The eigenvectors associated with the eigenvalues are also obtained. This procedure is described in more detail in "Eigenvalue buckling prediction," Section 6.2.3 of the Abaqus Analysis User's Guide.

The buckling predictions are summarized in Table 1.2.2–1 and Figure 1.2.2–6. The buckling load predictions from Abaqus are higher than those reported by Stanley. The eigenmode predictions given by the mesh using element types S4R5, S9R5, and STRI65 are all the same and agree well with those reported by Stanley. Stanley makes several important observations that remain valid for the Abaqus results: (1) the eigenvalues are closely spaced; (2) nevertheless, the mode shapes vary significantly in

character; (3) the first buckling mode bears the most similarity to the linear prebuckling solution; (4) there is no symmetry available that can be utilized for computational efficiency.

Following the eigenvalue buckling analyses, nonlinear postbuckling analysis is carried out by imposing an imperfection based on the fourth buckling mode. The maximum initial perturbation is 10% of the thickness of the shell. The load versus normalized displacement plots for the S9R5 mesh, the S4R5 mesh, and the STRI65 mesh are compared with the experimental results and those given by Stanley in Figure 1.2.2–7. The overall response prediction is quite similar for the Abaqus elements, although the general behavior predicted by Stanley is somewhat different. The Abaqus results show a peak load slightly above the buckling load predicted by the eigenvalue extraction, while Stanley's results show a significantly lower peak load. In addition, the Abaqus results show rather less loss of strength after the initial peak, followed quite soon by positive stiffness again. Neither the Abaqus results nor Stanley's results agree closely with the experimentally observed dramatic loss of strength after peak load. Stanley ascribes this to material failure (presumably delamination), which is not modeled in his analyses or in these.

Figure 1.2.2–8 shows the deformed configurations for the panel during its postbuckling response. The plots show the results for S4R5, but the pattern is similar for S9R5 and STRI65. The response is quite symmetric initially; but, as the critical load is approached, a nonsymmetric dimple develops and grows, presumably accounting for the panel's loss of strength. Later in the postbuckling response another wrinkle can be seen to be developing.

Input files

laminpanel_s9r5_prebuckle.inp	Prebuckling analysis for the 9-node (element type S9R5) mesh.
laminpanel_s9r5_buckle.inp	Eigenvalue buckling prediction using element type S9R5.
laminpanel_s9r5_postbuckle.inp	Nonlinear postbuckling analysis using element type S9R5.
laminpanel_s4r5_prebuckle.inp	Prebuckling analysis using element type S4R5.
laminpanel_s4r5_buckle.inp	Eigenvalue buckling prediction using element type S4R5.
laminpanel_s4r5_postbuckle.inp	Nonlinear postbuckling analysis using element type S4R5.
laminpanel_s4r5_node.inp	Nodal coordinate data for the imperfection imposed for the postbuckling analysis using element type S4R5.
laminpanel_s9r5_stri65_node.inp	Nodal coordinate data for the imperfection imposed for the postbuckling analysis using element types S9R5 and STRI65.
laminpanel_stri65_prebuckle.inp	Prebuckling analysis using element type STRI65.
laminpanel_stri65_buckle.inp	Eigenvalue buckling prediction using element type STRI65.
laminpanel_stri65_postbuckle.inp	Nonlinear postbuckling analysis using element type STRI65.
laminpanel_s4_prebuckle.inp	Prebuckling analysis using element type S4.
laminpanel_s4_buckle.inp	Eigenvalue buckling prediction using element type S4.

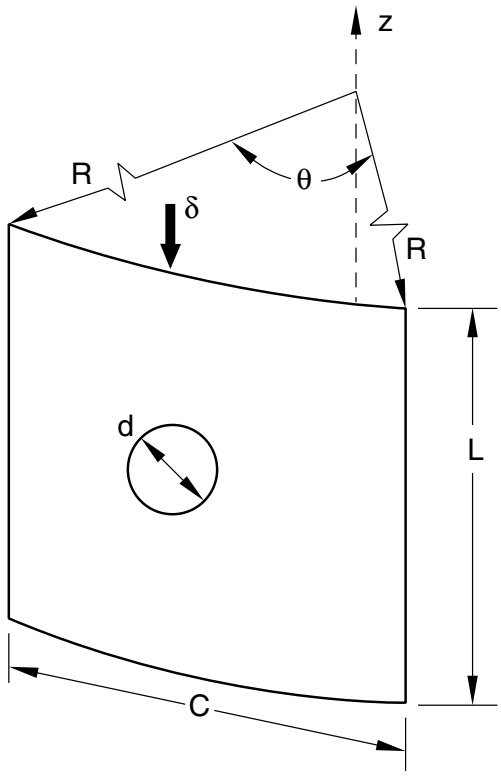
laminpanel_s4_postbuckle.inp	Nonlinear postbuckling analysis using element type S4.

References

- Knight, N. F., and J. H. Starnes, Jr., "Postbuckling Behavior of Axially Compressed Graphite-Epoxy Cylindrical Panels with Circular Holes," presented at the 1984 ASME Joint Pressure Vessels and Piping/Applied Mechanics Conference, San Antonio, Texas, 1984.
- Stanley, G. M., *Continuum-Based Shell Elements,* Ph.D. Dissertation, Department of Mechanical Engineering, Stanford University, 1985.
- Whitney, J. M., "Shear Correction Factors for Orthotropic Laminates Under Static Loads," *Journal of Applied Mechanics*, Transactions of the ASME, vol. 40, pp. 302–304, 1973.

Table 1.2.2–1 Summary of buckling load predictions.

Mode 1	Stanley	107.0 kN (24054 lb)
	S9R5	113.4 kN (25501 lb)
	S4R5	115.5 kN (25964 lb)
	S4	114.3 kN (25696 lb)
	STRI65	113.8 kN (25579 lb)
Mode 2	Stanley	109.6 kN (24638 lb)
	S9R5	117.6 kN (26429 lb)
	S4R5	121.2 kN (27244 lb)
	S4	116.5 kN (26196 lb)
	STRI65	117.8 kN (26492 lb)
Mode 3	Stanley	116.2 kN (26122 lb)
	S9R5	120.3 kN (27049 lb)
	S4R5	124.7 kN (28042 lb)
	S4	124.1 kN (27889 lb)
	STRI65	121.1 kN (27217 lb)
Mode 4	Stanley	140.1 kN (31494 lb)
	S9R5	147.5 kN (33161 lb)
	S4R5	156.1 kN (35092 lb)
	S4	152.3 kN (34247 lb)
	STRI65	146.9 kN (33015 lb)
Mode 5	Stanley	151.3 kN (34012 lb)
	S9R5	171.3 kN (38512 lb)
	S4R5	181.5 kN (40800 lb)
	S4	184.2 kN (41413 lb)
	STRI65	172.8 kN (38843 lb)



GEOMETRIC PROPERTIES:

L = 355.6 mm (14 in)
C = 355.6 mm (14 in)
R = 381.0 mm (15 in)
d = 50.80 mm (2 in)
$\theta = 55.6°$
h_{layer} = 0.142 mm (.0056 in)

LOADING:

Uniform axial compression δ = 0.803 mm (.0316 in)

Figure 1.2.2–1 Geometry for cylindrical panel with hole.

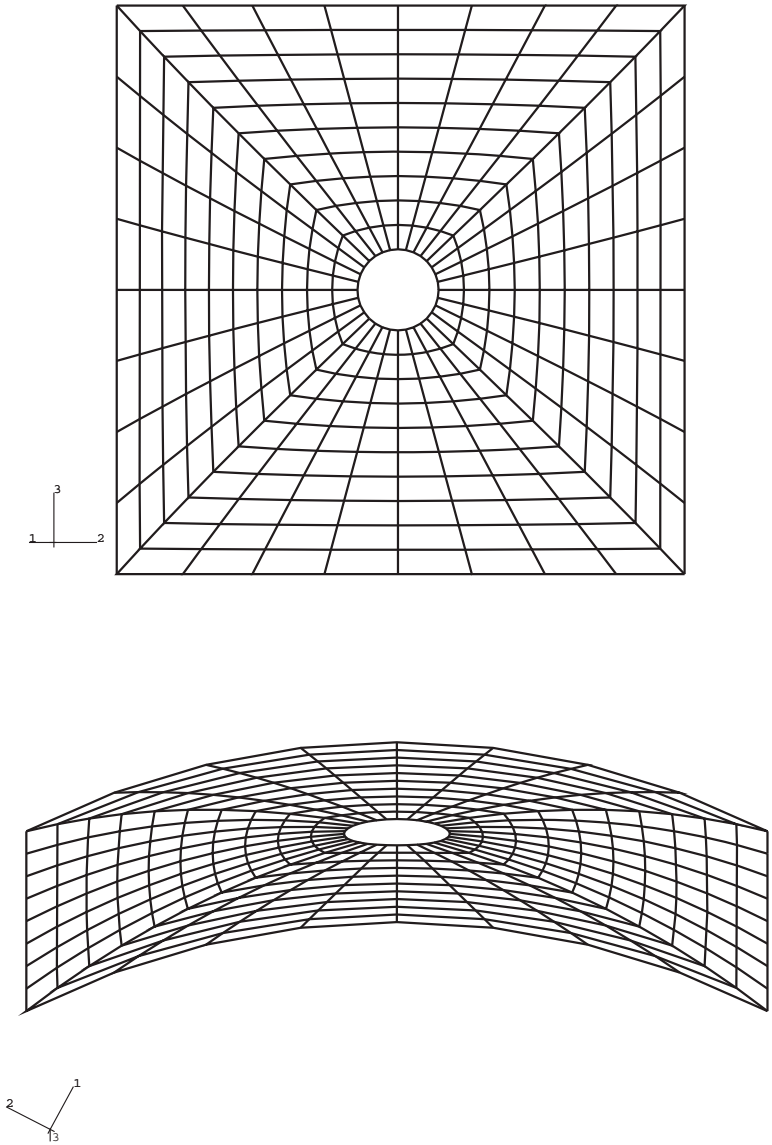


Figure 1.2.2–2 Mesh for cylindrical panel with hole.

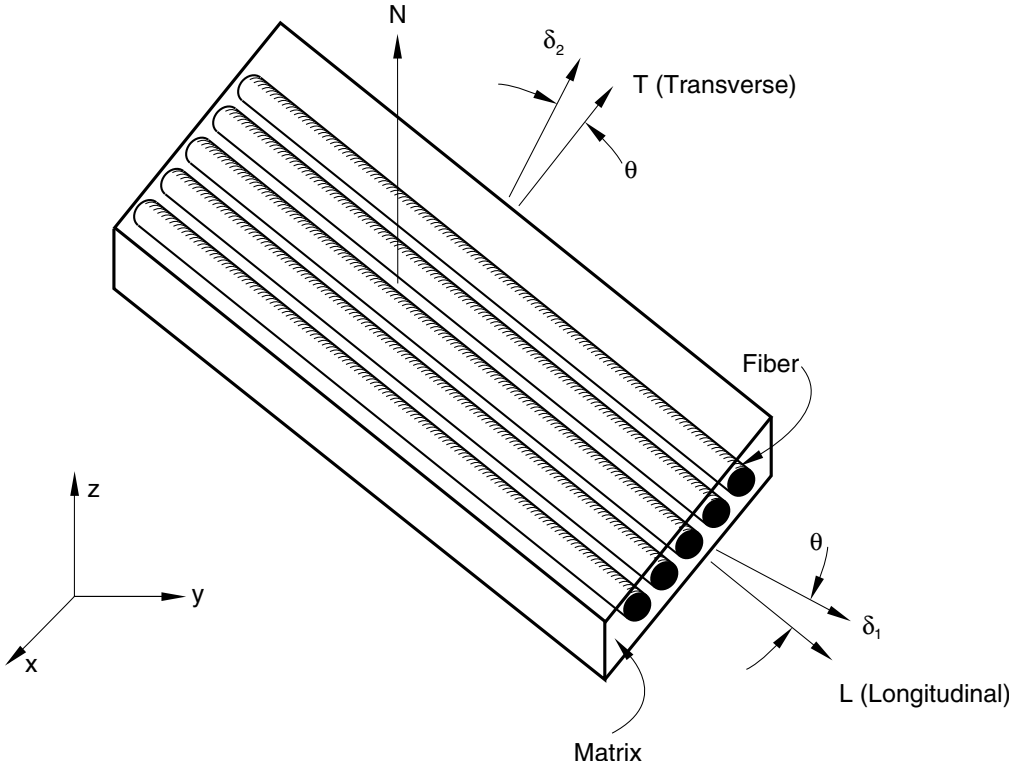


Figure 1.2.2–3 Typical lamina.

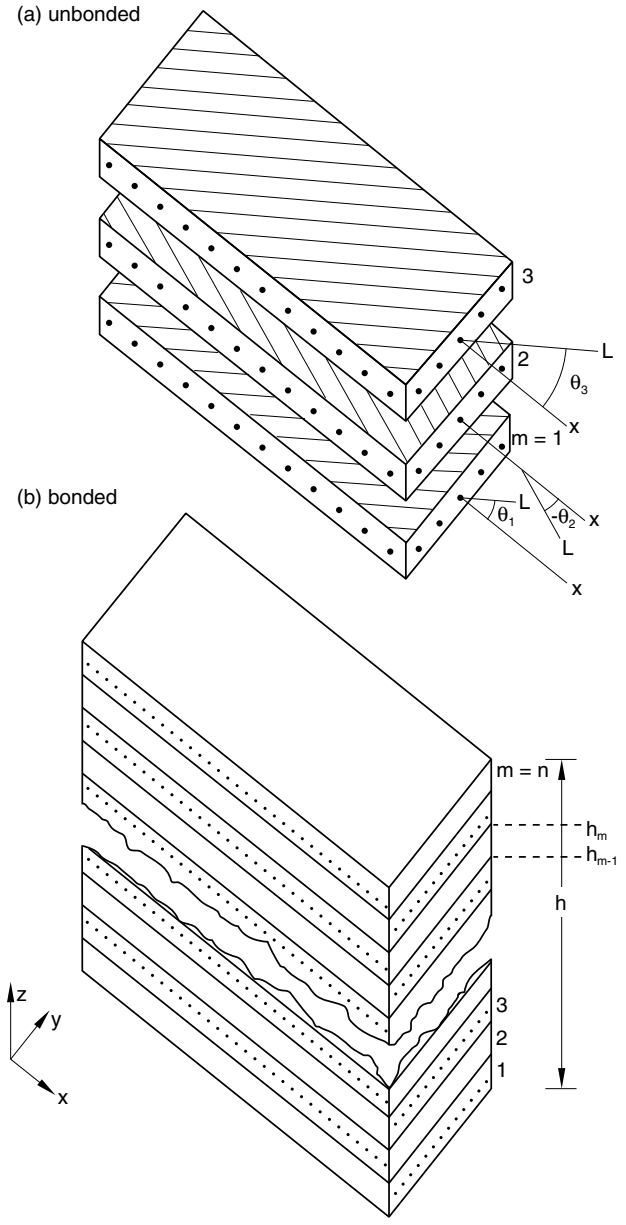


Figure 1.2.2–4 Typical laminate.

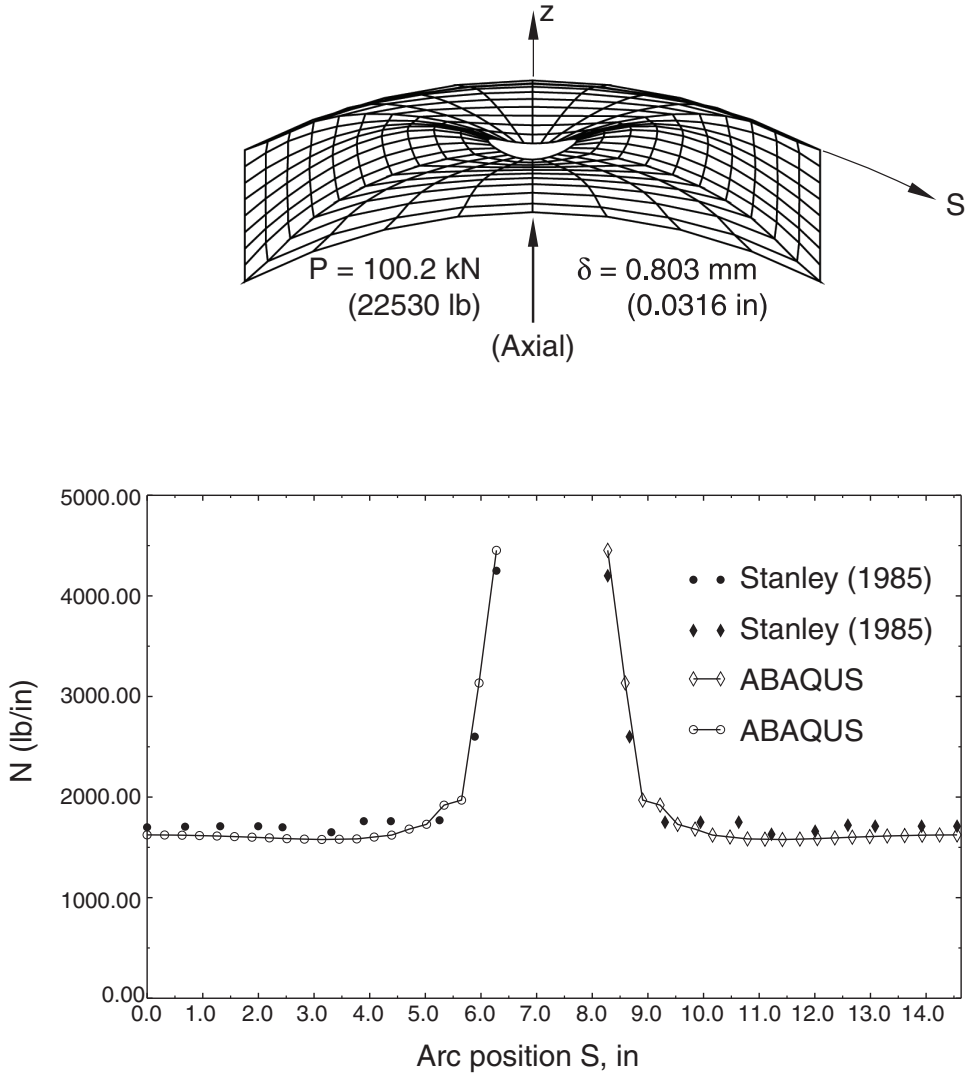


Figure 1.2.2–5 Displaced shape and axial force distribution.

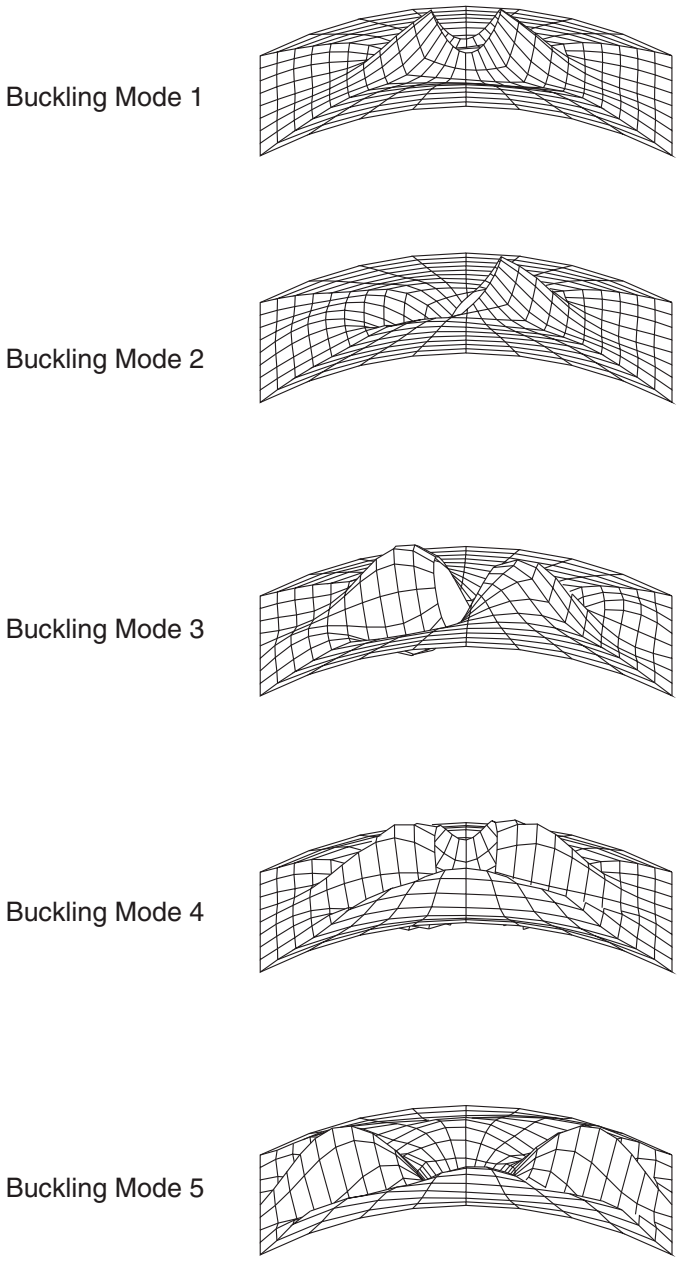


Figure 1.2.2–6 Buckling modes, element types S4R5, S9R5, and STRI65.

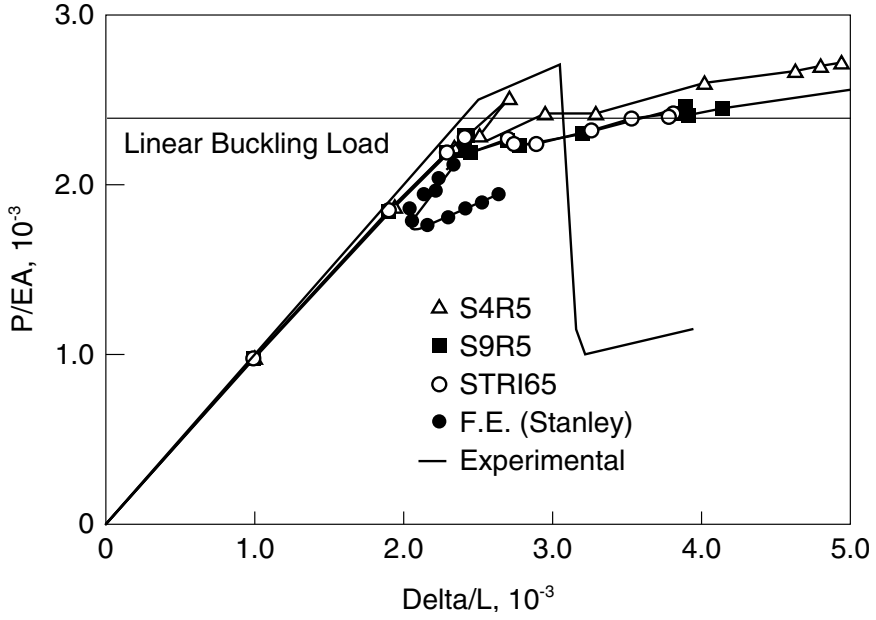


Figure 1.2.2–7 Load-displacement response.

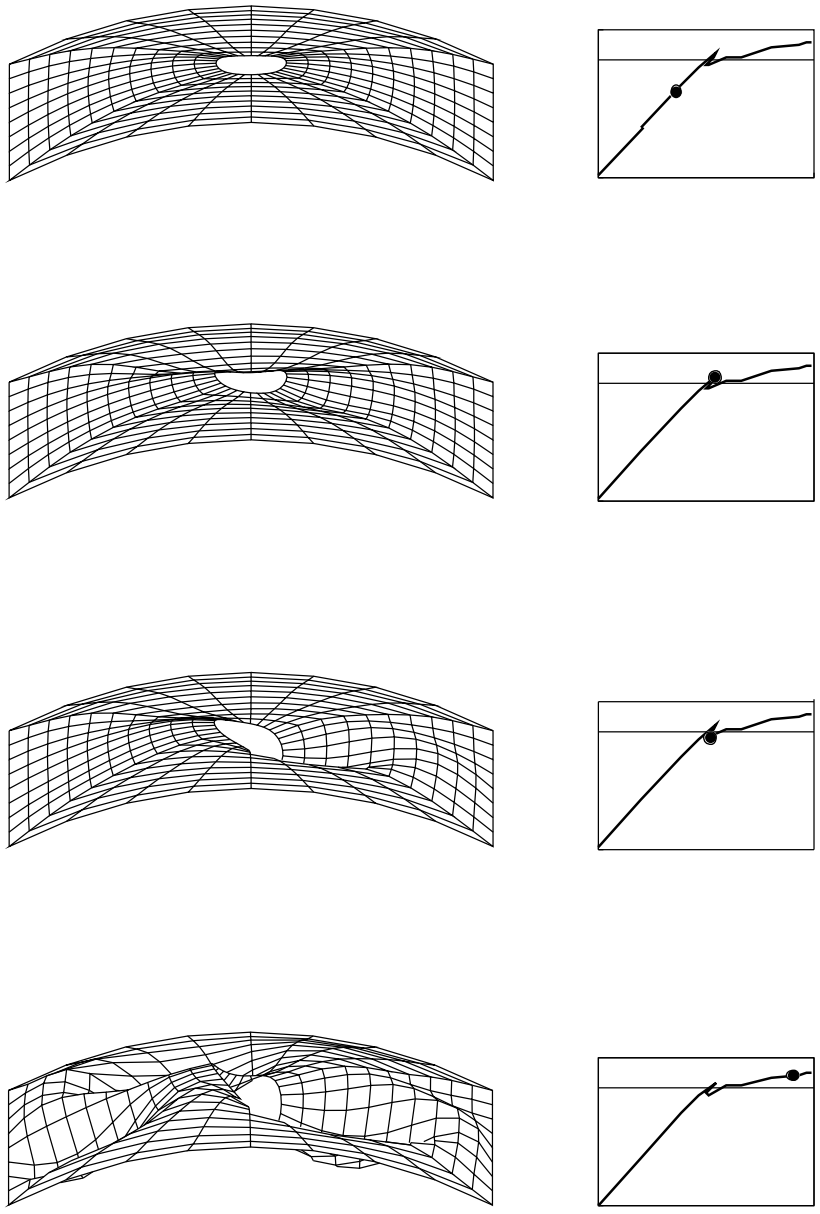


Figure 1.2.2–8 Postbuckling deformations: 10% h imperfection with S4R5.

1.2.3 BUCKLING OF A COLUMN WITH SPOT WELDS

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

This example illustrates both a static and dynamic collapse of a steel column constructed by spot welding two channel sections. It is intended to illustrate the modeling of spot welds. "Mesh-independent fasteners," Section 35.3.4 of the Abaqus Analysis User's Guide, discusses the mesh-independent spot weld modeling capabilities provided in Abaqus; while "Breakable bonds," Section 37.1.9 of the Abaqus Analysis User's Guide, discusses the use of bonds and bonding properties to model breakable spot welds in Abaqus/Explicit.

Problem description

The pillar is composed of two columns of different cross-sections, one box-shaped and the other W-shaped, welded together with spot welds (Figure 1.2.3–1). The top end of the pillar is connected to a rigid body, which makes the deformation of the pillar easy to control by manipulating the rigid body reference node. The box-shaped column is welded to the W-shaped column with five spot welds on either side of the box-shaped column.

The columns are both composed of aluminum-killed steel, which is assumed to satisfy the Ramberg-Osgood relation between true stress and logarithmic strain,

$$\varepsilon = \sigma/E + (\sigma/K)^n,$$

where Young's modulus (E) is 206.8 GPa, the reference stress value (K) is 0.510 GPa, and the work-hardening exponent (n) is 4.76. In the present Abaqus analyses the Ramberg-Osgood relation is approximated using elastic and plastic material properties. The material is assumed to be linear elastic up to a yield stress of 170.0 MPa, and the stress-strain curve beyond the yield stress is defined in piecewise linear segments using plastic material properties. Poisson's ratio is 0.3.

The spot welds are modeled in both Abaqus/Standard and Abaqus/Explicit using the mesh-independent fastener capability. Connector elements with CARTESIAN and CARDAN sections are used to define deformable fasteners. Alternatively, a BUSHING connection type could have been used. The element set containing the connector elements is referenced in the mesh-independent fastener. The spot welds at nodes 5203, 15203, 25203, 35203, and 45203 are all located on the positive z-side of the box-shaped column, with node 5203 at the bottom end of the column and node 45203 at the top end of the column (see Figure 1.2.3–2). Spot welds at nodes 5211, 15211, 25211, 35211, and 45211 are all located on the negative z-side of the box-shaped column, with node 5211 at the bottom end of the column and node 45211 at the top end of the column. The surfaces of the box-shaped column and the W-shaped column are specified in the mesh-independent fastener. The spot welds are defined with a diameter of .002 m. The deformable behavior in the fastener is modeled using connector elasticity, with an elastic spring stiffness of 2×10^{11} N/m in translational as well as rotational components. For the Abaqus/Explicit analysis spot weld damage and failure are modeled using connector damage behavior. A force-based coupled damage initiation criterion that uses a connector potential with both connector force and connector moment ingredients is used. (For further description of the connector

potential used, see the spot weld example in "Connector functions for coupled behavior," Section 31.2.4 of the Abaqus Analysis User's Guide.) Damage initiates when the value of the potential exceeds 2×10^5 N. A post-damage-initiation equivalent displacement of 1×10^{-7} m is allowed. Once the post-damage-initiation equivalent displacement in a spot weld reaches this value, the spot weld ceases to carry any load. Both the continuum and structural coupling capabilities are used to define the fasteners.

To study spot weld failure and the post-yield behavior of the spot welds in detail, the problem is also solved using the bond properties available in Abaqus/Explicit. The column with the box-shaped cross-section is defined to be the slave surface in contact with the column with the W-shaped cross-section. The spot welds on the two sides of the box-shaped column are modeled with different yield forces and post-yield behavior to illustrate the two failure models. For the spot-welded nodes 5203, 15203, 25203, 35203, and 45203, the force to cause failure for the spot welds is 3000 N in pure tension and 1800 N in pure shear. Once the spot welds start to fail, the maximum force that they can bear is assumed to decay linearly with time over the course of 2.0 msec, which illustrates the modeling of complete loss of strength over a given time period. For the spot-welded nodes 5211, 15211, 25211, 35211, and 45211, the force to cause failure for these spot welds is 4000 N in pure tension and 2300 N in pure shear. These spot welds fail according to the damaged failure model, which assumes that the maximum forces that the spot welds can carry decay linearly with relative displacement between the welded node and the master surface. The welds are defined to fail completely once their total relative displacement reaches 0.3mm, which illustrates the modeling of loss of strength in the spot welds based on energy absorption.

A Python script is included that reproduces the model using the Scripting Interface in Abaqus/CAE. The script creates and assembles Abaqus/CAE parts and uses discrete fasteners to model the spot welds. The script creates an Abaqus/Standard model that is ready to be submitted for analysis from the Job module. The discrete fasteners created by the script result in the following differences compared with the mesh-independent, or point-based, fasteners used by the example input files:

- When you submit the Abaqus/CAE job for analysis, the discrete fasteners created by the Python script generate coupling constraints and distributing coupling constraints in the input file, together with connector elements. The example input files use mesh-independent fasteners to model point-based fasteners using connector elements.
- You must define the radius of influence when you create a discrete fastener using Abaqus/CAE. In contrast, the example input files allow Abaqus to compute a default value of the radius of influence based on the geometric properties of the fastener, the characteristic length of connected facets, and the type of weighting function selected.
- The input files share nodes between the pillar and the rigid body. To achieve similar behavior, the Python script creates tie constraints between the pillar and the rigid body.

For a description of the differences between discrete fasteners and point-based fasteners in Abaqus/CAE, see "About fasteners," Section 29.1 of the Abaqus/CAE User's Guide.

Loading

The bottom of the pillar is fully fixed. In the Abaqus/Standard analysis the reference node for the rigid body at the top of the pillar moves 0.25 m in the *y*-direction, thus loading it in compression, together with

a displacement of .02 m in the z-direction that shears it slightly. At the same time the end of the pillar is rotated about the negative z-axis by 0.785 rad and rotated about the negative x-axis by 0.07 rad.

In the Abaqus/Explicit analyses the reference node for the rigid body at the top of the pillar moves at a constant velocity of 25 m/sec in the y-direction, thus loading it in compression, together with a velocity of 2 m/sec in the z-direction that shears it slightly. At the same time the end of the pillar is rotated about the negative z-axis at 78.5 rad/sec and rotated about the negative x-axis at 7 rad/sec. This loading is applied by prescribing the velocities of the rigid body reference node that is attached to the top end of the compound pillar.

The analysis is carried out over 10 milliseconds.

Results and discussion

The mesh-independent spot weld capability and the contact-based spot weld capability predict very similar deformation patterns and deformed shapes for the pillar. Figure 1.2.3–3 shows the deformed shape of the pillar after 5.0 msec in the Abaqus/Explicit analysis. Figure 1.2.3–4 shows the deformed shape of the pillar after 10.0 msec. The spot welds in the mesh-independent Abaqus/Explicit analysis undergo damage and fail. For the current choice of parameters for the connector damage model, it is found that damage initiates in the spot welds at nodes 15203 through 45203 on the positive side of the box-shaped column and at nodes 15211 through 45211 on the negative side of the box-shaped column. However, the post-damage-initiation displacement is sufficient to cause ultimate failure of the spot welds at nodes 15203, 25203, 15211, and 25211 only. Figure 1.2.3–9 illustrates the undamaged connector force CTF3 in the spot welds associated with reference nodes 25203 and 25211 as computed in the Abaqus/Standard analyses. Figure 1.2.3–10 illustrates the damaged connector force CTF3 in the spot welds associated with reference nodes 25203 and 25211 as computed in the Abaqus/Explicit analyses. Forces in both spot welds drop to zero when ultimate failure occurs in the Abaqus/Explicit analyses.

The failure and post-yield behaviors of the pillar are also studied using the contact-based spot weld capability. Figure 1.2.3–5 and Figure 1.2.3–6 show the status of the spot welds on the positive z-side of the column and the negative z-side of the column, respectively. In these figures a status of 1.0 means that the weld is fully intact, and a status of 0.0 means that the weld has failed completely. Figure 1.2.3–7 shows the load on spot weld node 25203 relative to the failure load. This relative value is called the bond load and is defined to be 1.0 when the spot weld starts to fail and 0.0 when the spot weld is broken. Figures showing the bond status and bond load may not match the analysis results on a particular platform. This is due to the fact that contact forces in this analysis show significant noise, which can vary across platforms. When the time-to-failure model is used, spot weld behavior is very sensitive to any spike in the bond force that reaches the bond strength. Spot weld behavior is less sensitive to individual spikes in the bond force when the damaged failure model is used. Figure 1.2.3–8 shows the time history of the total kinetic energy, the total work done on the model, the total energy dissipated by friction, the total internal energy, and the total energy balance.

Input files

pillar_fastener_xpl.inp	Input data for the Abaqus/Explicit mesh-independent spot weld analysis.

pillar_fastener_structcoup_xpl.inp	Input data for the Abaqus/Explicit mesh-independent spot weld analysis using structural coupling in the fastener definitions.
pillar_fastener_std.inp	Input data for the Abaqus/Standard mesh-independent spot weld analysis.
pillar_fastener_structcoup_std.inp	Input data for the Abaqus/Standard mesh-independent spot weld analysis using structural coupling in the fastener definitions.
pillar_fastener_smslide_std.inp	Input data for the Abaqus/Standard mesh-independent spot weld analysis using small-sliding contact with shell thickness taken into account.
pillar.inp	Input data for the contact-pair-based spot weld analysis.
pillar_gcont.inp	Input data for the general-contact-based spot weld analysis.
pillar_rest.inp	Input data used to test the restart capability with spot welds.
pillar_ds.inp	Analysis using the double-sided surface capability.

Python script

pillar_fastener_std.py	Script that creates a model with discrete fasteners using Abaqus/CAE.

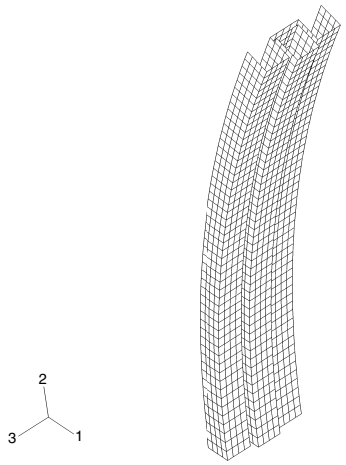


Figure 1.2.3–1 Initial configuration of the compound pillar.

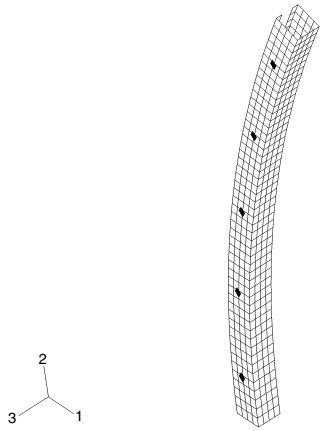


Figure 1.2.3–2 Initial configuration of the box-shaped column showing spot welds.

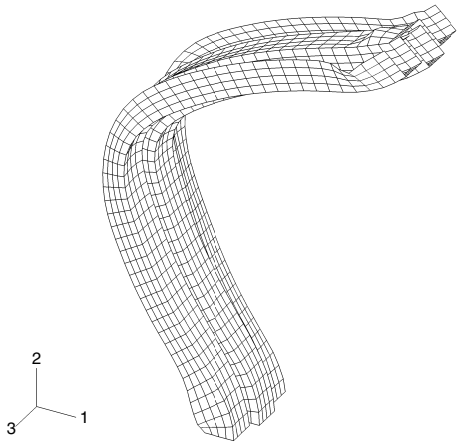


Figure 1.2.3–3 Deformed shape at 5.0 msec.

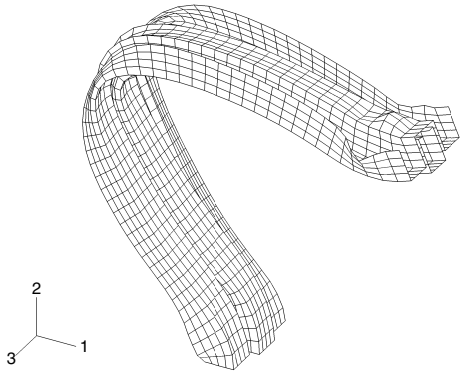


Figure 1.2.3–4 Deformed shape at 10.0 msec.

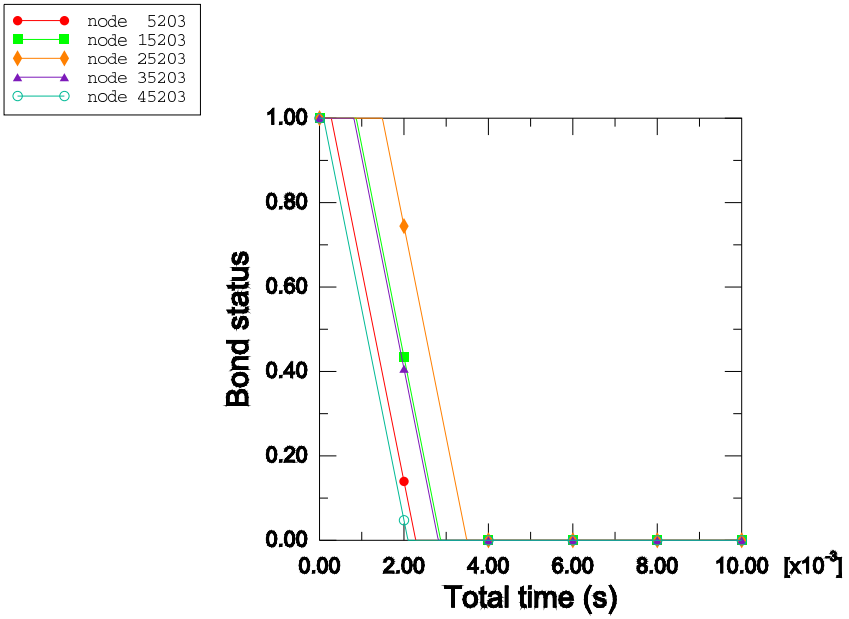


Figure 1.2.3–5 Time histories of the status of all spot welds on positive *z*-side of column.

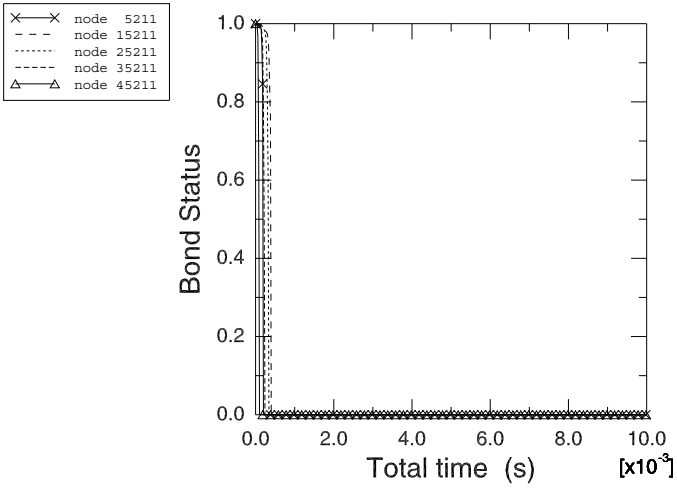


Figure 1.2.3–6 Time histories of the status of all spot welds on negative *z*-side of column.

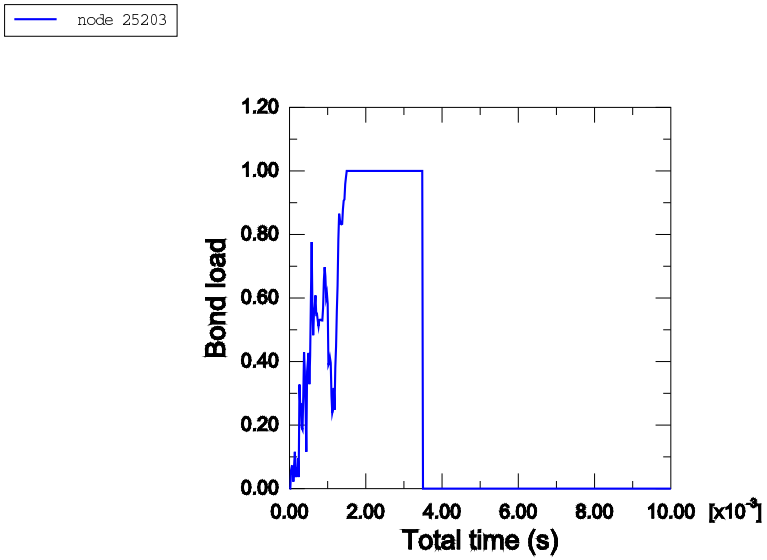


Figure 1.2.3–7 Time histories of the load on spot weld node 25203 relative to the failure load.

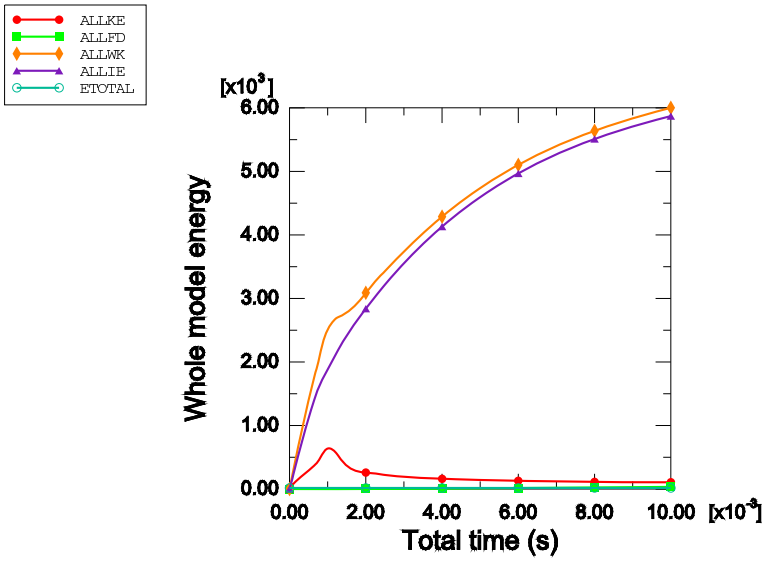


Figure 1.2.3–8 Time histories of the total kinetic energy, energy dissipated by friction, work done on the model, internal energy, and total energy.

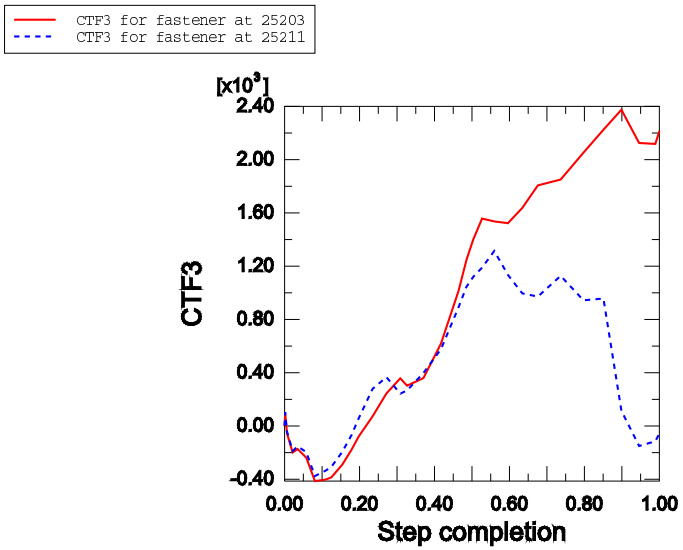


Figure 1.2.3–9 Connector force CTF3 in spot welds at reference nodes 25203 and 25211 for Abaqus/Standard.

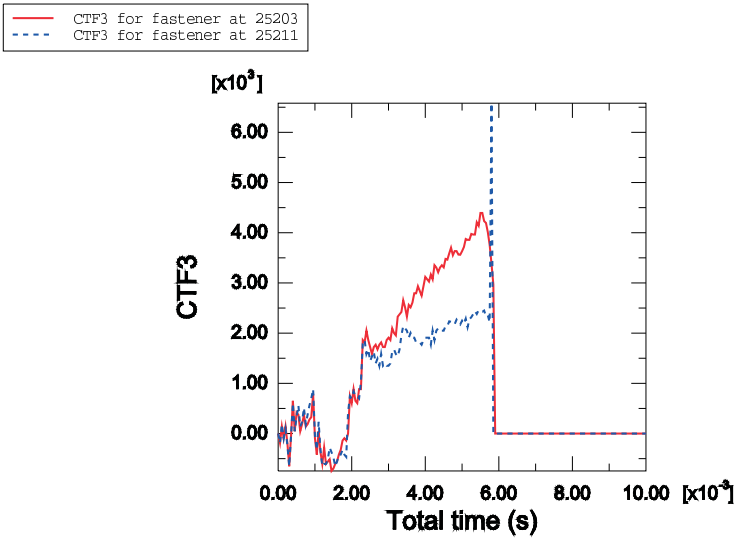


Figure 1.2.3–10 Connector force CTF3 in spot welds at reference nodes 25203 and 25211 for Abaqus/Explicit.

1.2.4 ELASTIC-PLASTIC K-FRAME STRUCTURE

Product: Abaqus/Standard

This example illustrates the use of the frame element FRAME2D. Frame elements ("Frame elements," Section 29.4.1 of the Abaqus Analysis User's Guide) can be used to model elastic, elastic-plastic, and buckling strut responses of individual members of frame-like structures. The elastic response is defined by Euler-Bernoulli beam theory. The elastic-plastic response is modeled with nonlinear kinematic hardening plasticity concentrated at the element's ends, simulating the development of plastic hinges. The buckling strut response is a simplified, phenomenological representation of the highly nonlinear cross-section collapse and material yielding that takes place when slender members are loaded in compression .Therefore, frame elements can be elastic, elastic-plastic, behave as struts (with or without buckling), or switch during the analysis to strut behavior followed by postbuckling behavior. Both the elastic-plastic and buckling strut responses are simplifications of highly nonlinear responses. They are designed to approximate these complex responses with a single finite element representing a structural member between connections. For parts of the model where higher solution resolution is required, such as stress prediction, the model should be refined with beam elements.

The geometry in this example is a typical K-frame construction used in applications such as offshore structures (see Figure 1.2.4–1). A push-over analysis is performed to determine the maximum horizontal load that the structure can support before collapse results from the development of plastic hinges or buckling failure. During a push-over test, many structural members are loaded in compression. Slender members loaded in compression often fail due to geometric buckling, cross-section collapse, and/or material yielding. The buckling strut response, which models such compressive behavior, is added in separate simulations to investigate the effect of the compressive failure of critical members in the structure. A dead load is applied to the top of the structure representing the weight supported by the K-frame. Push-over analyses are either load or displacement control tests.

Geometry and model

The structure consists of 19 members between structural connections. Each finite element models a member of the frame. Hence, 19 frame elements are used: 17 elements with PIPE cross-sections of varying properties and 2 elements (the top platform) with I cross-sections. The plastic response of the elements is calculated from the yield stress of the material, using the plastic default values provided by Abaqus. (The default values for the plastic response are based on experiments with slender steel members. For details on the default values, see "Frame section behavior," Section 29.4.2 of the Abaqus Analysis User's Guide.) The default plastic response includes mild hardening for axial forces and strong hardening for bending moments. The default hardening responses for a typical element in the model are shown in Figure 1.2.4–2 and Figure 1.2.4–3.

A dead load of 444.8 kN (100,000 lb) is applied to the top of the K-frame, representing the part of the structure above the K-frame. Subsequently, the top platform is loaded or displaced horizontally. The load level or applied displacement is chosen to be large enough so that the entire structure fails by the formation of plastic hinges and, consequently, loses load carrying capacity.

Three different models are investigated. A limit load is expected, since the goal of the analysis is to determine when the structure loses overall stiffness. Large- and small-displacement analyses are performed for all three models for comparison. Large-displacement analyses using frame elements are valid for large overall rotations but small strains, since frame elements assume that the strains are small. In the first model all elements use elastic-plastic material response. In the second model buckling is checked for all elements with PIPE cross-sections. The ISO equation is used as a criteria for buckling, and the default Marshall strut envelope is followed for the postbuckling behavior. The buckling strut envelope is calculated from the yield stress of the material and the default Marshall Strut theory. (For details on the default buckling strut envelope, see "Frame section behavior," Section 29.4.2 of the Abaqus Analysis User's Guide.) All frame members that use the buckling strut response check the ISO criteria for the switching-to-strut algorithm. In the third model the member that switches to strut behavior in the second model (element 7) is replaced by a frame element with buckling strut response from the beginning of the analysis. To proceed beyond the unstable phase of the response, the Riks static solution procedure is used in the elastic-plastic problems. To decrease the number of solution iterations, the solution controls are used in the elastic-plastic problem with large displacement, with the value of the ratio of the largest solution correction to the largest incremental solution set to 1.0, since displacement increments are very small after plasticity occurs.

Results and discussion

The structure is loaded or displaced to the point at which all load carrying capacity is lost. In the first model with elastic-plastic frame elements, the results for the linear and nonlinear geometries compare as expected. That is, the limit load for the large-displacement analysis is reached at a load of 1141 kN (256,000 lb) as compared to a higher load of 1290 kN (291,000 lb) in the small-displacement analysis. The plastic hinge pattern is the same in both cases.

The second model uses the switching algorithm. It shows that element 7 first violates the ISO equation (buckles) at a prescribed displacement equal to 1.32 cm (0.52 in), before any elements form plastic hinges. The critical compressive force in this element is –318 kN (–71,400 lb). Next, plasticity develops at several elements, and the structure reaches its limit capacity. The frame elements with the switching algorithm predict the structural behavior in the most accurate way, since the possibility of buckling is checked for all elements in the model, and highly compressed members switch automatically to postbuckling behavior (see the plastic and buckled frame elements in Figure 1.2.4–4). When the structure can no longer support horizontal loading, the patterns of plastic hinges for linear and nonlinear geometry are the same. The results differ more for loads close to the limit load.

To investigate the effect of buckling, the first and the third (element 7 defined with buckling strut response from the beginning) models are compared (kframe_loadcntrl_nlgeom.inp and kframe_dispcntrl_buckle_nlgeom.inp). Load versus horizontal deflection curves for the large-displacement analyses are shown in Figure 1.2.4–5. Similar to the model with switching algorithm, first element 7 buckles. As the other members deform and absorb the load no longer carried by the buckled member, the structure regains stiffness and plasticity develops in other members. When seven members develop plastic hinges, the structure can no longer support additional horizontal loading. The limit load in the third model is only about 28% of the limit load in the model without buckling.

The load-displacement curves for the switching algorithm and for the example with element 7 using buckling strut response compare well and are not shown.

Input files

kframe_loadcntrl_nlgeom.inp	Elastic-plastic analysis with load control; large-displacement analysis.
kframe_loadcntrl.inp	Elastic-plastic analysis with load control; small-displacement analysis.
kframe_dispcntrl_switch_nlgeom.inp	Elastic-plastic frame element with the switching algorithm and displacement control; large-displacement analysis.
kframe_dispcntrl_switch.inp	Elastic-plastic frame element with the switching algorithm and displacement control; small-displacement analysis.
kframe_dispcntrl_buckle_nlgeom.inp	Elastic-plastic and buckling strut response with load control; large-displacement analysis.
kframe_dispcntrl_buckle.inp	Elastic-plastic and buckling strut response with displacement control; small-displacement analysis.

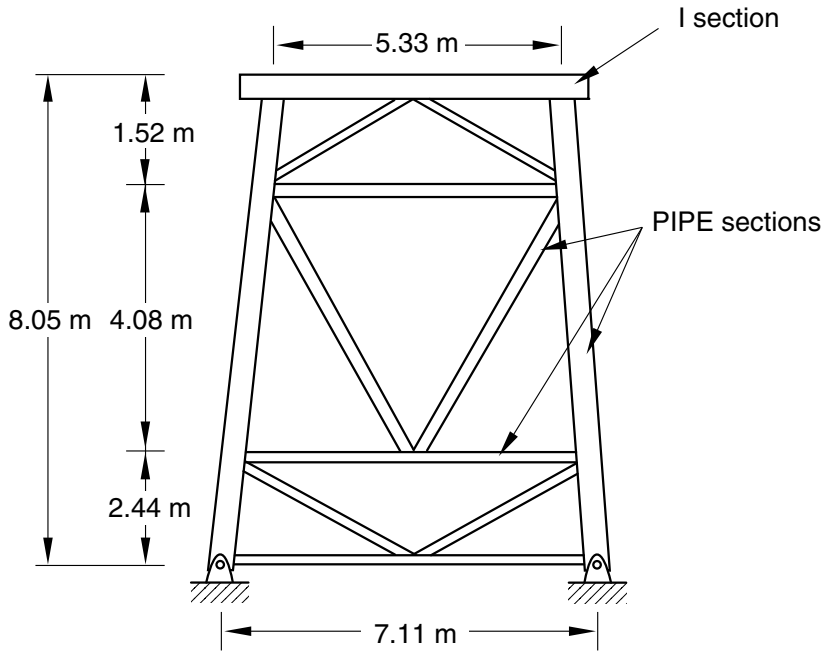


Figure 1.2.4–1 Two-dimensional K-frame structure.

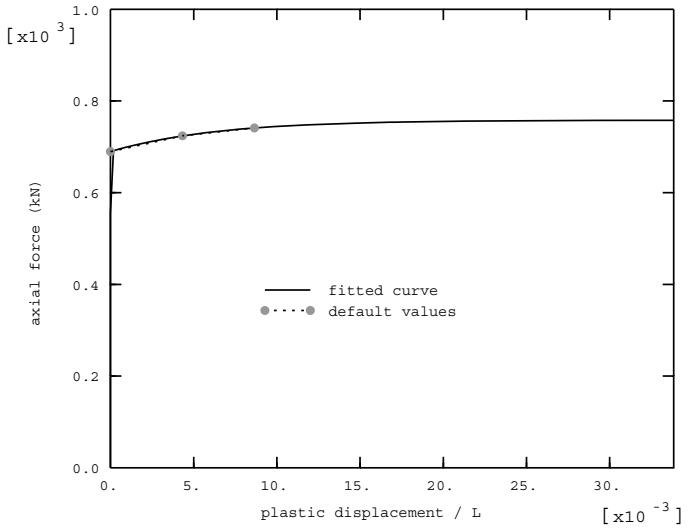


Figure 1.2.4–2 Default hardening response for axial force in a typical element with PIPE cross-section (element 7 in the model).

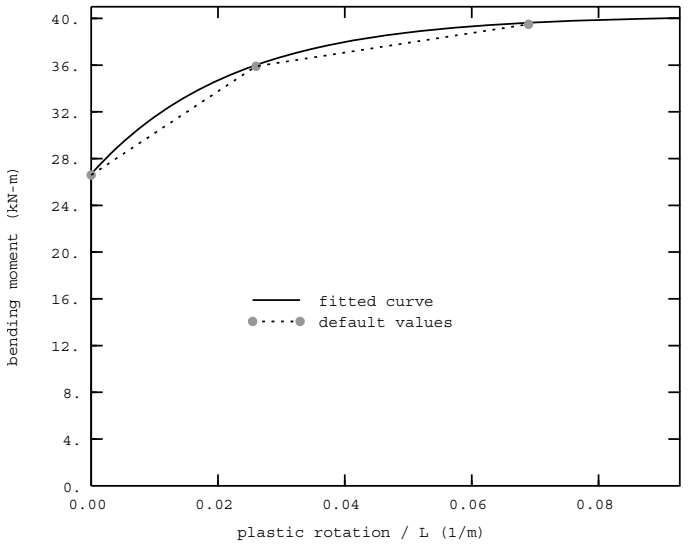


Figure 1.2.4–3 Default hardening response for bending moments in a typical element with PIPE cross-section (element 7 in the model).

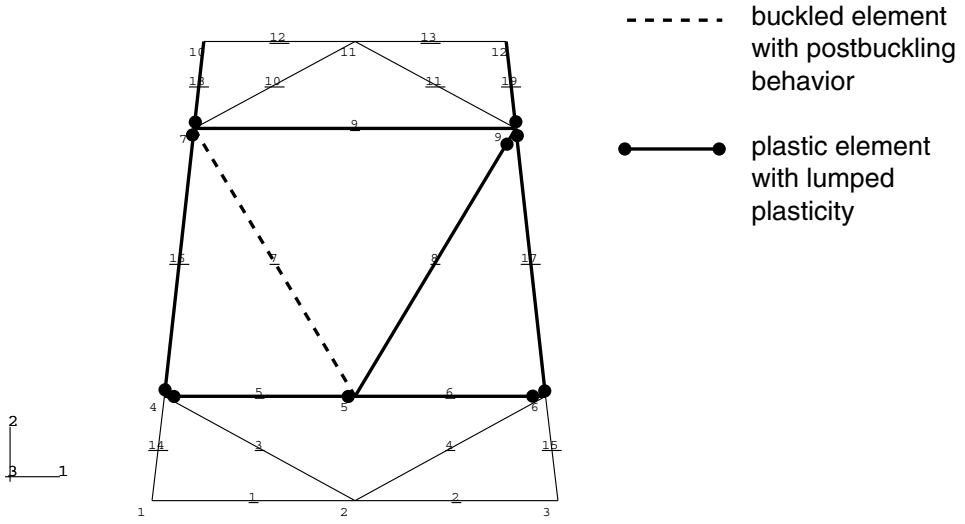


Figure 1.2.4–4 Results of analysis with switching algorithm: K-frame model with plastic and buckled elements.

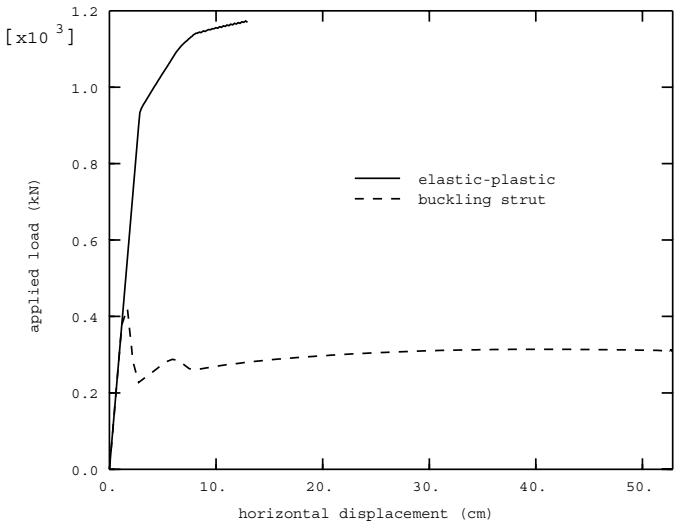


Figure 1.2.4–5 Applied force versus horizontal displacement of the load point for the elastic-plastic model and the model including buckling strut response.

1.2.5 UNSTABLE STATIC PROBLEM: REINFORCED PLATE UNDER COMPRESSIVE LOADS

Product: Abaqus/Standard

This example demonstrates the use of automatic techniques to stabilize unstable static problems. Geometrically nonlinear static problems can become unstable for a variety of reasons. Instability may occur in contact problems, either because of chattering or because contact intended to prevent rigid body motions is not established initially. Localized instabilities can also occur; they can be either geometrical, such as local buckling, or material, such as material softening.

This problem models a reinforced plate structure subjected to in-plane compressive loading that produces localized buckling. Structures are usually designed for service loads properly augmented by safety factors. However, it is quite often of interest to explore their behavior under extreme accident loads. This example looks into a submodel of a naval construction structure. It is a rectangular plate reinforced with beams in its two principal directions (Figure 1.2.5–1). The plate has symmetry boundary conditions along the longer edges and is pinned rigidly along the shorter sides. An in-plane load is applied to one of the pinned sides, compressing the plate. Gravity loads are also applied. The plate buckles under the load. The buckling is initially localized within each of the sections bounded by the reinforcements. At higher load levels the plate experiences global buckling in a row of sections closest to the applied load.

Standard analysis procedures typically provide the load at which the structure starts to buckle. The user may be interested in knowing the structure's additional load carrying capacity. This information could translate, for instance, into knowing when the onset of global buckling takes place or how far into the structure damage propagates. In such situations more sophisticated analysis techniques are necessary. Arc length methods such as the Riks method available in Abaqus are global load-control methods that are suitable for global buckling and postbuckling analyses; they do not function well when buckling is localized. Alternatives are to analyze the problem dynamically or to introduce damping. In the dynamic case the strain energy released locally from buckling is transformed into kinetic energy; in the damping case this strain energy is dissipated. To solve a quasi-static problem dynamically is typically an expensive proposition. In this example the automatic stabilization capability in Abaqus, which applies volume proportional damping to the structure, is used.

Geometry and model

The model consists of a rectangular plate 10.8 m (425.0 in) long, 6.75 m (265.75 in) wide, and 5.0 mm (0.2 in) thick. This plate has several reinforcements in both the longitudinal and transverse directions (Figure 1.2.5–1). The plate represents part of a larger structure: the two longitudinal sides have symmetry boundary conditions, and the two transverse sides have pinned boundary conditions. In addition, springs at two major reinforcement intersections represent flexible connections to the rest of the structure. The mesh consists of S4 shell elements for both the plate and larger reinforcements and additional S3 shell and B31 beam elements for the remaining reinforcements. The entire structure is made of the same construction steel, with an initial flow stress of 235.0 MPa (34.0 ksi).

To provide stability to the numerical solution upon the anticipated buckling, automatic stabilization based on volume proportional damping is added to the model. Two forms of automatic stabilization are considered: one with a constant damping factor that is chosen by default (see "Automatic stabilization of static problems with a constant damping factor" in "Solving nonlinear problems," Section 7.1.1 of the Abaqus Analysis User's Guide), and one with an adaptive damping factor (see "Adaptive automatic stabilization scheme" in "Solving nonlinear problems," Section 7.1.1 of the Abaqus Analysis User's Guide).

Results and discussion

The analysis consists of two steps. In the first step a gravity load perpendicular to the plane of the plate is applied. In the second step a longitudinal compressive load of 6.46×10^6 N (1.45×10^6 lbf) is applied to one of the pinned sides of the plate. All the nodes on that side are forced to move equally by means of multi-point constraints. The analysis is quasi-static, but buckling is expected.

Initially local out-of-plane buckling develops throughout the plate in an almost checkerboard pattern inside each one of the sections delimited by the reinforcements (Figure 1.2.5–2). Later, global buckling develops along a front of sections closer to the applied load (Figure 1.2.5–3). The evolution of the displacements produced by the applied load is very smooth (Figure 1.2.5–4) and does not reflect the early local instabilities in the structure. However, when the global instability develops, the curve becomes almost flat, indicating the complete loss of load carrying capacity. An inspection of the model's energy content (Figure 1.2.5–5 and Figure 1.2.5–6) reveals that while the load is increasing, the amount of dissipated energy is negligible. As soon as the load flattens out, the strain energy also flattens out (indicating a more or less constant load carrying capacity), while the dissipated energy increases dramatically to absorb the work done by the applied loads.

Figure 1.2.5–7 shows the ratio of the dissipated energy to the total strain energy obtained using a constant damping factor versus using an adaptive damping factor.

Acknowledgments

SIMULIA would like to thank IRCN (France) for providing this example.

Input files

unstablestatic_plate.inp	Plate model.
unstablestatic_plate_stabil_adap.inp	Plate model with adaptive stabilization.
unstablestatic_plate_node.inp	Node definitions for the plate model.
unstablestatic_plate_elem.inp	Element definitions for the plate model.

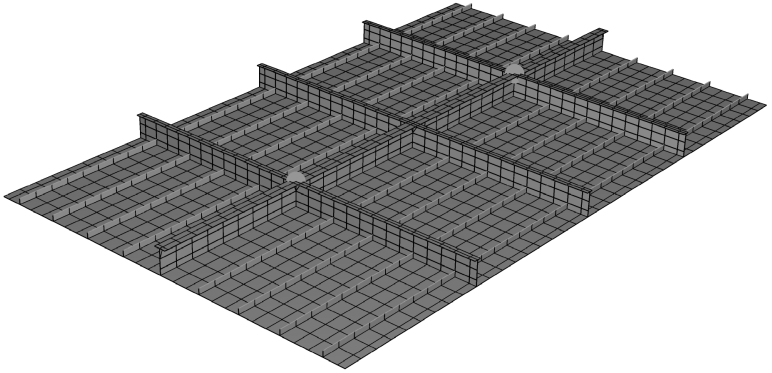


Figure 1.2.5–1 Reinforced plate initial mesh.

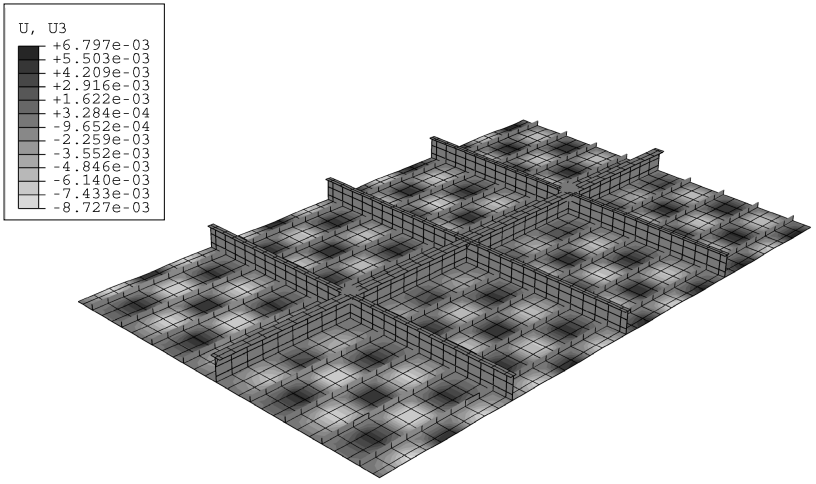


Figure 1.2.5–2 Plate localized buckling.

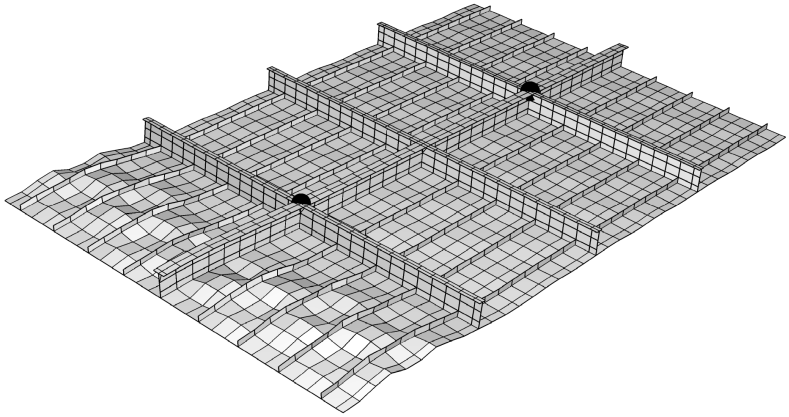


Figure 1.2.5–3 Plate global buckling.

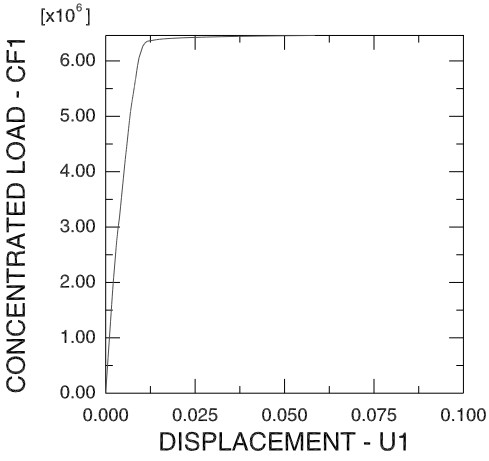


Figure 1.2.5–4 Plate load-displacement curve.

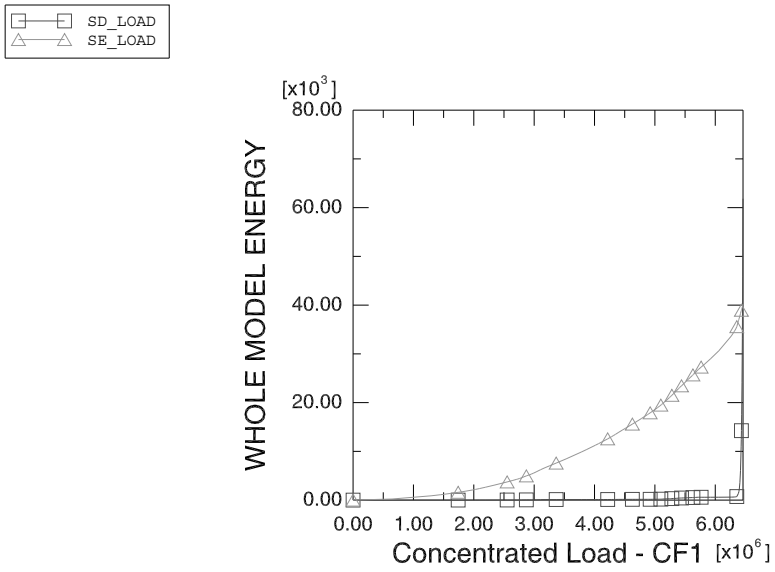


Figure 1.2.5–5 Dissipated and strain energies as functions of load.

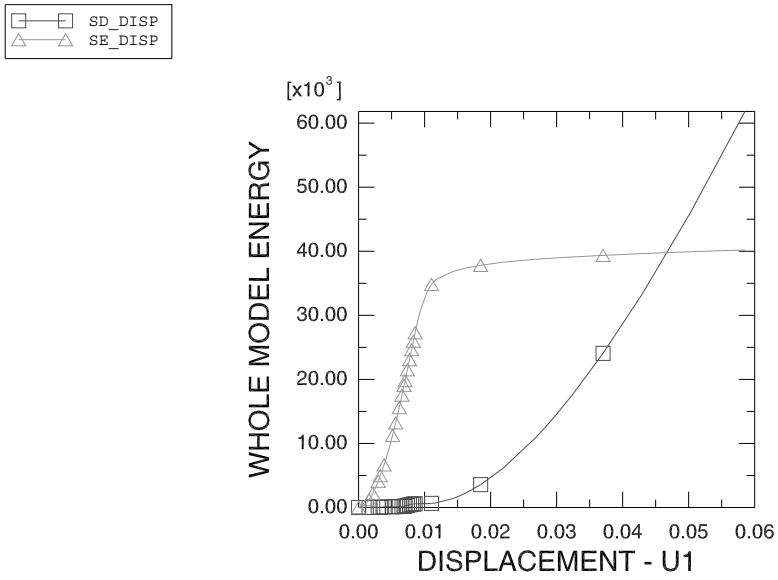


Figure 1.2.5–6 Dissipated and strain energies as functions of displacement.

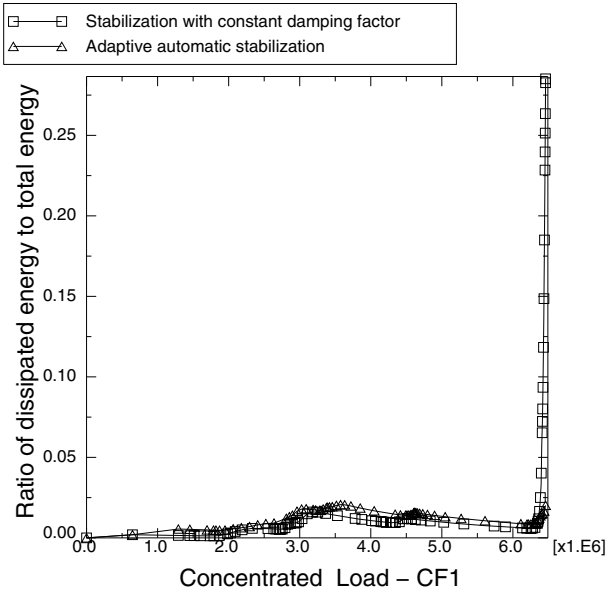


Figure 1.2.5–7 Ratio of dissipated energy to total strain energy using a constant damping factor and adaptive stabilization.

1.2.6 BUCKLING OF AN IMPERFECTION-SENSITIVE CYLINDRICAL SHELL

Product: Abaqus/Standard

This example serves as a guide to performing a postbuckling analysis using Abaqus for an imperfection-sensitive structure. A structure is imperfection sensitive if small changes in an imperfection change the buckling load significantly. Qualitatively, this behavior is characteristic of structures with closely spaced eigenvalues. For such structures the first eigenmode may not characterize the deformation that leads to the lowest buckling load. A cylindrical shell is chosen as an example of an imperfection-sensitive structure.

Geometry and model

The cylinder being analyzed is depicted in Figure 1.2.6–1. The cylinder is simply supported at its ends and is loaded by a uniform, compressive axial load. A uniform internal pressure is also applied to the cylinder. The material in the cylinder is assumed to be linear elastic. The thickness of the cylinder is 1/500 of its radius, so the structure can be considered to be a thin shell.

The finite element mesh uses the fully integrated S4 shell element. This element is based on a finite membrane strain formulation and is chosen to avoid hourglassing. A full-length model is used to account for both symmetric and antisymmetric buckling modes. A fine mesh, based on the results of a refinement study of the linear eigenvalue problem, is used. The convergence of the mesh density is based on the relative change of the eigenvalues as the mesh is refined. The mesh must have several elements along each spatial deformation wave; therefore, the level of mesh refinement depends on the modes with the highest wave number in the circumferential and axial directions.

Solution procedure

The solution strategy is based on introducing a geometric imperfection in the cylinder. In this study the imperfections are linear combinations of the eigenvectors of the linear buckling problem. If details of imperfections caused in a manufacturing process are known, it is normally more useful to use this information as the imperfection. However, in many instances only the maximum magnitude of an imperfection is known. In such cases assuming the imperfections are linear combinations of the eigenmodes is a reasonable way to estimate the imperfect geometry (Arbocz, 1987).

Determining the most critical imperfection shape that leads to the lowest collapse load of an axially compressed cylindrical shell is an open research issue. The procedure discussed in this example does not, therefore, claim to compute the lowest collapse load. Rather, this example discusses one approach that can be used to study the postbuckling response of an imperfection-sensitive structure.

The first stage in the simulation is a linear eigenvalue buckling analysis. To prevent rigid body motion, a single node is fixed in the axial direction. This constraint is in addition to the simply supported boundary conditions noted earlier and will not introduce an overconstraint into the problem since the axial load is equilibrated on opposing edges. The reaction force in the axial direction should be zero at this node.

The second stage involves introducing the imperfection into the structure using geometric imperfections. A single mode or a combination of modes is used to construct the imperfection. To compare the results obtained with different imperfections, the imperfection size must be fixed. The measure of the imperfection size used in this problem is the out-of-roundness of the cylinder, which is computed as the radial distance from the axis of the cylinder to the perturbed node minus the radius of the perfect structure. The scale factor associated with each eigenmode used to seed the imperfection is computed with a FORTRAN program. The program reads the results file produced by the linear analysis and determines the scale factors so that the out-of-roundness of the cylinder is equal to a specified value. This value is taken as a fraction of the cylinder thickness.

The final stage of the analysis simulates the postbuckling response of the cylinder for a given imperfection. The primary objective of the simulation is to determine the static buckling load. The modified Riks method is used to obtain a solution since the problem under consideration is unstable. The Riks method can also be used to trace the unstable and stable solution branches of a buckled structure. However, with imperfection-sensitive structures the first buckling mode is usually catastrophic, so further continuation of the analysis is usually not undertaken. When using a static Riks step, the tolerance used for the force residual convergence criteria may need to be tightened to ensure that the solution algorithm does not retrace its original loading path once the limit point is reached. Simply restricting the maximum arc length allowed in an increment is normally not sufficient.

Parametric study

There are two factors that significantly alter the buckling behavior: the shape of the imperfection and the size of the imperfection. A convenient way to investigate the effects of these factors on the buckling response is to use the parametric study capabilities of Abaqus. A Python script file is used to perform the study. The script executes the linear analysis, runs the FORTRAN routine to create an input file with a specified imperfection size, and finally executes the postbuckling analysis.

Before executing the script, copy the FORTRAN routine cylsh_maximp.f to your work directory using the Abaqus **fetch** command,

```
abaqus fetch job=cylsh_maximp.f
```

and compile it using the Abaqus **make** command,

```
abaqus make job=cylsh_maximp.f
```

Parametrized template input data are used to generate variations of the parametric study. The script allows the analyst to vary the eigenmodes used to construct the imperfection, out-of-roundness measure, cylindrical shell geometry (radius, length, thickness), mesh density, material properties (Young's modulus and Poisson's ratio), etc. The results presented in the following section, however, are based on an analysis performed with a single set of parameters.

Results and discussion

The results for both the linear eigenvalue buckling and postbuckling analyses are discussed below.

Linear eigenvalue buckling

The Lanczos eigensolver is used to extract the linear buckling modes. This solver is chosen because of its superior accuracy and convergence rate relative to wavefront solvers for problems with closely spaced eigenvalues. Table 1.2.6–1 lists the first 19 eigenvalues of the cylindrical shell. The eigenvalues are closely spaced with a maximum percentage difference of 1.3%.

The geometry, loading, and material properties of the cylindrical shell analyzed in this example are characterized by their axisymmetry. As a consequence of this axisymmetry the eigenmodes associated with the linear buckling problem will be either (1) axisymmetric modes associated with a single eigenvalue, including the possibility of eigenmodes that are axially symmetric but are twisted about the symmetry axis or (2) nonaxisymmetric modes associated with repeated eigenvalues (Wohlever, 1999). The nonaxisymmetric modes are characterized by sinusoidal variations (n-fold symmetry) about the circumference of the cylinder. For most practical engineering problems and as illustrated in Table 1.2.6–1, it is usually found that a majority of the buckling modes of the cylindrical shell are nonaxisymmetric.

The two orthogonal eigenmodes associated with each repeated eigenvalue span a two-dimensional space, and as a result any linear combination of these eigenmodes is also an eigenmode; i.e., there is no preferred direction. Therefore, while the shapes of the orthogonal eigenmodes extracted by the eigensolver will always be the same and span the same two-dimensional space, the phase of the modes is not fixed and might vary from one analysis to another. The lack of preferred directions has consequences with regard to any imperfection study based upon a linear combination of nonaxisymmetric eigenmodes from two or more distinct eigenvalues. As the relative phases of eigenmodes change, the shape of the resulting imperfection and, therefore, the postbuckling response, also changes. To avoid this situation, postprocessing is performed after the linear buckling analysis on each of the nonaxisymmetric eigenmode pairs to fix the phase of the eigenmodes before the imperfection studies are performed. The basic procedure involves calculating a scaling factor for each of the eigenvectors corresponding to a repeated eigenvalue so that their linear combination generates a maximum displacement of 1.0 along the global X-axis. This procedure is completely arbitrary but ensures that the postbuckling response calculations are repeatable.

For the sake of consistency the maximum radial displacement associated with a unique eigenmode is also scaled to 1.0. These factors are further scaled to satisfy the out-of-roundness criterion mentioned earlier.

Postbuckling response

The modes used to seed the imperfection are taken from the first 19 eigenmodes obtained in the linear eigenvalue buckling analysis. Different combinations are considered: all modes, unique eigenmodes, and pairs of repeated eigenmodes. An imperfection size (i.e., out-of-roundness) of 0.5 times the shell thickness is used in all cases. The results indicate that the cylinder buckles at a much lower load than the value predicted by the linear analysis (i.e., the value predicted using only the lowest eigenmode of the system). An imperfection based on mode 1 (a unique eigenmode) results in a buckling load of about 90% of the predicted value. When the imperfection was seeded with a combination of all modes (1–19), a buckling load of 33% of the predicted value was obtained. Table 1.2.6–2 lists the buckling

loads predicted by Abaqus (as a fraction of linear eigenvalue buckling load) when different modes are used to seed the imperfection.

The smallest predicted buckling load in this study occurs when using modes 12 and 13 to seed the imperfection, yet the results obtained when the imperfection is seeded using all 19 modes indicate that a larger buckling load can be sustained. One possible explanation for this is that the solution strategy used in this study (discussed earlier) involves using a fixed value for the out-of-roundness of the cylinder as a measure of the imperfection size. Thus, when multiple modes are used to seed the imperfection, the overall effect of any given mode is less than it would be if only that mode were used to seed the imperfection. The large number of closely spaced eigenvalues and innumerable combinations of eigenmodes clearly demonstrates the difficulty of determining the collapse load of structures such as the cylindrical shell. In practice, designing imperfection-sensitive structures against catastrophic failure usually requires a combination of numerical and experimental results as well as practical building experience.

The final deformed configuration shown in Figure 1.2.6–2 uses a displacement magnification factor of 5 and corresponds to using all the modes to seed the imperfection. Even though the cylinder appears to be very short, it can in fact be classified as a moderately long cylinder using the parameters presented in Chajes (1985). The cylinder exhibits thin wall wrinkling; the initial buckling shape can be characterized as dimples appearing on the side of the cylinder. The compression of the cylinder causes a radial expansion due to Poisson's effect; the radial constraint at the ends of the cylinder causes localized bending to occur at the ends. This would cause the shell to fold into an accordion shape. (Presumably this would be seen if self-contact was specified and the analysis was allowed to run further. This is not a trivial task, however, and modifications to the solution controls would probably be required. Such a simulation would be easier to perform with Abaqus/Explicit.) This deformed configuration is in accordance with the perturbed reference geometry, shown in Figure 1.2.6–3. To visualize the imperfect geometry, an imperfection size of 5.0 times the shell thickness (i.e., 10 times the value actually used in the analysis) was used to generate the perturbed mesh shown in this figure. The deformed configuration in the postbuckling analysis depends on the shape of the imperfection introduced into the structure. Seeding the structure with different combinations of modes and imperfection sizes produces different deformed configurations and buckling loads. As the results vary with the size and shape of the imperfection introduced into the structure, there is no solution to which the results from Abaqus can be compared.

The load-displacement curve for the case when the first 19 modes are used to seed the imperfection is shown in Figure 1.2.6–4. The figure shows the variation of the applied load (normalized with respect to the linear eigenvalue buckling load) versus the axial displacement of an end node. The peak load that the cylinder can sustain is clearly visible.

Input files

cylsh_buck.inp	Linear eigenvalue buckling problem.
cylsh_postbuck.inp	Postbuckling problem.
cylsh_maximp.f	FORTRAN program to compute the scaling factors for the imperfection size.
cylsh_script.psf	Python script to generate the parametrized input files.

References

- Arbocz, J., "Post-Buckling Behaviour of Structures: Numerical Techniques for More Complicated Structures," in *Lecture Notes in Physics*, Ed. H. Araki et al., Springer-Verlag, Berlin, 1987, pp. 84–142.
- Chajes, A., "Stability and Collapse Analysis of Axially Compressed Cylindrical Shells," in *Shell Structures: Stability and Strength*, Ed. R. Narayanan, Elsevier, New York, 1985, pp. 1–17.
- Wohlever, J. C., "Some Computational Aspects of a Group Theoretic Finite Element Approach to the Buckling and Postbuckling Analyses of Plates and Shells-of-Revolution," in *Computer Methods in Applied Mechanics and Engineering*, vol. 170, pp. 373–406, 1999.

Table 1.2.6–1 Eigenvalue estimates for the first 19 modes.

Mode number	Eigenvalue
1	11723
2, 3	11724
4, 5	11728
6, 7	11734
8, 9	11744
10, 11	11757
12, 13	11775
14, 15	11798
16, 17	11827
18, 19	11864

Table 1.2.6–2 Summary of predicted buckling loads.

Mode used to seed the imperfection	Normalized buckling load
1	0.902
2, 3	0.625
4, 5	0.480
6, 7	0.355
8, 9	0.351
10, 11	0.340
12, 13	0.306
14, 15	0.323
16, 17	0.411
18, 19	0.422
All modes (1–19)	0.325

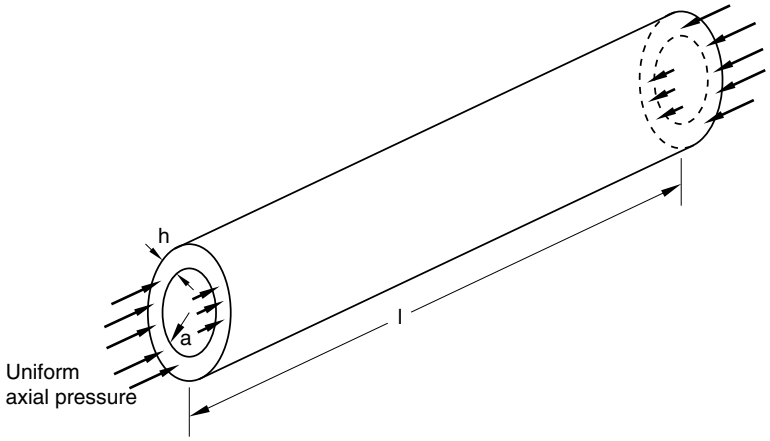


Figure 1.2.6–1 Cylindrical shell with uniform axial loading.

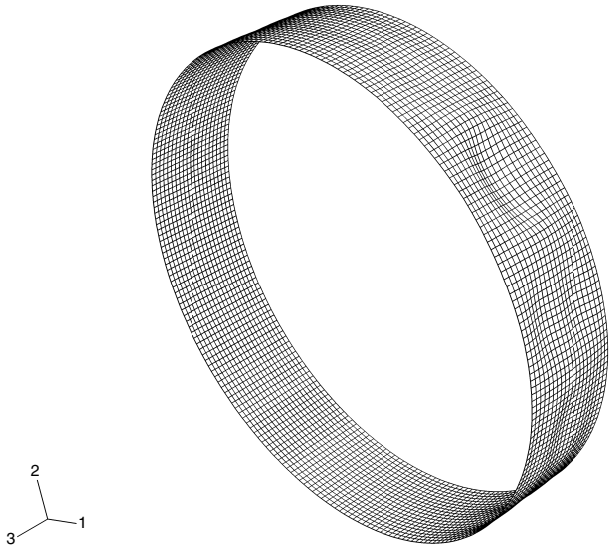


Figure 1.2.6–2 Final deformed configuration of the cylindrical shell (first 19 eigenmodes used to seed the imperfection; displacement magnification factor of 5.0; normalized end load = 0.29).



Figure 1.2.6–3 Perturbed geometry of the cylindrical shell (imperfection factor = 5 × thickness for illustration only; actual imperfection factor used = .5 × thickness).

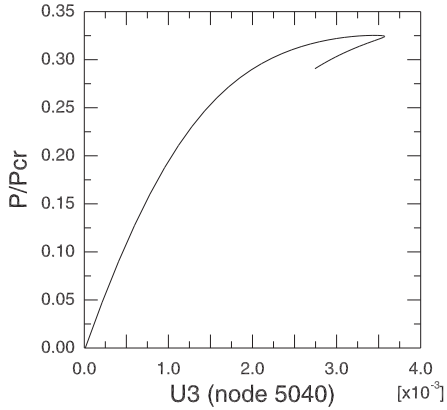


Figure 1.2.6–4 Normalized applied load versus axial displacement at node 5040 (first 19 modes used to seed the imperfection).

1.3 Forming analyses

- "Upsetting of a cylindrical billet: quasi-static analysis with mesh-to-mesh solution mapping (Abaqus/Standard) and adaptive meshing (Abaqus/Explicit)," Section 1.3.1
- "Superplastic forming of a rectangular box," Section 1.3.2
- "Stretching of a thin sheet with a hemispherical punch," Section 1.3.3
- "Deep drawing of a cylindrical cup," Section 1.3.4
- "Extrusion of a cylindrical metal bar with frictional heat generation," Section 1.3.5
- "Rolling of thick plates," Section 1.3.6
- "Axisymmetric forming of a circular cup," Section 1.3.7
- "Cup/trough forming," Section 1.3.8
- "Forging with sinusoidal dies," Section 1.3.9
- "Forging with multiple complex dies," Section 1.3.10
- "Flat rolling: transient and steady-state," Section 1.3.11
- "Section rolling," Section 1.3.12
- "Ring rolling," Section 1.3.13
- "Axisymmetric extrusion: transient and steady-state," Section 1.3.14
- "Two-step forming simulation," Section 1.3.15
- "Upsetting of a cylindrical billet: coupled temperature-displacement and adiabatic analysis," Section 1.3.16
- "Unstable static problem: thermal forming of a metal sheet," Section 1.3.17
- "Inertia welding simulation using Abaqus/Standard and Abaqus/CAE," Section 1.3.18
- "Moldflow translation examples," Section 1.3.19

1.3.1 UPSETTING OF A CYLINDRICAL BILLET: QUASI-STATIC ANALYSIS WITH MESH-TO-MESH SOLUTION MAPPING (Abaqus/Standard) AND ADAPTIVE MESHING (Abaqus/Explicit)

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

This example illustrates the use of the solution mapping capabilities of Abaqus/Standard and the adaptive meshing capabilities of Abaqus/Explicit in a metal forming application; the analysis results are compared with the results of Taylor (1981). The same problem is also analyzed using the coupled temperature-displacement elements in "Upsetting of a cylindrical billet: coupled temperature-displacement and adiabatic analysis," Section 1.3.16. Coupled temperature-displacement elements are included in this example only for solution mapping verification purposes; no heat generation occurs in these elements for this example.

When the strains become large in a geometrically nonlinear analysis, the elements often become so severely distorted that they no longer provide a good discretization of the problem. When this occurs, it is necessary to map the solution onto a new mesh that is better designed to continue the analysis. In Abaqus/Standard the procedure is to monitor the distortion of the mesh—for example, by observing deformed configuration plots—and decide when the mesh needs to be mapped. When mesh distortion is so severe that a new mesh must be created, the new mesh can be generated using the mesh generation options in Abaqus/CAE. The output database is useful in this context since the current geometry of the model can be extracted from the data in the output database. Once a new mesh is defined, the analysis is continued by beginning a new problem using the solution from the old mesh at the point of mapping as initial conditions by specifying the step number and increment number at which the solution should be read from the previous analysis. Abaqus/Standard interpolates the solution from the old mesh onto the new mesh to begin the new problem. This technique provides considerable generality. For example, the new mesh might be more dense in regions of high-strain gradients and have fewer elements in regions that are moving rigidly—there is no restriction that the number of elements be the same or that element types agree between the old and new meshes. In a typical practical analysis of a manufacturing process, mesh-to-mesh solution mapping may have to be done several times because of the large shape changes associated with such a process.

Abaqus/Explicit has capabilities that allow automatic solution mapping using adaptive meshing. Therefore, the mapping process is easier since it is contained within the analysis and the user only has to decide how frequently remeshing should be done and what method to use to map the solution from the old mesh to the new mesh as the solution progresses. Abaqus/Explicit offers default choices for adaptive meshing that have been shown to work for a wide variety of problems. Finally, solution-dependent meshing is used to concentrate mesh refinement areas of evolving boundary curvature. This counteracts the tendency of the basic smoothing methods to reduce the mesh refinement near concave boundaries where solution accuracy is important.

Geometry and model

The geometry is the standard test case of Lippmann (1979) and is defined in "Upsetting of a cylindrical billet: coupled temperature-displacement and adiabatic analysis," Section 1.3.16. It is a circular billet,

30 mm long, with a radius of 10 mm, compressed between two flat, rigid dies that are defined to be perfectly rough.

The mesh used to begin the analysis is shown in Figure 1.3.1–1. The finite element model is axisymmetric and includes the top half of the billet only since the middle surface of the billet is a plane of symmetry. In both the Abaqus/Standard and Abaqus/Explicit simulations, element type CAX4R is used: this is a 4-node quadrilateral with a single integration point and "hourglass control" to control spurious mechanisms caused by the fully reduced integration. The element is chosen here because it is relatively inexpensive for problems involving nonlinear constitutive behavior since the material calculations are only done at one point in each element. In addition, in the Abaqus/Standard simulations element types CGAX4R, CGAX4T, CAX4P, and CAX4I are also used to model the billet; in the Abaqus/Explicit simulations element type CAX6M is also used to model the billet.

The contact between the top and lateral exterior surfaces of the billet and the rigid die is modeled with a contact pair. The billet surface is specified as a surface definition in the model. The rigid die is modeled in a variety of different ways as described in Table 1.3.1–1. The mechanical interaction between the contact surfaces is assumed to be nonintermittent, rough frictional contact. Therefore, the contact property includes two additional specifications: rough friction to enforce a no-slip constraint between the two surfaces, and a no-separation contact pressure-overclosure relationship to ensure that separation does not occur once contact has been established. All the contact simulations in Abaqus/Standard use the node-to-surface formulation except one case where the surface-to-surface formulation is introduced.

Table 1.3.1–1 summarizes the different analysis cases that are studied. The column headings indicate whether the problem was analyzed using Abaqus/Standard and/or Abaqus/Explicit.

For Case 1 several different analyses are performed to compare the different section control options available in Abaqus/Explicit and to evaluate the effects of mesh refinement for the billet modeled with CAX4R elements. A coarse mesh (analysis COARSE_SS) and a fine mesh (analysis FINE_SS) are analyzed with the pure stiffness form of hourglass control. A coarse mesh (analysis COARSE_CS) is analyzed with the combined hourglass control. A coarse mesh (analysis COARSE_ENHS) and a fine mesh (analysis FINE_ENHS) are analyzed with the hourglass control based on the enhanced strain method. The default section controls, using the integral viscoelastic form of hourglass control, are tested on a coarse mesh (analysis COARSE) and a fine mesh (analysis FINE). Since this is a quasi-static analysis, the viscous hourglass control option should not be used. All other cases use the default section controls.

The Abaqus/Standard analyses for Case 1 compare the two hourglass control options and evaluate the effect of mesh refinement for the billet modeled with CAX4R elements. A coarse mesh (analysis COARSE_S) and a fine mesh (analysis FINE_S) are analyzed with the pure stiffness form of hourglass control. A coarse mesh (analysis COARSE_EH) and a fine mesh (analysis FINE_EH) are analyzed with hourglass control based on the enhanced strain method. A coarse mesh (analysis COARSE_EHG) with CGAX4R elements is also analyzed with hourglass control based on the enhanced strain method for comparison purposes.

No mesh convergence studies have been done, but the agreement with the results given in Lippmann (1979) suggests that the meshes used here are good enough to provide reasonable predictions of the overall force on the dies.

Material

The material model assumed for the billet is that given in Lippmann (1979). Young's modulus is 200 GPa, Poisson's ratio is 0.3, and the density is 7833 kg/m^3. A rate-independent von Mises elastic-plastic material model is used, with a yield stress of 700 MPa and a hardening slope of 0.3 GPa.

Boundary conditions and loading

The kinematic boundary conditions are symmetry on the axis (nodes at $r = 0$, in node set **AXIS**, have $u_r = 0$ prescribed) and symmetry about $z = 0$ (all nodes at $z = 0$, in node set **MIDDLE**, have $u_z = 0$ prescribed). The node on the top surface of the billet that lies on the symmetry axis is not part of the node set **AXIS** to avoid overconstraint: the radial motion of this node is already constrained by a no slip frictional constraint (see "Common difficulties associated with contact modeling in Abaqus/Standard," Section 39.1.2 of the Abaqus Analysis User's Guide, and "Common difficulties associated with contact modeling using contact pairs in Abaqus/Explicit," Section 39.2.2 of the Abaqus Analysis User's Guide).

In Abaqus/Standard the rigid die is displaced by −9 mm in the axial direction using a displacement boundary condition. In Abaqus/Explicit the u_z-displacement of the rigid die is prescribed using a velocity boundary condition whose value is ramped up to a velocity of 20 m/s and then held constant until the die has moved a total of 9 mm. The total simulation time of the Abaqus/Explicit analysis is 0.55 millisec, and the loading rate is slow enough to be considered quasi-static. In both Abaqus/Standard and Abaqus/Explicit the radial and rotational degrees of freedom of the rigid die are constrained.

For all cases the analyses are done in two steps so that the first step can be stopped at a die displacement corresponding to 44% upsetting; the second step carries the analysis to 60% upsetting. In the Abaqus/Standard simulations the solution mapping analysis restarts from the end of the first step with a new mesh and proceeds until 60% upsetting is achieved.

Mesh-to-mesh solution mapping in Abaqus/Standard

The interpolation technique used in solution mapping is a two-step process. First, values of all solution variables are obtained at the nodes of the old mesh by extrapolating the values from the integration points to the nodes of each element and averaging those values over all elements abutting each node. The second step is to locate each integration point in the new mesh with respect to the old mesh (this assumes all integration points in the new mesh lie within the bounds of the old mesh: warning messages are issued if this is not so, and new model solution variables at the integration point are set to zero). The variables are then interpolated from the nodes of the element in the old mesh to the location in the new mesh. All solution variables are interpolated automatically in this way so that the solution can proceed on the new mesh. Whenever a model is mapped, it can be expected that there will be some discontinuity in the solution because of the change in the mesh. If the discontinuity is significant, it is an indication that the meshes are too coarse or that the mapping should have been done at an earlier stage before too much distortion occurred.

Extracting two-dimensional profiles and remeshing using Abaqus/CAE

The model is built and meshed using Abaqus/CAE. The solution mapping for the Abaqus/Standard analysis is done by extracting the two-dimensional profile of the deformed billet from the output; the user must enter commands into the command line interface at the bottom of the Abaqus/CAE main window. To extract the deformed geometry from the output database as an orphan mesh part, use the command **`PartFromOdb`**, which takes the following arguments:

name	The name of the orphan mesh part to be created.
odb	The output database object returned from the command **`openOdb`**.
instance	The name of the part instance in the initial model in capital letters.
shape	Determines whether to import the part in its UNDEFORMED or DEFORMED shape.

The command **`PartFromOdb`** returns a Part object that is passed to the command **`Part2DGeomFrom2DMesh`**. This command creates a geometric Part object from the orphan mesh imported earlier. It takes the following arguments:

name	The name of the part to be created.
part	The part object returned from the command **`PartFromOdb`**.
featureAngle	A float specifying the angle (in degrees) between line segments that triggers a break in the geometry.

Once the profile of the deformed part has been created, the user can switch to the Mesh module, remesh the part, and write out the new node and element definitions to be used in the mapping analysis. The Python script file billet_rezone.py is included to demonstrate the process described above.

Adaptive meshing in Abaqus/Explicit

Adaptive meshing consists of two fundamental tasks: creating a new mesh, and remapping the solution variables from the old mesh to the new mesh with a process called advection. A new mesh is created at a specified frequency for each adaptive mesh domain. The mesh is found by sweeping iteratively over the adaptive mesh domain and moving nodes to smooth the mesh. The process of mapping solution variables from an old mesh to a new mesh is referred to as an advection sweep. At least one advection sweep is performed in every adaptive mesh increment. The methods used for advecting solution variables to the new mesh are consistent; monotonic; (by default) accurate to the second order; and conserve mass, momentum, and energy. This example problem uses the default settings for adaptive mesh domains.

Results and discussion

The following discussion focuses primarily on the results for Case 1, where the billet is modeled with CAX4R elements, the rigid die is modeled using an analytical rigid surface, and the pure stiffness hourglass control is used in Abaqus/Explicit. The deformed meshes at 44% billet upsetting (73.3% of the total die displacement) are shown in Figure 1.3.1–2, Figure 1.3.1–3, and Figure 1.3.1–4. The folding of the top outside surface of the billet onto the die is clearly visible. In Abaqus/Standard

(Figure 1.3.1–2) severe straining and element distortion can be seen through the center of the specimen. At this point the Abaqus/Standard mesh is mapped. The new mesh for the mapped model is shown in Figure 1.3.1–3. Figure 1.3.1–4 clearly indicates the benefits of adaptive meshing as the mesh used in Abaqus/Explicit has very little distortion.

The final configurations at 60% billet upsetting are shown in Figure 1.3.1–5 and Figure 1.3.1–6. Both the Abaqus/Standard and Abaqus/Explicit results compare well, and the meshes appear only slightly distorted. Similarily, the equivalent plastic strain magnitudes compare well (Figure 1.3.1–7 and Figure 1.3.1–8).

Figure 1.3.1–9 is a plot of upsetting force versus vertical displacement at the rigid surface reference node. The results of both the Abaqus/Standard and the Abaqus/Explicit analyses show excellent agreement with the rate-independent results obtained by Taylor (1981). Also worth noting is that the mapping in Abaqus/Standard does not appear to have a significant effect on the total upsetting force.

Figure 1.3.1–10 is a plot of upsetting force versus vertical displacement at the rigid surface reference node with the section control options identified in Table 1.3.1–2. The curves obtained using CAX4R and CAX6M elements are very close and agree well with the rate-independent results obtained by Taylor (1981). The results from the COARSE_SS analysis are virtually the same as the results from the FINE analysis but at a much reduced cost; therefore, such analysis options are recommended for this problem. The results for all the other cases (which use the default section controls but different rigid surface models) are the same as the results for Case 1 using the default section controls.

Input files

Abaqus/Standard input files

billet_case1_std_coarse.inp	Original COARSE CAX4R mesh using STIFFNESS hourglass control.
billet_coarse_nodes.inp	Node definitions for original COARSE mesh.
billet_coarse_elem.inp	Element definitions for original COARSE mesh.
billet_case1_std_coarse_rez.inp	Mapped COARSE CAX4R mesh.
billet_coarse_nodes_rez.inp	Node definitions for mapped COARSE mesh.
billet_coarse_elem_rez.inp	Element definitions for mapped COARSE mesh.
billet_case1_std_coarse_eh.inp	Original COARSE CAX4R mesh using ENHANCED hourglass control.
billet_case1_std_fine.inp	Original FINE CAX4R mesh using STIFFNESS hourglass control.
billet_case1_std_fine_rez.inp	Mapped FINE CAX4R mesh.
billet_case1_std_fine_eh.inp	Original FINE CAX4R mesh using ENHANCED hourglass control.
billet_case1_std_coarse_cax4i.inp	Original COARSE CAX4I mesh.
billet_case1_std_coarse_cax4i_surf.inp	Original COARSE CAX4I mesh using surface-to-surface contact formulation.
billet_case1_std_coarse_cax4i_rez.inp	Mapped COARSE CAX4I mesh.
billet_case1_std_coarse_cgax4r.inp	Original COARSE CGAX4R mesh.

billet_case1_std_coarse_cgax_eh.inp	Original COARSE CGAX4R mesh using ENHANCED hourglass control.
billet_case1_std_coarse_cgax4r_rez.inp	Mapped COARSE CGAX4R mesh.
billet_case1_std_coarse_cgax4t.inp	Original COARSE CGAX4T mesh.
billet_case1_std_coarse_cgax4t_rez.inp	Mapped COARSE CGAX4T mesh.
billet_case1_std_coarse_cax4p.inp	Original COARSE CAX4P mesh.
billet_case1_std_coarse_cax4p_rez.inp	Mapped COARSE CAX4P mesh.
billet_rezone.py	Python script showing an example of the command usage to extract the geometric profile of the deformed mesh from an output database.
billet_case2_std.inp	Original COARSE CAX4R mesh.
billet_case2_std_rez.inp	Mapped COARSE CAX4R mesh.
billet_case3_std.inp	Original COARSE CAX4R mesh.
billet_case3_std_rez.inp	Mapped COARSE CAX4R mesh.
billet_case6_std.inp	Original COARSE CAX4R mesh.
billet_case6_std_rez.inp	Mapped COARSE CAX4R mesh.

Abaqus/Explicit input files

billet_case1_xpl_coarse.inp	COARSE CAX4R mesh using RELAX STIFFNESS hourglass control.
billet_case1_xpl_coarse_ss.inp	COARSE CAX4R mesh using STIFFNESS hourglass control.
billet_case1_xpl_coarse_cs.inp	COARSE CAX4R mesh using COMBINED hourglass control.
billet_case1_xpl_coarse_enhs.inp	COARSE CAX4R mesh using ENHANCED hourglass control.
billet_case1_xpl_fine.inp	FINE CAX4R mesh using RELAX STIFFNESS hourglass control.
billet_case1_xpl_fine_ss.inp	FINE CAX4R mesh using STIFFNESS hourglass control.
billet_case1_xpl_fine_cs.inp	FINE CAX4R mesh using COMBINED hourglass control.
billet_case1_xpl_fine_enhs.inp	FINE CAX4R mesh using ENHANCED hourglass control.
billet_case1_xpl_coarse_cax6m.inp	COARSE CAX6M mesh using RELAX STIFFNESS hourglass control.
billet_case1_xpl_fine_cax6m.inp	FINE CAX6M mesh using RELAX STIFFNESS hourglass control.
billet_case2_xpl.inp	COARSE CAX4R mesh using RELAX STIFFNESS hourglass control.
billet_case3_xpl.inp	COARSE CAX4R mesh using RELAX STIFFNESS hourglass control.

billet_case4_xpl.inp	COARSE CAX4R mesh using RELAX STIFFNESS hourglass control.
billet_case5_xpl.inp	COARSE CAX4R mesh using RELAX STIFFNESS hourglass control.
billet_case6_xpl.inp	COARSE CAX4R mesh using RELAX STIFFNESS hourglass control.
billet_case7_xpl.inp	COARSE CAX4R mesh using RELAX STIFFNESS hourglass control.

References

- Lippmann, H., *Metal Forming Plasticity,* Springer-Verlag, Berlin, 1979.
- Taylor, L. M., "A Finite Element Analysis for Large Deformation Metal Forming Problems Involving Contact and Friction," Ph.D. Thesis, U. of Texas at Austin, 1981.

Table 1.3.1–1 Cases describing the modeling of the rigid die.

Case	Description	STD	XPL
1	The die is modeled as an analytical rigid surface using a planar analytical surface and a rigid body constraint. The rigid surface is associated with a rigid body by its specified reference node.	Yes	Yes
2	Axisymmetric rigid elements of type RAX2 are used to model the rigid die.	Yes	Yes
3	The die is modeled with RAX2 elements, as in Case 2. However, the die is assigned a mass by specifying point masses at the nodes of the RAX2 elements.	Yes	Yes
4	The rigid die is modeled with RAX2 elements, as in Case 2. The rigid elements are assigned a thickness and density values such that the mass of the die is the same as in Case 3.	No	Yes
5	The die is modeled with RAX2 elements, as in Case 2. In this case the thickness of the rigid elements is interpolated from the thickness specified at the nodes. The same thickness value is prescribed as in Case 4.	No	Yes
6	Axisymmetric shell elements of type SAX1 are used to model the die, and they are included in the rigid body definition.	Yes	Yes
7	The die is modeled with axisymmetric shell elements of type SAX1 and with axisymmetric rigid elements of type RAX2. The deformable elements are included in the rigid body definition. Both element types have the same thickness and density as in Case 4.	No	Yes

Table 1.3.1–2 Analysis options for Case 1 using CAX4R elements.

Analysis Label	Mesh Type	Hourglass Control	Analysis Type
COARSE_SS	coarse	stiffness	XPL
FINE_SS	fine	stiffness	XPL
COARSE_CS	coarse	combined	XPL
COARSE	coarse	relax stiffness	XPL
FINE	fine	relax stiffness	XPL
COARSE_ENHS	coarse	enhanced	XPL
FINE_ENHS	fine	enhanced	XPL
COARSE_S	coarse	stiffness	STD
FINE_S	fine	stiffness	STD
COARSE_EH	coarse	enhanced	STD
FINE_EH	fine	enhanced	STD
COARSE_EHG	coarse	enhanced	STD

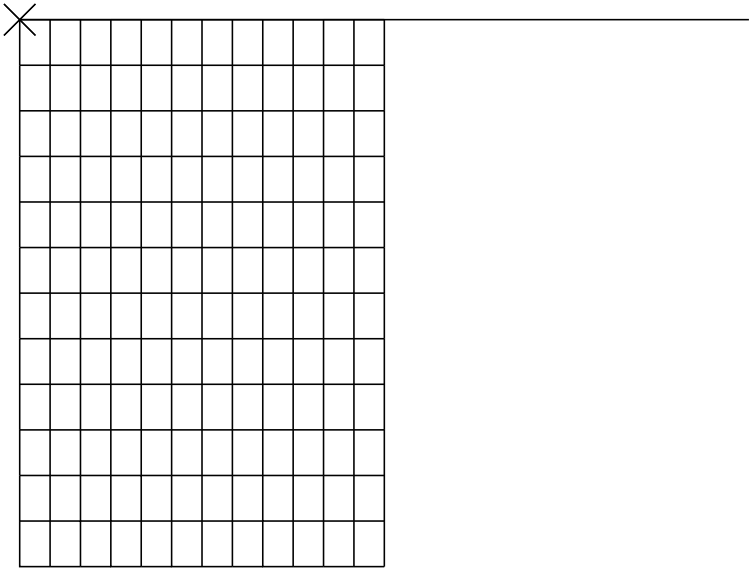


Figure 1.3.1–1 Axisymmetric upsetting example: initial mesh.

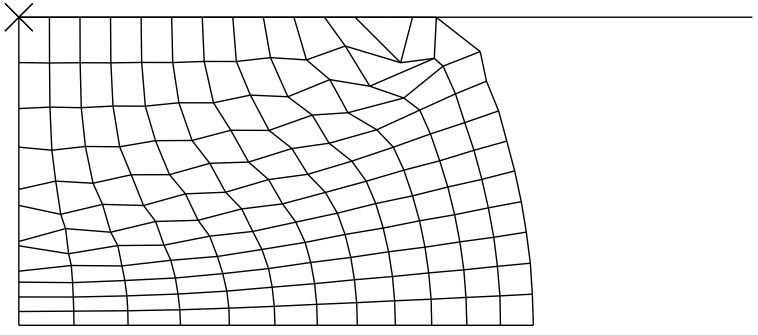


Figure 1.3.1–2 Abaqus/Standard: Deformed configuration at 44% upset (original mesh).

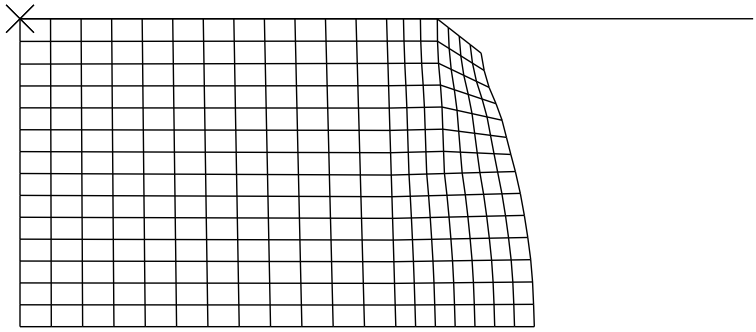


Figure 1.3.1–3 Abaqus/Standard: New mesh at 44% upset.

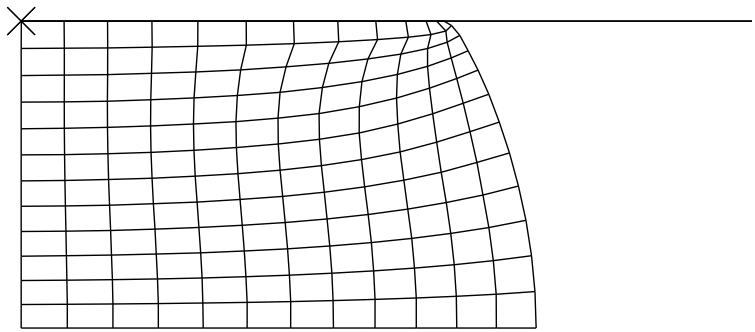


Figure 1.3.1–4 Abaqus/Explicit: Deformed configuration at 44% upset (CAX4R elements).

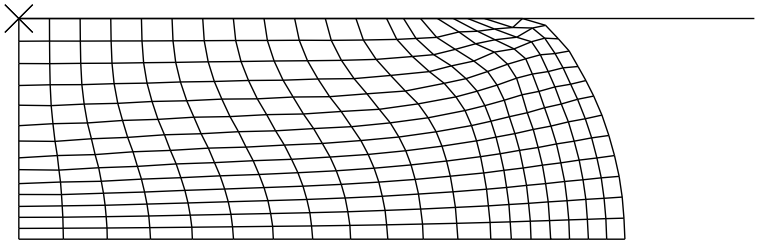


Figure 1.3.1–5 Abaqus/Standard: New mesh at 60% upset.

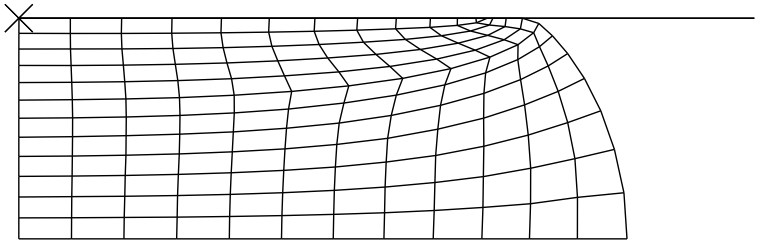


Figure 1.3.1–6 Abaqus/Explicit: Deformed mesh at 60% upset (CAX4R elements).

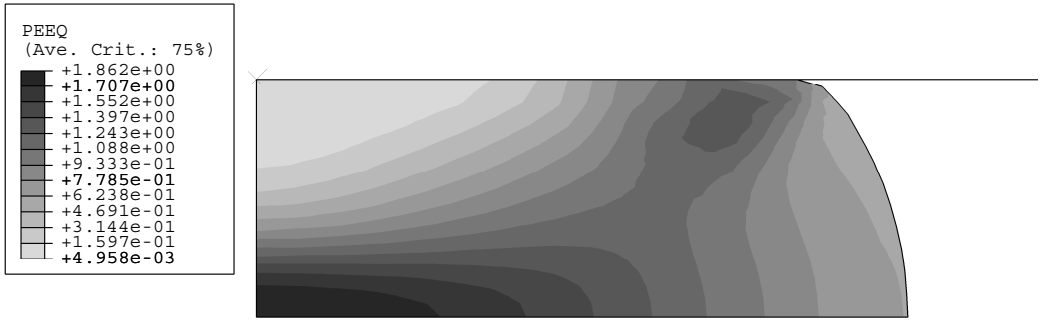


Figure 1.3.1–7 Abaqus/Standard: Plastic strain of new mesh at 60% upset.

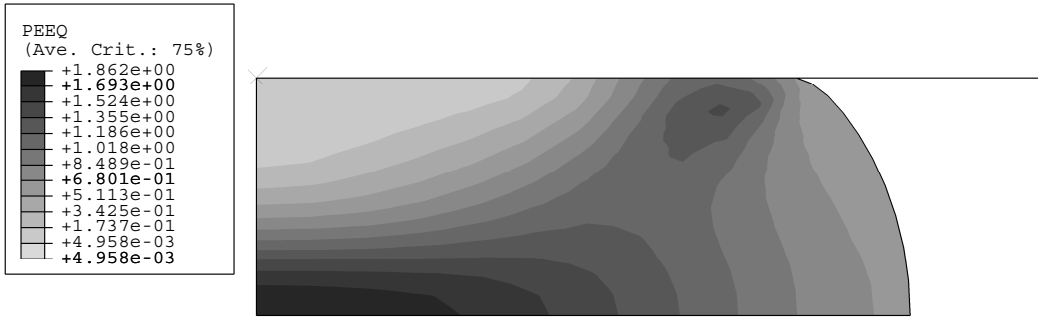


Figure 1.3.1–8 Abaqus/Explicit: Plastic strain at 60% upset (CAX4R elements).

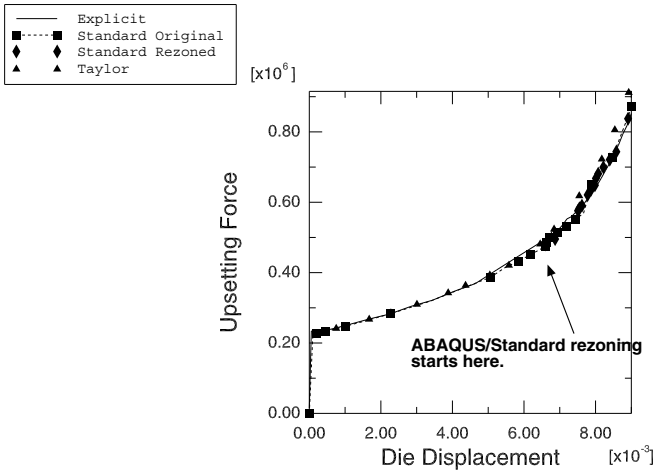


Figure 1.3.1–9 Force-deflection response for cylinder upsetting.

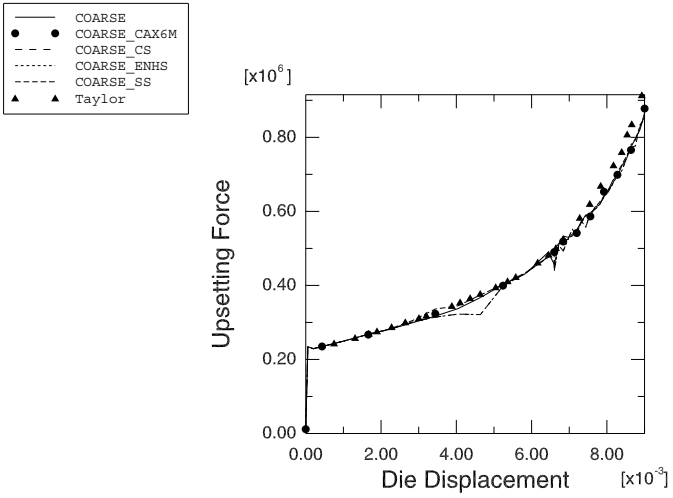


Figure 1.3.1–10 Force-deflection response for cylinder upsetting. Comparison of Abaqus/Explicit hourglass controls.

1.3.2 SUPERPLASTIC FORMING OF A RECTANGULAR BOX

Product: Abaqus/Standard

In this example we consider the superplastic forming of a rectangular box. The example illustrates the use of rigid elements to create a smooth three-dimensional rigid surface.

Superplastic metals exhibit high ductility and very low resistance to deformation and are, thus, suitable for forming processes that require very large deformations. Superplastic forming has a number of advantages over conventional forming methods. Forming is usually accomplished in one step rather than several, and intermediate annealing steps are usually unnecessary. This process allows the production of relatively complex, deep-shaped parts with quite uniform thickness. Moreover, tooling costs are lower since only a single die is usually required. Drawbacks associated with this method include the need for tight control of temperature and deformation rate. Very long forming times make this method impractical for high volume production of parts.

A superplastic forming process usually consists of clamping a sheet against a die whose surface forms a cavity of the shape required. Gas pressure is then applied to the opposite surface of the sheet, forcing it to acquire the die shape.

Rigid surface

A rigid faceted surface can be created from an arbitrary mesh of three-dimensional rigid elements (either triangular R3D3 or quadrilateral R3D4 elements). See "Analytical rigid surface definition," Section 2.3.4 of the Abaqus Analysis User's Guide, for a discussion of smoothing of master surfaces. Abaqus automatically smoothes any discontinuous surface normal transitions between the surface facets.

Solution-dependent amplitude

One of the main difficulties in superplastically forming a part is the control of the processing parameters. The temperature and the strain rates that the material experiences must remain within a certain range for superplasticity to be maintained. The former is relatively easy to achieve. The latter is more difficult because of the unknown distribution of strain rates in the part. The manufacturing process must be designed to be as rapid as possible without exceeding a maximum allowable strain rate at any material point. For this purpose Abaqus has a feature that allows the loading (usually the gas pressure) to be controlled by means of a solution-dependent amplitude and a target maximum creep strain rate. In the loading options the user specifies a reference value. The amplitude definition requires an initial, a minimum, and a maximum load multiplier. During a quasi-static procedure Abaqus will then monitor the maximum creep strain rate and compare it with the target value. The load amplitude is adjusted based on this comparison. This controlling algorithm is simple and relatively crude. The purpose is not to follow the target values exactly but to obtain a practical loading schedule.

Geometry and model

The example treated here corresponds to superplastic forming of a rectangular box whose final dimensions are 1524 mm (60 in) long by 1016 mm (40 in) wide by 508 mm (20 in) deep with a 50.8 mm (2 in) flange around it. All fillet radii are 101.6 mm (4 in). The box is formed by means of a uniform fluid pressure.

A quarter of the blank is modeled using 704 membrane elements of type M3D4. These are fully integrated bilinear membrane elements. The initial dimensions of the blank are 1625.6 mm (64 in) by 1117.6 mm (44 in), and the thickness is 3.175 mm (0.125 in). The blank is clamped at all its edges. The flat initial configuration of the membrane model is entirely singular in the normal direction unless it is stressed in biaxial tension. This difficulty is prevented by applying a small biaxial initial stress of 6.89 kPa (1 lb/in^2) by means of the initial stress conditions.

The female die is modeled as a rigid body and is meshed with rigid R3D3 elements. The rigid surface can be defined by grouping together those faces of the 231 R3D3 elements used to model the die that face the contact direction. See Figure 1.3.2–1 for an illustration of the rigid element mesh.

To avoid having points "fall off" the rigid surface during the course of the analysis, more than a quarter of the die has been modeled, as shown in Figure 1.3.2–2. It is always a good idea to extend the rigid surface far enough so that contacting nodes will not slide off the master surface.

By default, Abaqus generates a unique normal to the rigid surface at each node point, based on the average of the normals to the elements sharing each node. There are times, however, when the normal to the surface should be specified directly. This is discussed in "Node definition," Section 2.1.1 of the Abaqus Analysis User's Guide. In this example the flange around the box must be flat to ensure compatibility between the originally flat blank and the die. Therefore, an outer normal (0, 1, 0) has been specified at the 10 nodes that make up the inner edge of the flange. This is done by entering the direction cosines after the node coordinates. The labels of these 10 vertices are 9043, 9046, 9049, 9052, 9089, 9090, 9091, 9121, 9124, and 9127; and their definitions can be found in superplasticbox_node.inp.

Material

The material in the blank is assumed to be elastic-viscoplastic, and the properties roughly represent the 2004 (Al-6Cu-0.4Zr)-based commercial superplastic aluminum alloy Supral 100 at 470°C. It has a Young's modulus of 71 GPa (10.3×10^6 lb/in^2) and a Poisson's ratio of 0.34. The flow stress is assumed to depend on the plastic strain rate according to

$$\sigma^f = A(\dot{\varepsilon}^{pl})^{1/2},$$

where A is 179.2 MPa ($26. \times 10^3$ lb/in^2) and the time is in seconds.

Loading and controls

We perform two analyses to compare constant pressure loading and a pressure schedule automatically adjusted to achieve a maximum strain rate of 0.02/sec. In the constant load case the prestressed blank is subjected to a rapidly applied external pressure of 68.8 kPa (10 lb/in^2), which is then held constant for

3000 sec until the box has been formed. In the second case the prestressed blank is subjected to a rapidly applied external pressure of 1.38 kPa (0.2 lb/in^2). The pressure schedule is then chosen by Abaqus.

The initial application of the pressures is assumed to occur so quickly that it involves purely elastic response. This is achieved by using the static procedure. The creep response is developed in a second step using a quasi-static procedure.

During the quasi-static step an accuracy tolerance controls the time increment and, hence, the accuracy of the transient creep solution. Abaqus compares the equivalent creep strain rate at the beginning and the end of an increment. The difference should be less than this tolerance divided by the time increment. Otherwise, the increment is reattempted with a smaller time increment. The usual guideline for setting this accuracy tolerance is to decide on an acceptable error in stress and convert it to an error in strain by dividing by the elastic modulus. For this problem we assume that moderate accuracy is required and choose this tolerance as 0.5%. In general, larger tolerance values allow Abaqus to use larger time increments, resulting in a less accurate and less expensive analysis.

In the automatic scheduling analysis the pressure refers to an amplitude that allows for a maximum pressure of 1.38 MPa (200 lb/in^2) and a minimum pressure of 0.138 kPa (0.02 lb/in^2). The target creep strain rate is a constant entered using creep strain rate control. The node-to-surface (default) and surface-to-surface contact formulations are considered for this case where the creep strain rate is used to control the pressure amplitude. In the node-to-surface contact formulation the thickness of the blank is ignored and the blank is positioned such that its midsurface is used in the contact calculations as an approximation. However, the surface-to-surface contact formulation explicitly accounts for the blank thickness, as in reality.

Results and discussion

Figure 1.3.2–3 through Figure 1.3.2–5 show a sequence of deformed configurations during the automatically controlled forming process. The stages of deformation are very similar in the constant load process. However, the time necessary to obtain the deformation is much shorter with automatic loading—the maximum allowable pressure is reached after 83.3 seconds. The initial stages of the deformation correspond to inflation of the blank because there is no contact except at the edges of the box. Contact then occurs at the box's bottom, with the bottom corners finally filling. Although there is some localized thinning at the bottom corners, with strains on the order of 100%, these strains are not too much larger than the 80% strains seen on the midsides.

Figure 1.3.2–6 and Figure 1.3.2–7 show the equivalent plastic strain at the end of the process using the surface-to-surface and the node-to-surface contact formulations, respectively. The differences in the results are primarily due to differences in the way that the blank thickness is handled. Consequently, results from the surface-to-surface contact formulation are more reliable since the blank thickness is considered.

Figure 1.3.2–8 shows the evolution in time of the ratio between the maximum creep strain rate found in the model and the target creep strain rate for the two contact formulations. The load applied initially produces a low maximum creep strain rate at the beginning of the analysis. At the end the maximum creep strain rate falls substantially as the die cavity fills up. Although the curve appears very jagged, it indicates that the maximum peak strain rate is always relatively close to the target value. This is quite acceptable in practice. Figure 1.3.2–9 shows the pressure schedule that Abaqus calculates for this problem. For

most of the time, while the sheet does not contact the bottom of the die, the pressure is low. Once the die starts restraining the deformation, the pressure can be increased substantially without producing high strain rates. Again, the differences in the pressure schedule toward the end of the simulation are due primarily to the differences in the handling of the blank thickness.

Input files

superplasticbox_constpress.inp	Constant pressure main analysis.
superplasticbox_autopress.inp	Automatic pressurization main analysis.
superplasticbox_autopress_surf.inp	Automatic pressurization main analysis using the surface-to-surface contact formulation.
superplasticbox_node.inp	Node definitions for the rigid elements.
superplasticbox_element.inp	Element definitions for the rigid R3D3 elements.

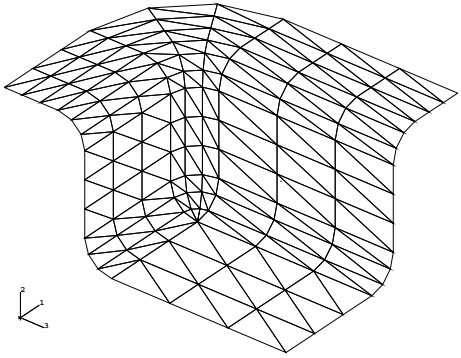


Figure 1.3.2–1 Rigid surface for die.

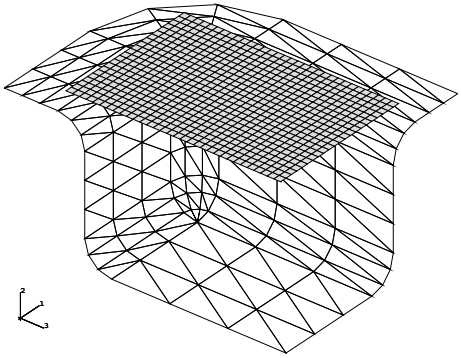


Figure 1.3.2–2 Initial position of blank with respect to die.

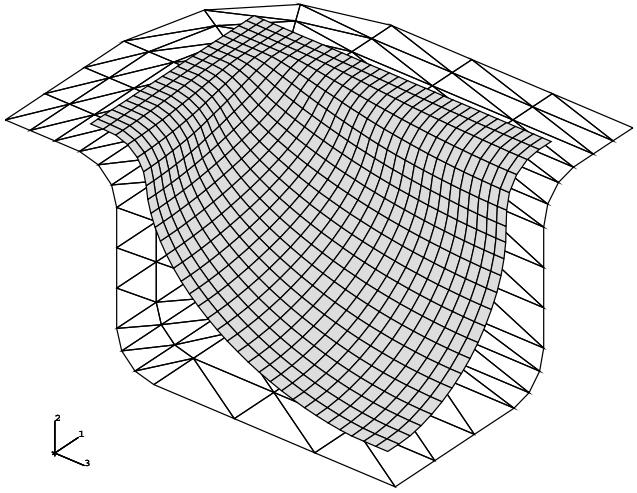


Figure 1.3.2–3 Automatic loading: deformed configuration after 34 sec in Step 2 using the node-to-surface contact formulation.

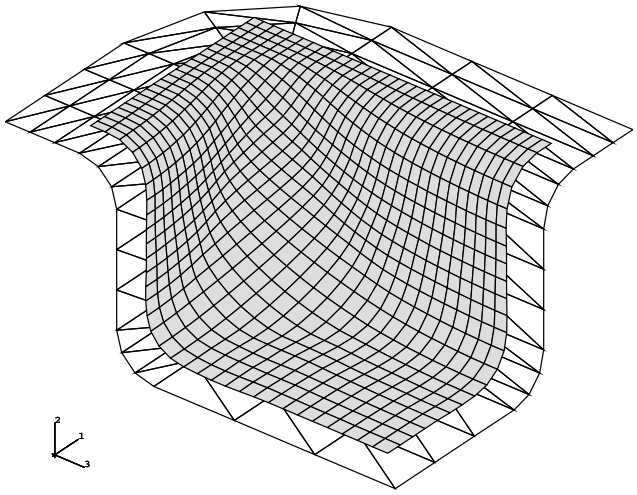


Figure 1.3.2–4 Automatic loading: deformed configuration after 63 sec in Step 2 using the node-to-surface contact formulation.

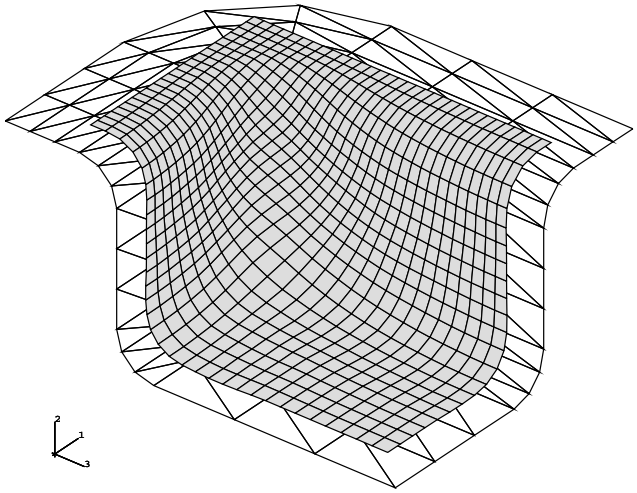


Figure 1.3.2–5 Automatic loading: deformed configuration after 83 sec in Step 2 using the node-to-surface contact formulation.

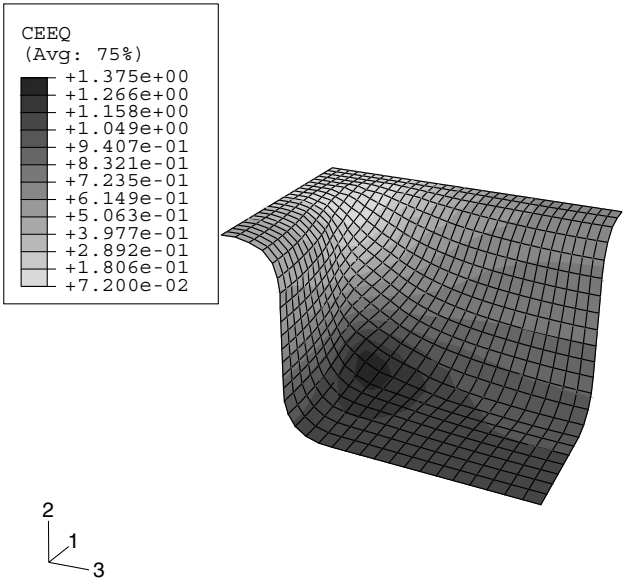


Figure 1.3.2–6 Automatic loading: inelastic strain in the formed box using the surface-to-surface contact formulation.

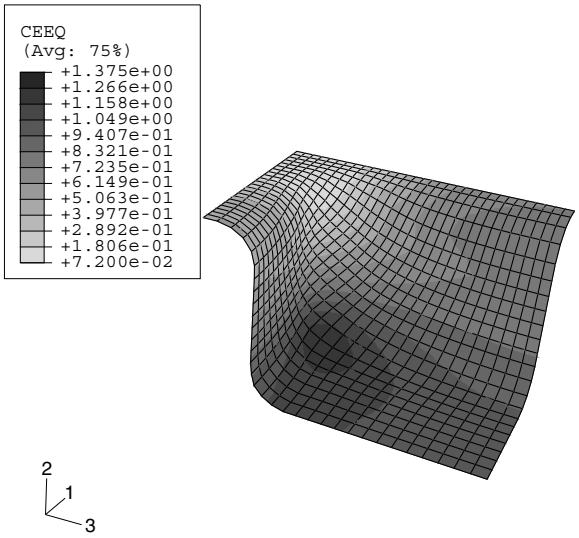


Figure 1.3.2–7 Automatic loading: inelastic strain in the formed box using the node-to-surface contact formulation.

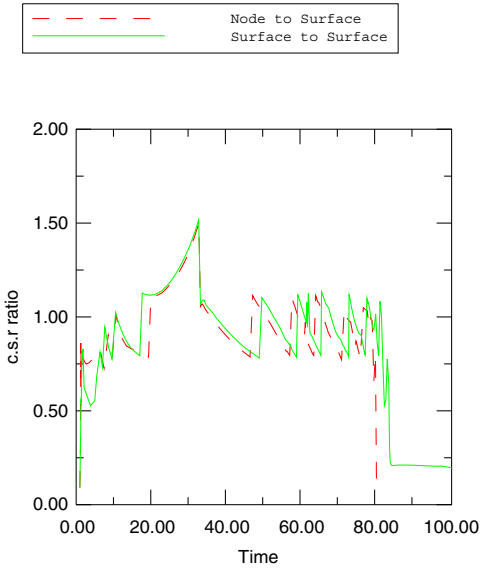


Figure 1.3.2–8 History of ratio between maximum creep strain rate and target creep strain rate.

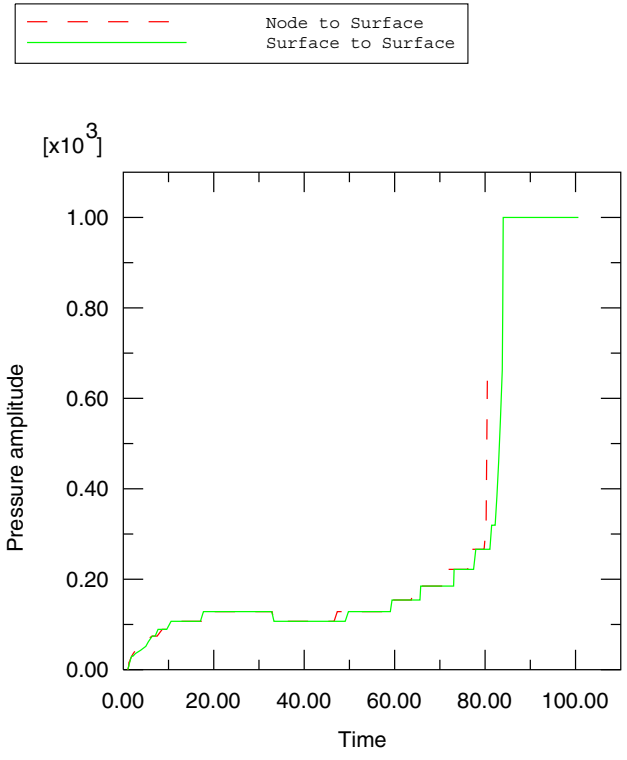


Figure 1.3.2–9 History of pressure amplitude.

1.3.3 STRETCHING OF A THIN SHEET WITH A HEMISPHERICAL PUNCH

Products: Abaqus/Standard Abaqus/Explicit

Stamping of sheet metals by means of rigid punches and dies is a standard manufacturing process. In most bulk forming processes the loads required for the forming operation are often the primary concern. However, in sheet forming the prediction of strain distributions and limit strains (which define the onset of local necking) are most important. Such analysis is complicated in that it requires consideration of large plastic strains during deformation, an accurate description of material response including strain hardening, the treatment of a moving boundary that separates the region in contact with the punch head from the unsupported one, and the inclusion of friction between the sheet and the punch head.

The stretching of a thin circular sheet with a hemispherical punch is considered in this example.

Geometry and model

The geometry of this problem is shown in Figure 1.3.3–1. The sheet being stretched has a clamping radius, r_0, of 59.18 mm. The radius of the punch, r_p, is 50.8 mm; the die radius, r_d, is 6.35 mm; and the initial thickness of the sheet, t_0, is 0.85 mm. Such a sheet has been tested experimentally by Ghosh and Hecker (1975) and has been analyzed by Wang and Budiansky (1978) using an axisymmetric membrane shell finite element formulation. The analysis is conducted statically in Abaqus/Standard and dynamically in Abaqus/Explicit such that inertial forces are relatively small. The initial configuration for the analysis is shown in Figure 1.3.3–2.

The sheet, the punch, and the die are modeled as separate parts, each instanced once. As an axisymmetric problem in Abaqus/Standard the sheet is modeled using 50 elements of type SAX1 (or MAX1) or 25 elements of type SAX2 (or MAX2). The Abaqus/Explicit model uses 50 elements of type SAX1. Mesh convergence studies (not reported here) have been done and indicate that these meshes give acceptably accurate results for most of the values of interest. To test the three-dimensional membrane and shell elements in Abaqus/Standard, a 10° sector is modeled using 100 elements of type S4R, S4, SC8R, or M3D4R or 25 elements of type M3D9R. All these meshes are reasonably fine; they are used to obtain good resolution of the moving contact between the sheet and the dies. In the Abaqus/Standard shell models nine integration points are used through the thickness of the sheet to ensure the development of yielding and elastic-plastic bending response; in Abaqus/Explicit five integration points are used through the thickness of the sheet.

The rigid punch and die are modeled in Abaqus/Standard as analytical rigid surfaces with a surface in conjunction with a rigid body constraint. The top and bottom surfaces of the sheet are defined with surface definitions. In Abaqus/Explicit the punch and die are modeled as rigid bodies; the surface of the punch and die are modeled either by analytical rigid surfaces or RAX2 elements. In the Abaqus/Explicit analyses the rigid surfaces are offset from the blank by half the thickness of the blank because the contact algorithm in Abaqus/Explicit takes the shell thickness into account. Similarly, when the surface-to-surface contact formulation in Abaqus/Standard is used, the blank thickness is considered by default and the rigid surfaces are offset from the blank by half the blank thickness, consistent with the physical reality. However, most of the input files presented in this section use the node-to-surface contact formulation in

Abaqus/Standard. In these cases the shell thickness is ignored, and the mid-surface of the shell is used in the contact calculations as an approximation.

Material properties

The material (aluminum-killed steel) is assumed to satisfy the Ramberg-Osgood relation between true stress and logarithmic strain,

$$\varepsilon = \sigma/E + (\sigma/K)^n,$$

where Young's modulus, E, is 206.8 GPa; the reference stress value, K, is 0.510 GPa; and the work-hardening exponent, n, is 4.76. In the present Abaqus analyses the Ramberg-Osgood relation is approximated using an elastic-plastic material. The material is assumed to be linear elastic up to a yield stress of 170.0 MPa, and the stress-strain curve beyond the yield stress is defined in piecewise linear segments. Poisson's ratio is 0.3.

The membrane element models in Abaqus/Standard are inherently unstable in a static analysis unless some prestress is present in the elements prior to the application of external loading. Therefore, an equibiaxial initial stress condition equal to 5% of the initial yield stress is prescribed for the membrane elements in Abaqus/Standard.

Contact interactions

The contact between the sheet and the rigid punch and the rigid die is modeled with a contact pair. The mechanical interaction between the contact surfaces is assumed to be frictional contact, with a coefficient of friction of 0.275.

Loading

The Abaqus/Standard analysis is carried out in five steps; the Abaqus/Explicit analysis is carried out in four steps. In Abaqus/Explicit the velocity of the punch head is prescribed using a velocity boundary condition whose amplitude is ramped up to 30 m/s at 1.24 milliseconds during the first step and then kept constant until time reaches 1.57 milliseconds at the end of the second step. It is then ramped down to zero at a time of 1.97 milliseconds at the end of the third step. In the first three steps of the Abaqus/Standard and Abaqus/Explicit analyses, the punch head is moved toward the sheet through total distances of 18.6 mm, 28.5 mm, and 34.5 mm, respectively. The purpose of these three steps is to compare the results with those provided experimentally by Ghosh and Hecker for these punch displacements. More typically the punch would be moved through its entire travel in one step.

Two final steps are included in the Abaqus/Standard analysis. In the first step the metal sheet is held in place and the contact pairs are removed from the model. In the second step the original boundary conditions for the metal sheet are reintroduced for springback analysis. However, this springback step is not included for the analyses using membrane elements, since these elements do not have any bending stiffness and residual bending stress is often a key determinant of springback.

In the final step of the Abaqus/Explicit analysis the punch head is moved away from the sheet for springback analysis. A viscous pressure load is applied to the surface of the shell during this step to damp

out transient wave effects so that quasi-static equilibrium can be reached quickly. This effect happens within approximately 2 milliseconds from the start of unloading. The coefficient of viscous pressure is chosen to be 0.35 MPa sec/m, approximately 1% of the value of ρc_d, where ρ is the material density of the sheet and c_d is the dilatational wave speed. A value of viscous pressure of ρc_d would absorb all the energy in a pressure wave. For typical structural problems choosing a small percentage of this value provides an effective way of minimizing ongoing dynamic effects. Static equilibrium is reached when residual stresses in the sheet are reasonably constant over time.

Results and discussion

Figure 1.3.3–2 shows the initial, undeformed profile of the blank, the die, and the punch. Figure 1.3.3–3 illustrates the deformed sheet and the punch and the die. Figure 1.3.3–4 shows a plot of the same system after the punch is lifted back, showing the springback of the sheet.

Figure 1.3.3–5 shows the distribution of nominal values of radial and circumferential membrane strain in the sheet for an 18.6 mm punch head displacement. Figure 1.3.3–6 shows the strain distributions at a punch head displacement of 28.5 mm, and Figure 1.3.3–7 shows the strain distributions at a punch head displacement of 34.5 mm. The strain distributions for the SAX1 models compare well with those obtained experimentally by Ghosh and Hecker (1975) and those obtained numerically by Wang and Budiansky (1978), who used a membrane shell finite element formulation. The important phenomenon of necking during stretching is reproduced at nearly the same location, although slightly different strain values are obtained. Draw beads are used to hold the edge of the sheet in the experiment, but in this analysis the sheet is simply clamped at its edge. Incorporation of the draw bead boundary conditions may further improve the correlation with the experimental data.

A spike can be observed in the radial strain distribution toward the edge of the sheet in some of the Abaqus/Standard shell models. This strain spike is the result of the sheet bending around the die. The spike is not present in the membrane element models since they possess no bending stiffness. All of the Abaqus/Standard results presented in Figure 1.3.3–5 through Figure 1.3.3–9 use the node-to-surface contact formulation. Similar results are obtained when using the surface-to-surface contact formulation.

The results obtained with the axisymmetric membrane models are compared with those obtained from the axisymmetric shell models and are found to be in good agreement.

These analyses assume a value of 0.275 for the coefficient of friction. Ghosh and Hecker do not give a value for their experiments, but Wang and Budiansky assume a value of 0.17. The coefficient of friction has a marked effect on the peak strain during necking and may be a factor contributing to the discrepancy of peak strain results during necking. The values used in these analyses have been chosen to provide good correlation with the experimental data.

The distributions of the residual stresses on springback of the sheet are shown in Figure 1.3.3–8 and Figure 1.3.3–9.

Input files

Abaqus/Standard input files

thinsheetstretching_m3d4r.inp	Element type M3D4R.

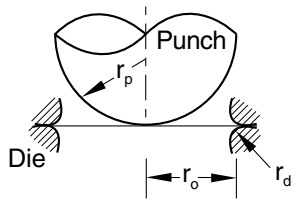
thinsheetstretching_m3d4r_surf.inp	Element type M3D4R using surface-to-surface contact while accounting for shell thickness.
thinsheetstretching_m3d9r.inp	Element type M3D9R.
thinsheetstretching_max1.inp	Element type MAX1.
thinsheetstretching_max2.inp	Element type MAX2.
thinsheetstretching_s4.inp	Element type S4.
thinsheetstretching_s4r.inp	Element type S4R.
thinsheetstretching_s4r_po.inp	*POST OUTPUT analysis.
thinsheetstretching_sc8r.inp	Element type SC8R.
thinsheetstretching_sax1.inp	Element type SAX1.
thinsheetstretching_sax2.inp	Element type SAX2.
thinsheetstretching_restart.inp	Restart of thinsheetstretching_sax2.inp.

Abaqus/Explicit input files

hemipunch_anl.inp	Model using analytical rigid surfaces to describe the rigid surface.
hemipunch.inp	Model using rigid elements to describe the rigid surface.

References

- Ghosh, A. K., and S. S. Hecker, "Failure in Thin Sheets Stretched Over Rigid Punches," Metallurgical Transactions, vol. 6A, pp. 1065–1074, 1975.
- Wang, N. M., and B. Budiansky, "Analysis of Sheet Metal Stamping by a Finite Element Method," Journal of Applied Mechanics, vol. 45, pp. 73–82, 1978.



r_o = 59.18 mm
r_p = 50.8 mm
r_d = 6.35 mm

Figure 1.3.3–1 Configuration and dimensions for hemispherical punch stretching.

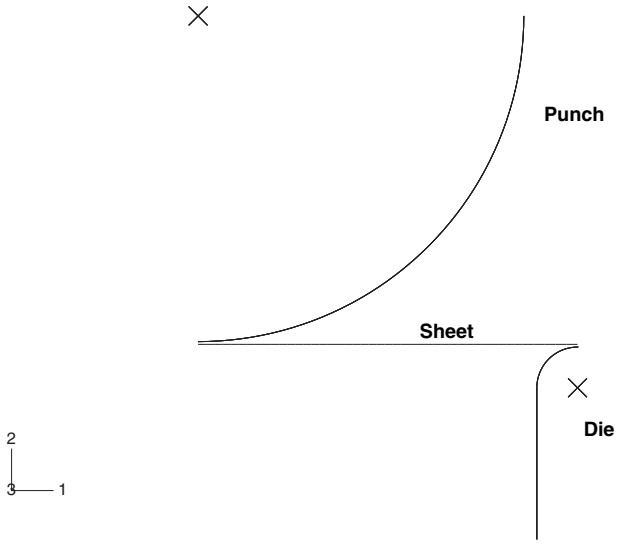


Figure 1.3.3–2 Initial configuration.

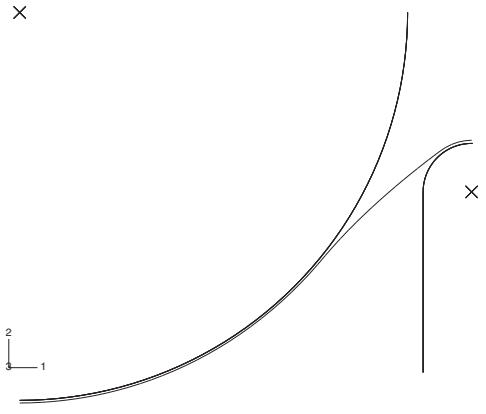


Figure 1.3.3–3 Configuration for punch head displacement of 34.5 mm, Abaqus/Explicit.

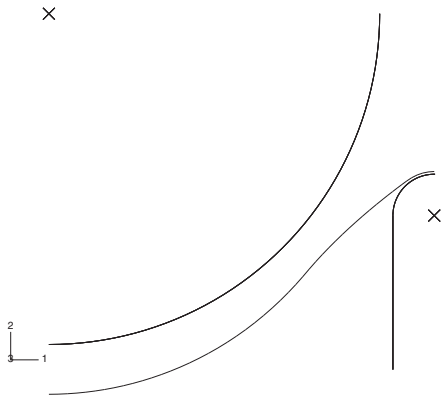


Figure 1.3.3–4 Final configuration after springback, Abaqus/Explicit.

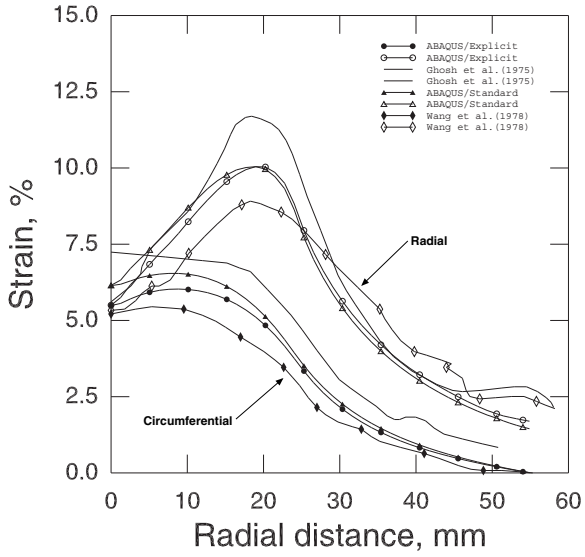


Figure 1.3.3–5 Strain distribution for punch head displacement of 18.6 mm.

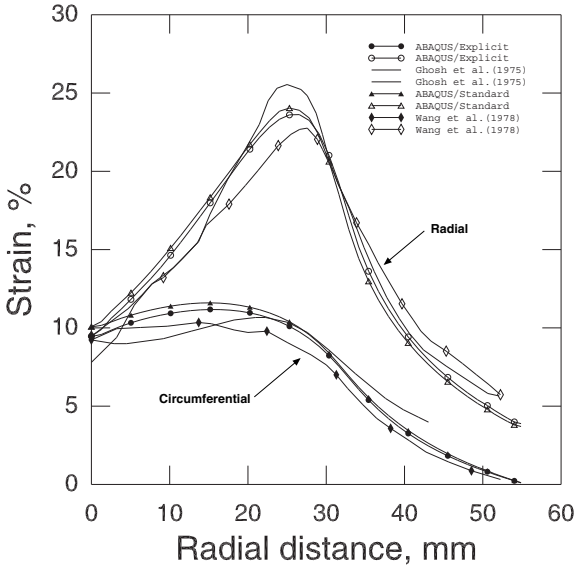


Figure 1.3.3–6 Strain distribution for punch head displacement of 28.5 mm.

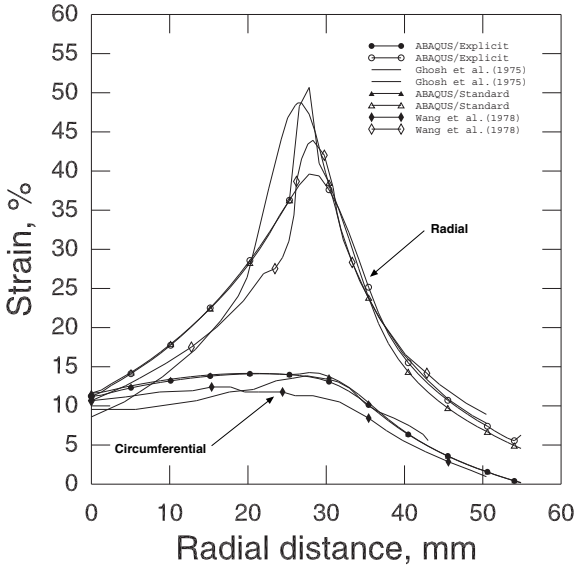


Figure 1.3.3–7 Strain distribution for punch head displacement of 34.5 mm.

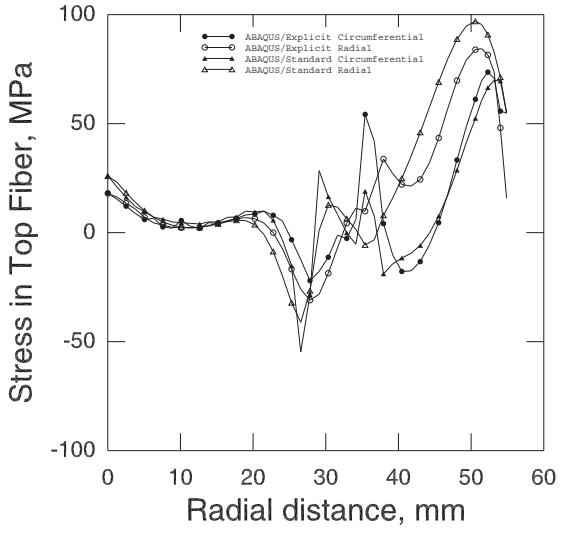


Figure 1.3.3–8 Residual stress on top surface after springback.

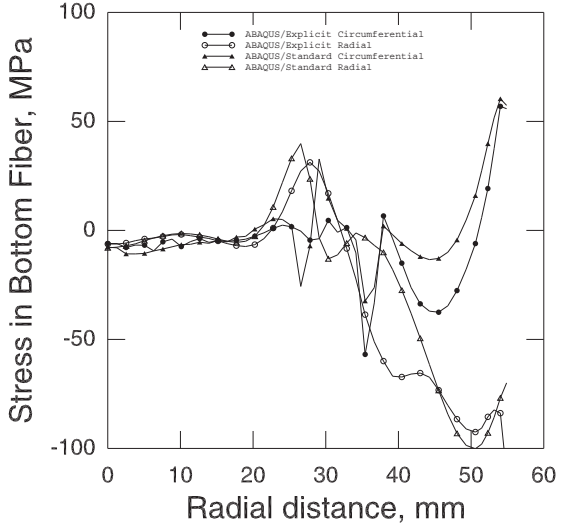


Figure 1.3.3–9 Residual stress on bottom surface after springback.

1.3.4 DEEP DRAWING OF A CYLINDRICAL CUP

Product: Abaqus/Standard

Deep drawing of sheet metal is an important manufacturing technique. In the deep drawing process a "blank" of sheet metal is clamped by a blank holder against a die. A punch is then moved against the blank, which is drawn into the die. Unlike the operation described in the hemispherical punch stretching example ("Stretching of a thin sheet with a hemispherical punch," Section 1.3.3), the blank is not assumed to be fixed between the die and the blank holder; rather, the blank is drawn from between these two tools. The ratio of drawing versus stretching is controlled by the force on the blank holder and the friction conditions at the interface between the blank and the blank holder and the die. Higher force or friction at the blank/die/blank holder interface limits the slip at the interface and increases the radial stretching of the blank. In certain cases drawbeads, shown in Figure 1.3.4–1, are used to restrain the slip at this interface even further.

To obtain a successful deep drawing process, it is essential to control the slip between the blank and its holder and die. If the slip is restrained too much, the material will undergo severe stretching, thus potentially causing necking and rupture. If the blank can slide too easily, the material will be drawn in completely and high compressive circumferential stresses will develop, causing wrinkling in the product. For simple shapes like the cylindrical cup here, a wide range of interface conditions will give satisfactory results. But for more complex, three-dimensional shapes, the interface conditions need to be controlled within a narrow range to obtain a good product.

During the drawing process the response is determined primarily by the membrane behavior of the sheet. For axisymmetric problems in particular, the bending stiffness of the metal yields only a small correction to the pure membrane solution, as discussed by Wang and Tang (1988). In contrast, the interaction between the die, the blank, and the blank holder is critical. Thus, thickness changes in the sheet material must be modeled accurately in a finite element simulation, since they will have a significant influence on the contact and friction stresses at the interface. In these circumstances the most suitable elements in Abaqus are the 4-node reduced-integration axisymmetric quadrilateral, CAX4R; the first-order axisymmetric shell element, SAX1; the first-order axisymmetric membrane element, MAX1; the first-order finite-strain quadrilateral shell element, S4R; the fully integrated general-purpose finite-membrane-strain shell element, S4; and the 8-node continuum shell element, SC8R.

Membrane effects and thickness changes are modeled properly with CAX4R. However, the bending stiffness of the element is low. The element does not exhibit "locking" due to incompressibility or parasitic shear. It is also very cost-effective. For shells and membranes the thickness change is calculated from the assumption of incompressible deformation of the material.

Geometry and model

The geometry of the problem is shown in Figure 1.3.4–2. The circular blank being drawn has an initial radius of 100 mm and an initial thickness of 0.82 mm. The punch has a radius of 50 mm and is rounded off at the corner with a radius of 13 mm. The die has an internal radius of 51.25 mm and is rounded off at the corner with a radius of 5 mm. The blank holder has an internal radius of 56.25 mm.

The blank is modeled using 40 elements of type CAX4R or 31 elements of type SAX1, MAX1, S4R, S4, or SC8R. An 11.25° wedge of the circular blank is used in the three-dimensional S4R and S4 models. These meshes are rather coarse for this analysis. However, since the primary interest in this problem is to study the membrane effects, the analysis will still give a fair indication of the stresses and strains occurring in the process.

The contact between the blank and the rigid punch, the rigid die, and the rigid blank holder is modeled with a contact pair. The top and bottom surfaces of the blank are defined as surfaces in the model. The rigid punch, the die, and the blank holder are modeled as analytical rigid surfaces. The mechanical interaction between the contact surfaces is assumed to be frictional contact. Therefore, friction is used in conjunction with the various contact property definitions to specify coefficients of friction.

At the start of the analysis for the CAX4R model, the blank is positioned precisely on top of the die and the blank holder is precisely in touch with the top surface of the blank. The punch is positioned 0.18 mm above the top surface of the blank.

In the case of shells and membranes, the positioning of the blank depends on the contact formulation used. Node-to-surface and surface-to-surface contact formulations are available in Abaqus/Standard. For the node-to-surface formulation, the shell/membrane thickness is modeled using an exponential pressure-overclosure relationship ("Contact pressure-overclosure relationships," Section 37.1.2 of the Abaqus Analysis User's Guide). The blank holder is positioned a fixed distance above the blank. This fixed distance is the distance at which the contact pressure is set to zero using an exponential pressure-overclosure relationship. However, the surface-to-surface contact formulation automatically takes thickness into account, and the need for specifying pressure overclosure relations is eliminated. Examples of the surface-to-surface contact formulation with S4 and S4R elements are provided in this problem.

Material properties

The material (aluminum-killed steel) is assumed to satisfy the Ramberg-Osgood relation between true stress and logarithmic strain:

$$\epsilon = (\sigma/K)^{1/n}.$$

The reference stress value, K, is 513 MPa; and the work-hardening exponent, n, is 0.223. The Young's modulus is 211 GPa, and the Poisson's ratio is 0.3. An initial yield stress of 91.3 MPa is obtained with these data. The stress-strain curve is defined in piecewise linear segments in the metal plasticity specification, up to a total (logarithmic) strain level of 107%.

The coefficient of friction between the interface and the punch is taken to be 0.25; and that between the die and the blank holder is taken as 0.1, the latter value simulating a certain degree of lubrication between the surfaces. The stiffness method of sticking friction is used in these analyses. The numerics of this method make it necessary to choose an acceptable measure of relative elastic slip between mating surfaces when sticking should actually be occurring. The basis for the choice is as follows. Small values of elastic slip best simulate the actual behavior but also result in a slower convergence of the solution. Permission of large relative elastic displacements between the contacting surfaces can cause higher strains at the center of the blank. In these runs we let Abaqus choose the allowable elastic

slip, which is done by determining a characteristic interface element length over the entire mesh and multiplying by a small fraction to get an allowable elastic slip measure. This method typically gives a fairly small amount of elastic slip.

Although the material in this process is fully isotropic, the local coordinate system is used with the CAX4R elements to define a local orientation that is coincident initially with the global directions. The reason for using this option is to obtain the stress and strain output in more natural coordinates: if the local coordinate system is used in a geometrically nonlinear analysis, stress and strain components are given in a corotational framework. Hence, in our case throughout the motion, S11 will be the stress in the r–z plane in the direction of the middle surface of the cup. S22 will be the stress in the thickness direction, S33 will be the hoop stress, and S12 will be the transverse shear stress, which makes interpreting the results considerably easier. This orientation definition is not necessary with the SAX1 or MAX1 elements since the output for shell and membrane elements is already given in the local shell system. For the SAX1 and MAX1 model, S11 is the stress in the meridional direction and S22 is the circumferential (hoop) stress. An orientation definition would normally be needed for the S4R and S4 models but can be avoided by defining the wedge in such a manner that the single integration point of each element lies along the global x-axis. Such a model definition, along with appropriate kinematic boundary conditions, keeps the local stress output definitions for the shells as S11 being the stress in the meridional plane and S22 the hoop stress. There should be no in-plane shear, S12, in this problem. A transformation is used in the S4R and S4 models to impose boundary constraints in a cylindrical system.

Loading

The entire analysis is carried out in five steps. In the first step the blank holder is pushed onto the blank with a prescribed displacement to establish contact. In the shell models this displacement roughly corresponds to zero clearance across the interface.

In the second step the boundary condition is removed and replaced by the applied force of 100 kN on the blank holder. This force is kept constant during Steps 2 and 3. This technique of simulating the clamping process is used to avoid potential problems with rigid body modes of the blank holder, since there is no firm contact between the blank holder, the blank, and the die at the start of the process. The two-step procedure creates contact before the blank holder is allowed to move freely.

In the third step the punch is moved toward the blank through a total distance of 60 mm. This step models the actual drawing process. During this step the time incrementation parameters are set to improve efficiency for severely discontinuous behavior associated with frictional contact.

The last two steps are used to simulate springback. In the fourth step all the nodes in the model are fixed in their current positions and the contact pairs are removed from the model. This is the most reliable method for releasing contact conditions. In the fifth, and final, step the regular set of boundary conditions is reinstated and the springback is allowed to take place. This part of the analysis with the CAX4R elements is included to demonstrate the feasibility of the unloading procedure only and is not expected to produce realistic results, since the reduced-integration elements have a purely elastic bending behavior. The springback is modeled with more accuracy in the shell element models.

Results and discussion

Figure 1.3.4–3 shows deformed shapes that are predicted at various stages of the drawing process for the CAX4R model. The profiles show that the metal initially bends and stretches and is then drawn in over the surface of the die. The distributions of radial and circumferential strain for all three models and thickness strain for the CAX4R model are shown in Figure 1.3.4–4. The thickness for the shell or membrane models can be monitored with output variable STH (current shell or membrane thickness). The thickness does not change very much: the change ranges from approximately −12% in the cylindrical part to approximately +16% at the edge of the formed cup. Relatively small thickness changes are usually desired in deep drawing processes and are achieved because the radial tensile strain and the circumferential compressive strain balance each other.

The drawing force as a function of punch displacement for various element types is shown in Figure 1.3.4–5, where the curves are seen to match closely. Similarly, the drawing force as a function of punch displacement with S4R elements using the node-to-surface and surface-to-surface contact formulations is shown in Figure 1.3.4–6. The differences in the reaction force history are due to consideration of the blank thickness explicitly in the surface-to-surface contact formulation as compared to the node-to-surface contact formulation where a pressure-overclosure relationship is specified. In all of the cases, oscillations in the force history are seen. These oscillations are a result of the rather coarse mesh—each oscillation represents an element being drawn over the corner of the die. Compared to the shell models, the membrane model predicts a smaller punch force for a given punch displacement. Thus, toward the end of the analysis the results for punch force versus displacement for the MAX1 model are closer to those for the CAX4R model.

The deformed shape after complete unloading is shown in Figure 1.3.4–7, superimposed on the deformed shape under complete loading. The analysis shows the lip of the cup springing back strongly after the blank holder is removed for the CAX4R model. No springback is evident in the shell models. As was noted before, this springback in the CAX4R model is not physically realistic: in the first-order reduced-integration elements an elastic “hourglass control” stiffness is associated with the “bending” mode, since this mode is identical to the “hourglass” mode exhibited by this element in continuum situations. In reality the bending of the element is an elastic-plastic process, so that the springback is likely to be much less. A better simulation of this aspect would be achieved by using several elements through the thickness of the blank, which would also increase the cost of the analysis. The springback results for the shell models do not exhibit this problem and are clearly more representative of the actual elastic-plastic process.

Input files

deepdrawcup_cax4r.inp	CAX4R model.
deepdrawcup_cax4r_surf.inp	CAX4R model using surface-to-surface contact.
deepdrawcup_cax4i.inp	Model using the incompatible mode element, CAX4I, as an alternative to the CAX4R element. In contrast to the reduced-integration, linear isoparametric elements such as the CAX4R element, the incompatible mode elements have excellent bending properties even with one layer of elements through the thickness (see "Geometrically nonlinear analysis of a cantilever beam," Section 2.1.2 of the Abaqus Benchmarks Guide) and have no hourglassing problems. However, they are computationally more expensive.
deepdrawcup_s4.inp	S4 model.
deepdrawcup_s4_surf.inp	S4 model using surface-to-surface contact.
deepdrawcup_s4r.inp	S4R model.
deepdrawcup_s4r_surf.inp	S4R model using surface-to-surface contact.
deepdrawcup_sc8r.inp	SC8R model.
deepdrawcup_sax1.inp	SAX1 model.
deepdrawcup_postoutput.inp	*POST OUTPUT analysis of deepdrawcup_sax1.inp.
deepdrawcup_max1.inp	MAX1 model.
deepdrawcup_mgax1.inp	MGAX1 model.

Reference

- Wang, N. M., and S. C. Tang, "Analysis of Bending Effects in Sheet Forming Operations," International Journal for Numerical Methods in Engineering, vol. 25, pp. 253–267, January 1988.

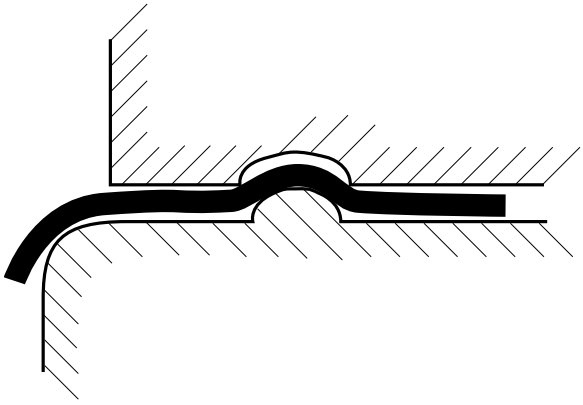


Figure 1.3.4–1 A typical drawbead used to limit slip between the blank and die.

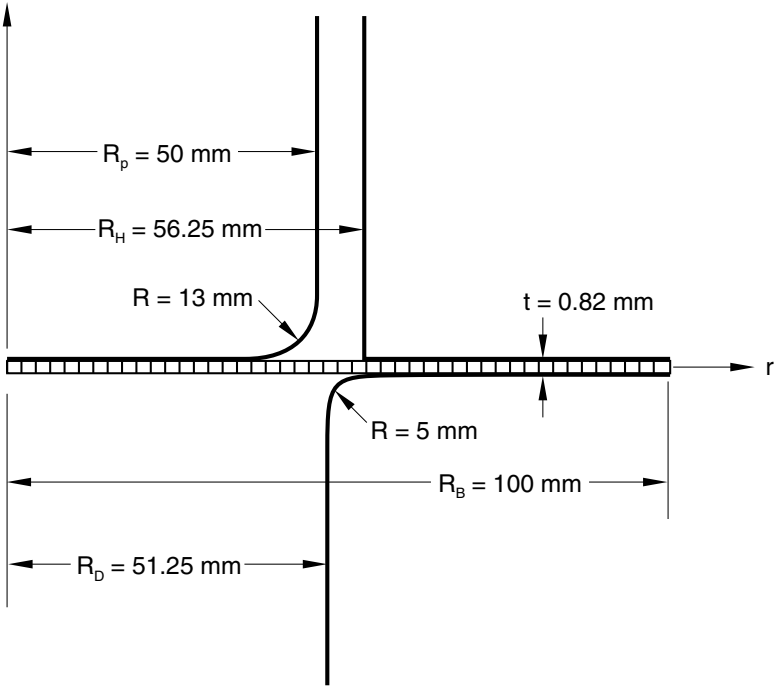


Figure 1.3.4–2 Geometry and mesh for the deep drawing problem.

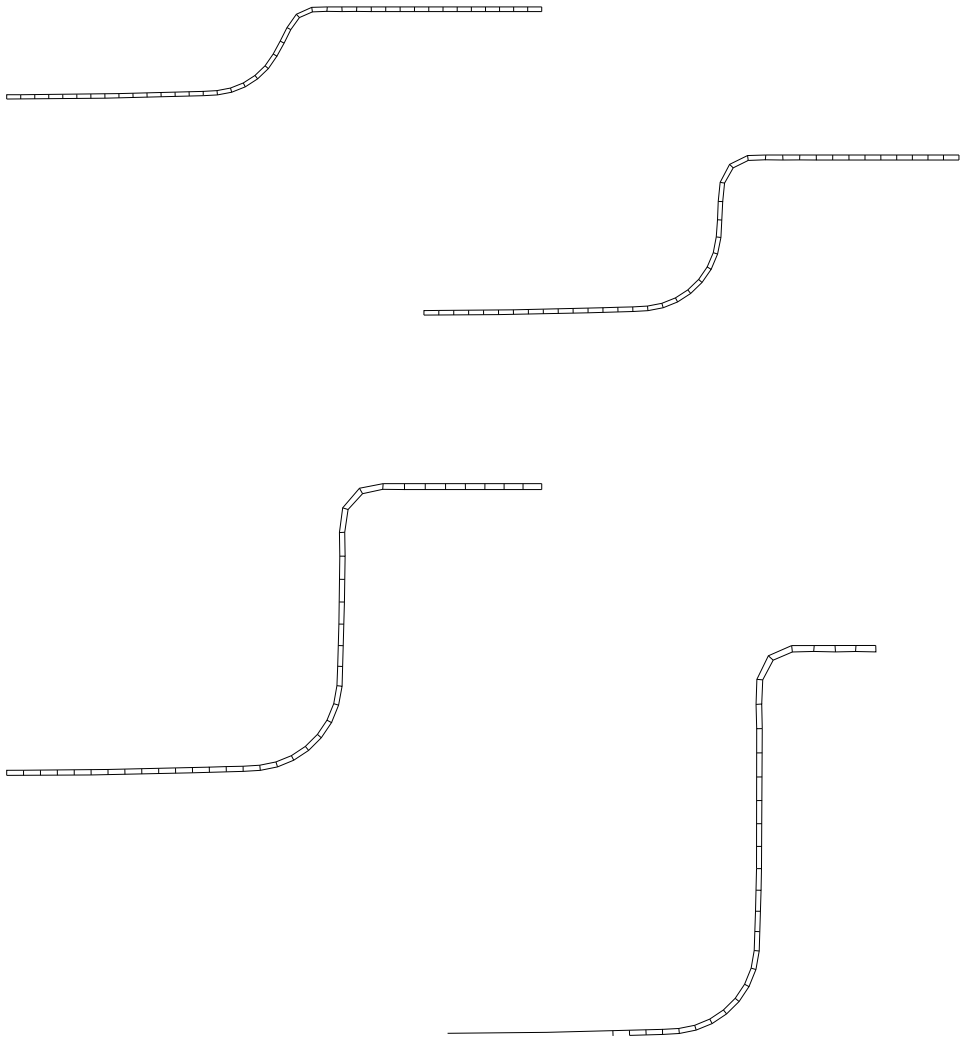


Figure 1.3.4–3 Deformed shapes at various stages of the analysis.

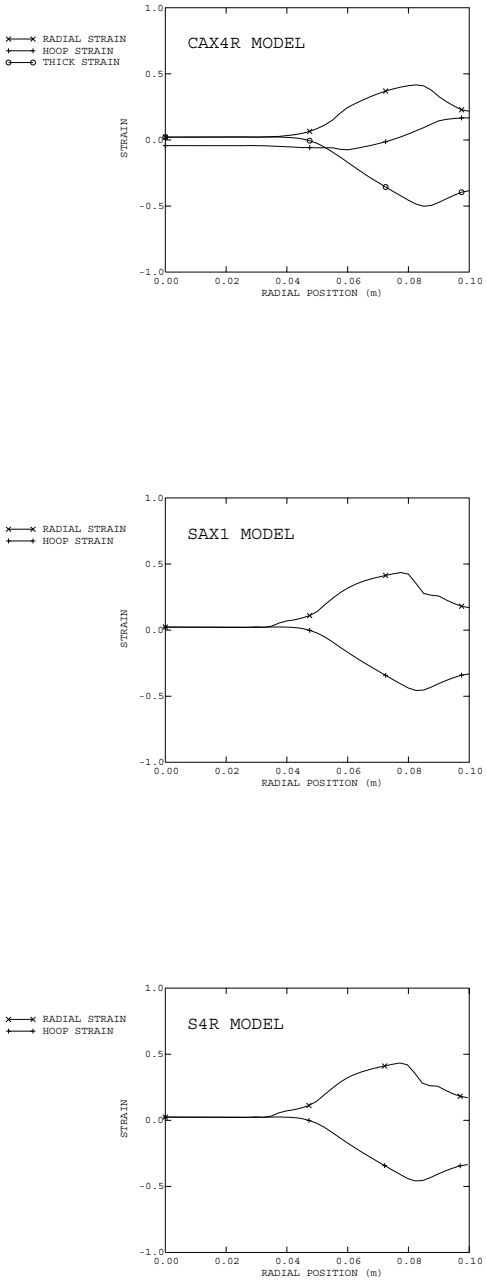


Figure 1.3.4–4 Strain distribution at the end of the deep drawing step.

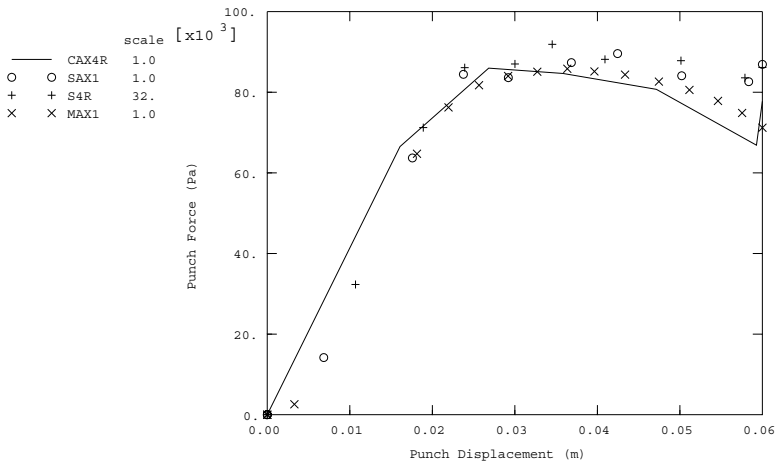


Figure 1.3.4–5 Punch force versus punch displacement using the node-to-surface contact formulation.

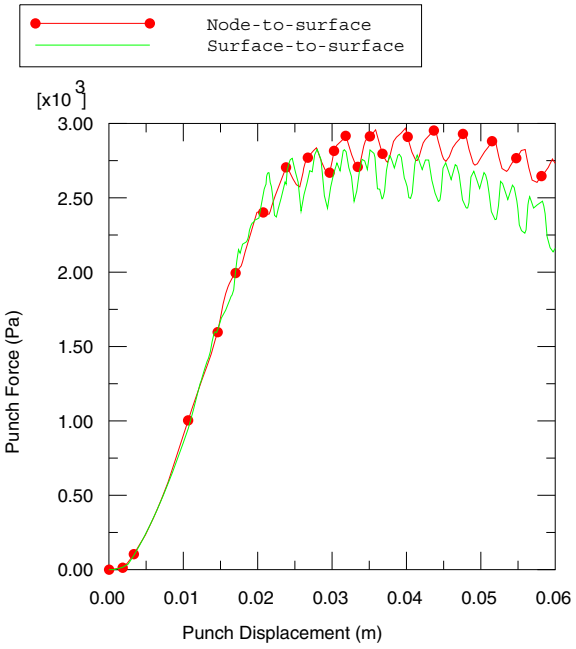


Figure 1.3.4–6 Comparison of punch force versus punch displacement with S4R elements for different contact formulations.

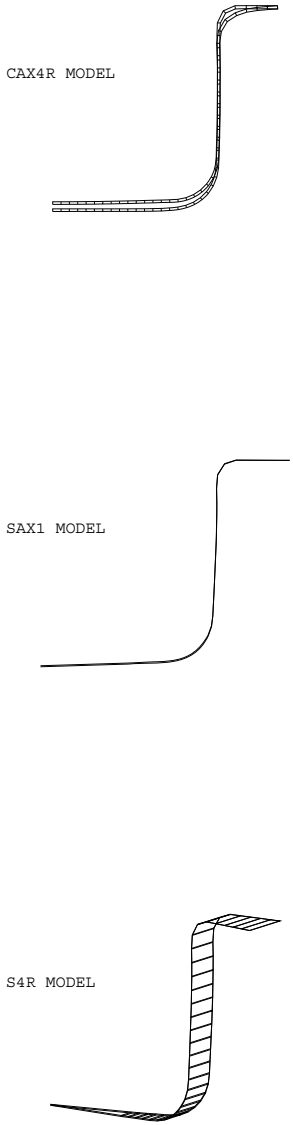


Figure 1.3.4–7 Deformed shape after unloading.

1.3.5 EXTRUSION OF A CYLINDRICAL METAL BAR WITH FRICTIONAL HEAT GENERATION

Products: Abaqus/Standard Abaqus/Explicit

This analysis illustrates how extrusion problems can be simulated with Abaqus. In this particular problem the radius of an aluminum cylindrical bar is reduced 33% by an extrusion process. The generation of heat due to plastic dissipation inside the bar and the frictional heat generation at the workpiece/die interface are considered.

Geometry and model

The bar has an initial radius of 100 mm and is 300 mm long. Figure 1.3.5–1 shows half of the cross-section of the bar, modeled with first-order axisymmetric elements (CAX4T and CAX4RT elements in Abaqus/Standard and CAX4RT elements in Abaqus/Explicit).

In the primary analysis in both Abaqus/Standard and Abaqus/Explicit, heat transfer between the deformable bar and the rigid die is not considered, although frictional heating is included. A fully coupled temperature-displacement analysis is performed with the die kept at a constant temperature. In addition, an adiabatic analysis is presented using Abaqus/Standard without accounting for frictional heat generation. Both the node-to-surface (default) and the surface-to-surface contact formulations in Abaqus/Standard are presented. In the case of Abaqus/Explicit, penalty and kinematic contact formulations are used in the definition of contact interactions.

Various techniques are used to model the rigid die. In Abaqus/Standard the die is modeled with CAX4T elements made into an isothermal rigid body using an isothermal rigid body and with an analytical rigid surface. In Abaqus/Explicit the die is modeled with an analytical rigid surface and discrete rigid elements (RAX2). The fillet radius is set to 0.075 for models using an analytical rigid surface to smoothen the die surface.

The Abaqus/Explicit simulations are also performed with Arbitrary Lagrangian-Eulerian (ALE) adaptive meshing and enhanced hourglass control.

Material model and interface behavior

The material model is chosen to reflect the response of a typical commercial purity aluminum alloy. The material is assumed to harden isotropically. The dependence of the flow stress on the temperature is included, but strain rate dependence is ignored. Instead, representative material data at a strain rate of 0.1 sec^{-1} are selected to characterize the flow strength.

The interface is assumed to have no conductive properties. Coulomb friction is assumed for the mechanical behavior, with a friction coefficient of 0.1. Gap heat generation is used to specify the fraction, f_g, of total heat generated by frictional dissipation that is transferred to the two bodies in contact. Half of this heat is conducted into the workpiece, and the other half is conducted into the die. Furthermore, 90% of the nonrecoverable work due to plasticity is assumed to heat the work material.

Boundary conditions, loading, and solution control

In the first step the bar is moved to a position where contact is established and slipping of the workpiece against the die begins. In the second step the bar is extruded through the die to realize the extrusion process. This is accomplished by prescribing displacements to the nodes at the top of the bar. In the third step the contact elements are removed in preparation for the cool down portion of the simulation. In Abaqus/Standard this is performed in a single step: the bar is allowed to cool down using film conditions, and deformation is driven by thermal contraction during the fourth step.

Volume proportional damping is applied to two of the analyses that are considered. In one case the automatic stabilization scheme with a constant damping factor is used. A nondefault damping density is chosen so that a converged and accurate solution is obtained. In another case the adaptive automatic stabilization scheme with a default damping density is used. In this case the damping factor is automatically adjusted based on the convergence history.

In Abaqus/Explicit the cool down simulation is broken into two steps: the first introduces viscous pressure to damp out dynamic effects and, thus, allow the bar to reach static equilibrium quickly; the balance of the cool down simulation is performed in a fifth step. The relief of residual stresses through creep is not analyzed in this example.

In Abaqus/Explicit mass scaling is used to reduce the computational cost of the analysis; nondefault hourglass control is used to control the hourglassing in the model. The default integral viscoelastic approach to hourglass control generally works best for problems where sudden dynamic loading occurs; a stiffness-based hourglass control is recommended for problems where the response is quasi-static. A combination of stiffness and viscous hourglass control is used in this problem.

For purposes of comparison a second problem is also analyzed, in which the first two steps of the previous analysis are repeated in a static analysis with the adiabatic heat generation capability. The adiabatic analysis neglects heat conduction in the bar. Frictional heat generation must also be ignored in this case. This problem is analyzed only in Abaqus/Standard.

Results and discussion

The following discussion centers around the results obtained with Abaqus/Standard. The results of the Abaqus/Explicit simulation are in close agreement with those obtained with Abaqus/Standard for both the node-to-surface and surface-to-surface contact formulations.

Figure 1.3.5–2 shows the deformed configuration after Step 2 of the analysis. Figure 1.3.5–3 and Figure 1.3.5–4 show contour plots of plastic strain and the Mises stress at the end of Step 2 for the fully coupled analysis using CAX4RT elements. These plots show good agreement between the results using the two contact formulations in Abaqus/Standard. The plastic deformation is most severe near the surface of the workpiece, where plastic strains exceed 100%. The peak stresses occur in the region where the diameter of the workpiece narrows down due to deformation and also along the contact surface. Figure 1.3.5–5 compares nodal temperatures obtained at the end of Step 2 using the surface-to-surface contact formulation in Abaqus/Standard with those obtained using kinematic contact in Abaqus/Explicit. In both cases CAX4RT elements are used. The results from both of the analyses match very well even though mass scaling is used in Abaqus/Explicit for computational savings. The

peak temperature occurs at the surface of the workpiece because of plastic deformation and frictional heating. The peak temperature occurs immediately after the radial reduction zone of the die. This is expected for two reasons. First, the material that is heated by dissipative processes in the reduction zone will cool by conduction as the material progresses through the postreduction zone. Second, frictional heating is largest in the reduction zone because of the larger values of shear stress in that zone.

Similar results were obtained with the two types of stabilization considered. Adaptive automatic stabilization is generally preferred because it is easier to use. It is often necessary to specify a nondefault damping factor for the stabilization approach with a constant damping factor; whereas, with an adaptive damping factor, the default settings are typically appropriate.

Figure 1.3.5–6 compares results of a thermally coupled analysis with an adiabatic analysis using the surface-to-surface contact formulation in Abaqus/Standard. If we ignore the zone of extreme distortion at the end of the bar, the temperature increase on the surface is not as large for the adiabatic analysis because of the absence of frictional heating. As expected, the temperature field contours for the adiabatic heating analysis, shown in Figure 1.3.5–6, are very similar to the contours for plastic strain from the thermally coupled analysis, shown in Figure 1.3.5–3.

As noted earlier, excellent agreement is observed for the results obtained with Abaqus/Explicit (using both the default and enhanced hourglass control) and Abaqus/Standard. Figure 1.3.5–7 compares the effects of ALE adaptive meshing on the element quality. The results obtained with ALE adaptive meshing show significantly reduced mesh distortion. The material point in the bar that experiences the largest temperature rise during the course of the simulation is indicated (node 2029 in the model without adaptivity). Figure 1.3.5–8 compares the results obtained for the temperature history of this material point using Abaqus/Explicit with the results obtained using the two contact formulations in Abaqus/Standard. Again, a very good match between the results is obtained.

Input files

Abaqus/Standard input files

metalbarextrusion_coupled_fric.inp	Thermally coupled extrusion using CAX4T elements with frictional heat generation.
metalbarextrusion_stabil.inp	Thermally coupled extrusion using CAX4T elements with frictional heat generation and automatic stabilization, user-defined damping.
metalbarextrusion_stabil_adap.inp	Thermally coupled extrusion using CAX4T elements with frictional heat generation and adaptive automatic stabilization, default damping.
metalbarextrusion_coupled_fric_surf.inp	Thermally coupled extrusion using CAX4T elements with frictional heat generation and the surface-to-surface contact formulation.
metalbarextrusion_s_coupled_fric_cax4rt.inp	Thermally coupled extrusion using CAX4RT elements with frictional heat generation.

metalbarextrusion_s_coupled_fric_cax4rt_surf.inp
: Thermally coupled extrusion using CAX4RT elements with frictional heat generation and the surface-to-surface contact formulation.

metalbarextrusion_adiab.inp
: Extrusion with adiabatic heat generation and without frictional heat generation.

metalbarextrusion_adiab_surf.inp
: Extrusion with adiabatic heat generation and without frictional heat generation using the surface-to-surface contact formulation.

Abaqus/Explicit input files

metalbarextrusion_x_cax4rt.inp
: Thermally coupled extrusion with frictional heat generation and without ALE adaptive meshing; die modeled with an analytical rigid surface; kinematic mechanical contact.

metalbarextrusion_x_cax4rt_enh.inp
: Thermally coupled extrusion with frictional heat generation and without ALE adaptive meshing; die modeled with an analytical rigid surface; kinematic mechanical contact; enhanced hourglass control.

metalbarextrusion_xad_cax4rt.inp
: Thermally coupled extrusion with frictional heat generation and ALE adaptive meshing; die modeled with an analytical rigid surface; kinematic mechanical contact.

metalbarextrusion_xad_cax4rt_enh.inp
: Thermally coupled extrusion with frictional heat generation and ALE adaptive meshing; die modeled with an analytical rigid surface; kinematic mechanical contact; enhanced hourglass control.

metalbarextrusion_xd_cax4rt.inp
: Thermally coupled extrusion with frictional heat generation and without ALE adaptive meshing; die modeled with RAX2 elements; kinematic mechanical contact.

metalbarextrusion_xd_cax4rt_enh.inp
: Thermally coupled extrusion with frictional heat generation and without ALE adaptive meshing; die modeled with RAX2 elements; kinematic mechanical contact; enhanced hourglass control.

metalbarextrusion_xp_cax4rt.inp
: Thermally coupled extrusion with frictional heat generation and without ALE adaptive meshing; die modeled with an analytical rigid surface; penalty mechanical contact.

metalbarextrusion_xp_cax4rt_enh.inp
: Thermally coupled extrusion with frictional heat generation and without ALE adaptive meshing; die modeled with an analytical rigid surface; penalty mechanical contact; enhanced hourglass control.

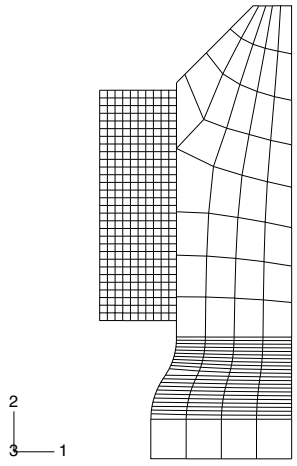


Figure 1.3.5–1 Mesh and geometry: axisymmetric extrusion with meshed rigid die, Abaqus/Standard.

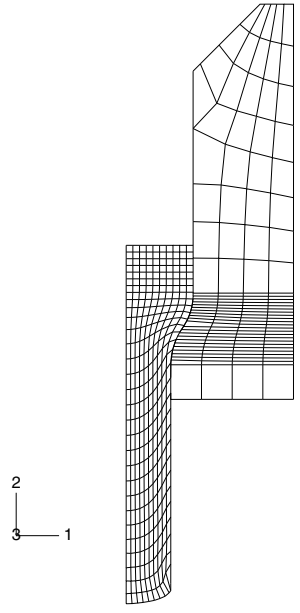


Figure 1.3.5–2 Deformed configuration: Step 2, Abaqus/Standard.

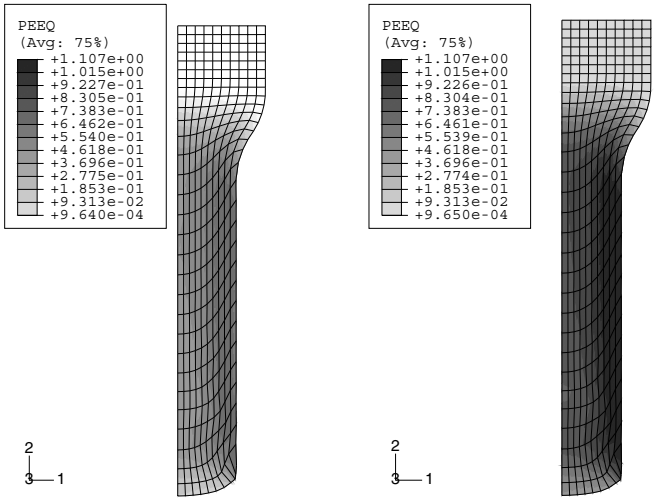


Figure 1.3.5–3 Plastic strain contours: Step 2, thermally coupled analysis (frictional heat generation), Abaqus/Standard (surface-to-surface contact formulation, left; node-to-surface contact formulation, right).

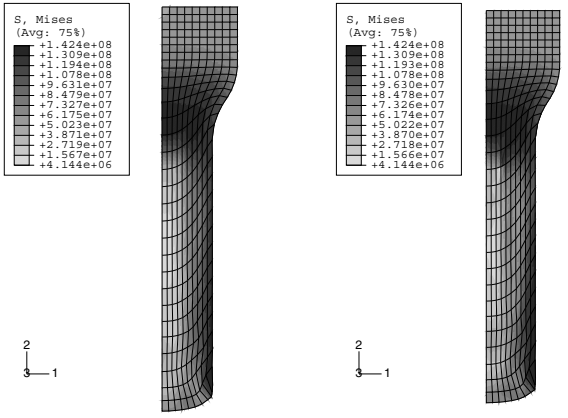


Figure 1.3.5–4 Mises stress contours: Step 2, thermally coupled analysis (frictional heat generation), Abaqus/Standard (surface-to-surface contact formulation, left; node-to-surface contact formulation, right).

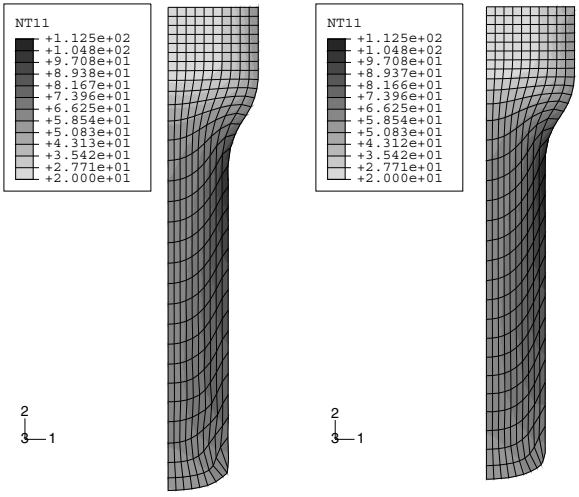


Figure 1.3.5–5 Temperature contours: Step 2, thermally coupled analysis (frictional heat generation); surface-to-surface contact formulation in Abaqus/Standard, left; Abaqus/Explicit, right.

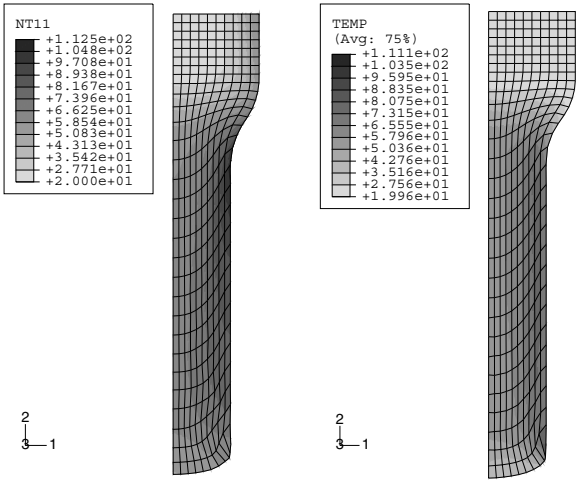


Figure 1.3.5–6 Temperature contours: Step 2, Abaqus/Standard using surface-to-surface contact formulation; thermally coupled analysis, left; adiabatic heat generation (without heat generation due to friction), right.

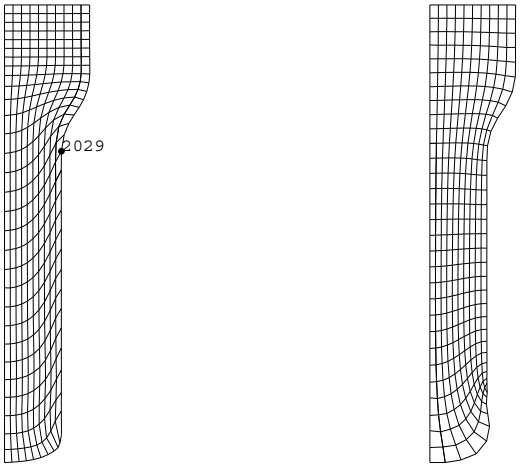


Figure 1.3.5–7 Deformed shape of the workpiece: Abaqus/Explicit; without adaptive remeshing, left; with ALE adaptive remeshing, right.

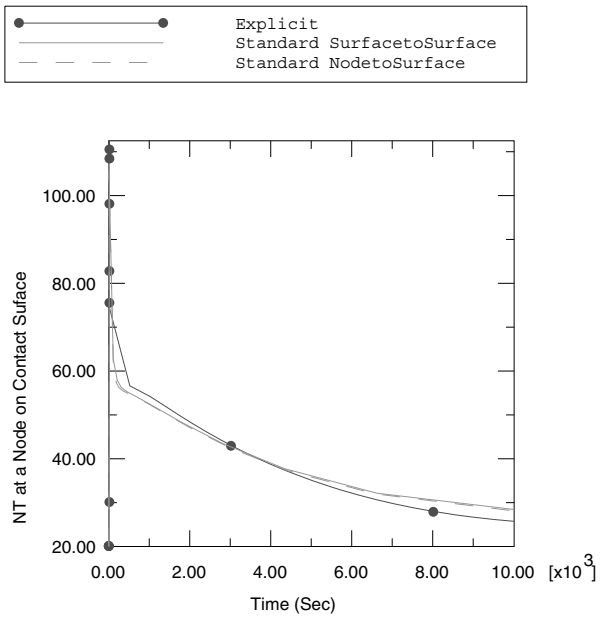


Figure 1.3.5–8 Temperature history of a node on the contact surface (nonadaptive result).

1.3.6 ROLLING OF THICK PLATES

Product: Abaqus/Explicit

Hot rolling is a basic manufacturing technique used to transform preformed shapes into a form suitable for further processing. Rolling processes can be divided into different categories, depending on the complexity of metal flow and on the geometry of the rolled product. Finite element computations are used increasingly to analyze the elongation and spread of the material during rolling (Kobayashi, 1989). Although the forming process is often carried out at low roll speed, this example shows that a considerable amount of engineering information can be obtained by using the explicit dynamics procedure in Abaqus/Explicit to model the process.

The rolling process is first investigated using plane strain computations. These results are used to choose the modeling parameters associated with the more computationally expensive three-dimensional analysis.

Since rolling is normally performed at relatively low speeds, it is natural to assume that static analysis is the proper modeling approach. Typical rolling speeds (surface speed of the roller) are on the order of 1 m/sec. At these speeds inertia effects are not significant, so the response—except for rate effects in the material behavior—is quasi-static. Representative rolling geometries generally require three-dimensional modeling, resulting in very large models, and include nonlinear material behavior and discontinuous effects—contact and friction. Because the problem size is large and the discontinuous effects dominate the solution, the explicit dynamics approach is often less expensive computationally and more reliable than an implicit quasi-static solution technique.

The computer time involved in running a simulation using explicit time integration with a given mesh is directly proportional to the time period of the event. This is because numerical stability considerations restrict the time increment to

$$\Delta t \leq \min\left(L^{el}\sqrt{\frac{\rho}{\lambda+2\mu}}\right),$$

where the minimum is taken over all elements in the mesh, L^{el} is a characteristic length associated with an element, ρ is the density of the material in the element, and λ and μ are the effective Lamé's constants for the material in the element. Since this condition effectively means that the time increment can be no larger than the time required to propagate a stress wave across an element, the computer time involved in running a quasi-static analysis can be very large. The cost of the simulation is directly proportional to the number of time increments required, $n = T/\Delta t$ if Δt remains constant, where T is the time period of the event being simulated. (Δt will not remain constant in general, since element distortion will change L^{el} and nonlinear material response will change the effective Lamé constants and density; but the assumption is acceptable for the purposes of this discussion.) Thus,

$$n = T\max\left(\frac{1}{L^{el}}\sqrt{\frac{\lambda+2\mu}{\rho}}\right).$$

To reduce n, we can speed up the simulation compared to the time of the actual process; that is, we can artificially reduce the time period of the event, T. This will introduce two possible errors. If the simulation

speed is increased too much, the inertia forces will be larger and will change the predicted response (in an extreme case the problem will exhibit wave propagation response). The only way to avoid this error is to find a speedup that is not too large. The other error is that some aspects of the problem other than inertia forces—for example, material behavior—may also be rate dependent. This implies that we cannot change the actual time period of the event being modeled. But we can see a simple equivalent—artificially increasing the material density, ρ, by a factor f^2 reduces $\boldsymbol{n}$ to n/f, just as decreasing $\boldsymbol{T}$ to T/f does. This concept, which is called "mass scaling," reduces the ratio of the event time to the time for wave propagation across an element while leaving the event time fixed, thus allowing treatment of rate-dependent material and other behaviors while having exactly the same effect on inertia forces as speeding up the time of simulation. Mass scaling is attractive because it allows us to treat rate-dependent quasi-static problems efficiently. But we cannot take it too far or we allow the inertia forces to dominate and, thus, change the solution. This example illustrates the use of mass scaling and shows how far we can take it for a practical case.

The trial and error method works well for most generic quasi-static problems; however, for rolling processes Abaqus/Explicit can set the mass scaling factor automatically based on the rolling geometry and mesh properties. An acceptable value for the stable time increment is calculated, and the appropriate mass scaling factor is applied on an element-by-element basis. The value of the stable time increment is based on the average element length in the rolling direction, the average velocity of the product through the rollers, and the number of nodes in the cross-section of the mesh.

Problem description

A steel plate of an original square cross-section of 40 mm by 40 mm and a length of 92 mm is reduced to a 30 mm height by rolling through one roll stand. The radius of the rollers is 170 mm. The single roller in the model (taking advantage of symmetry) is assumed to be rigid and is modeled as an analytical rigid surface. The isotropic hardening yield curve of the steel is taken from Kopp and Dohmen (1990). Isotropic elasticity is assumed, with Young's modulus of 150 GPa and Poisson's ratio of 0.3. The strain hardening is described using 11 points on the yield stress versus plastic strain curve, with an initial yield stress of 168.2 MPa and a maximum yield stress of 448.45 MPa. No rate dependence or temperature dependence is taken into account.

Coulomb friction is assumed between the roller and the plate, with a friction coefficient of 0.3. Friction plays an important role in this process, as it is the only mechanism by which the plate is pulled through the roll stand. If the friction coefficient is too low, the plate cannot be drawn through the roll stand. Initially, when a point on the surface of the plate has just made contact with the roller, the roller surface is moving faster than the point on the surface of the plate and there is a relative slip between the two surfaces. As the point on the plate is drawn into the process zone under the roller, it moves faster and, after a certain distance, sticks to the roller. As the point on the surface of the plate is pushed out of the process zone, it picks up speed and begins to move faster than the roller. This causes slip in the opposite direction before the point on the surface of the sheet finally loses contact with the roller.

For plane strain computations a half-symmetry model with CPE4R elements is used. For the three-dimensional computations a one-quarter symmetry model with C3D8R elements is used. The roller is modeled with analytical rigid surfaces for both the two-dimensional and three-dimensional cases. For quasi-static rolling problems perfectly round analytical surfaces can provide a more accurate

representation of the revolved roller geometry, improve computational efficiency, and reduce noise when compared to element-based rigid surfaces.

The roller is rotated through 32° at a constant angular velocity of 1 revolution per second (6.28 rad/sec), which corresponds to a roller surface speed of 1.07 m/sec. The plate is given an initial velocity in the global x-direction. The initial velocity is chosen to match the x-component of velocity of the roller at the point of first contact. This choice of initial velocity results in a net acceleration of zero in the x-direction at the point of contact and minimizes the initial impact between the plate and the roller. This minimizes the initial transient disturbance.

In all but one of the analyses performed in this example, the masses of all elements in the model are scaled by factors of either 110, 2758, or 68962. These scaling factors translate into effective roller surface speeds of 11.2 m/sec, 56.1 m/sec, and 280.5 m/sec. An alternative, but equivalent, means of mass scaling could be achieved by scaling the material mass density by the aforementioned factors. In one analysis, automatic mass scaling is used.

The element formulation for the two-dimensional (using CPE4R elements) and three-dimensional (using C3D8R elements) analyses uses the pure stiffness form of hourglass control. The element formulation is selected using section controls. In addition, the three-dimensional model (using C3D8R elements) uses the centroidal kinematic formulation. These options are economical yet provide the necessary level of accuracy for this class of problems. Two- and three-dimensional analyses using the default hourglass control formulation, the combined hourglass control formulation, and the enhanced hourglass control formulation are included for comparison. For the three-dimensional case, both the orthogonal kinematic formulation and the centroidal kinematic formulation are considered.

For the sole purpose of testing the performances of the modified triangular and tetrahedral elements, the problem is also analyzed in two dimensions using CPE6M elements and in three dimensions using C3D10M elements.

Results and discussion

Table 1.3.6–1 shows the effective rolling speeds and the relative CPU cost of the cases using the element formulations recommended for this problem. The relative costs are normalized with respect to the CPU time for the two-dimensional model (using CPE4R elements) with the intermediate mass scaling value. In addition, Table 1.3.6–2 compares the relative CPU cost and accuracy between the different element formulations of the solid elements using the intermediate mass scaling value.

Plane strain rolling (CPE4R elements)

A plane strain calculation allows the user to resolve a number of modeling questions in two dimensions before attempting a more expensive three-dimensional calculation. In particular, an acceptable effective mass scaling factor for running the transient dynamics procedure can be determined.

Figure 1.3.6–1 through Figure 1.3.6–3 show contours of equivalent plastic strain for the three mass scaling factors using the stiffness hourglass control. Figure 1.3.6–4 through Figure 1.3.6–6 show contours of shear stress for the same cases. These results show that there is very little difference between the lowest and the intermediate mass scaling cases. All the results are in good agreement with the quasi-static analysis results obtained with Abaqus/Standard. The results of the largest mass

scaling case show pronounced dynamic effects. Table 1.3.6–1 shows the relative run time of the quasi-static calculation, and Table 1.3.6–2 compares the different element formulations at the same level of mass scaling. The intermediate mass scaling case gives essentially the same results as the quasi-static calculation, using about one-thirteenth of the CPU time. In addition to the savings provided by mass scaling, more computational savings are achieved using the stiffness hourglass control element formulation; the results for this formulation compare well to the results for the computationally more expensive element formulations.

Three-dimensional rolling (C3D8R elements)

We have ascertained with the two-dimensional calculations that mass scaling by a factor of 2758 gives results that are essentially the same as a quasi-static solution. Figure 1.3.6–7 shows the distribution of the equivalent plastic strain of the deformed sheet for the three-dimensional case using the centroidal kinematic formulation and stiffness hourglass control. Figure 1.3.6–8 shows the distribution of the equivalent plastic strain of the deformed sheet for the three-dimensional case using the default section controls (average strain kinematic and relax stiffness hourglass). Table 1.3.6–1 compares this three-dimensional case with the plane strain, quasi-static, and three-dimensional automatic mass scaling cases; and Table 1.3.6–2 compares the five different three-dimensional element formulations included here with the two-dimensional cases at the same level of mass scaling. The accuracy for all five element formulations tested is very similar for this problem, but significant savings are realized in the three-dimensional analyses when using more economical element formulations.

Analyses using CPE6M and C3D10M elements

The total number of nodes in the CPE6M model is identical to the number in the CPE4R model. The number of nodes in the C3D10M model is 3440 (compared to 3808 in the C3D8R model). The analyses using the CPE6M and C3D10M elements use a mass scaling factor of 2758. Figure 1.3.6–9 and Figure 1.3.6–10 show the distribution of the equivalent plastic strain of the plate for the two-dimensional and three-dimensional cases, respectively. The results are in reasonably good agreement with other element formulations. However, the CPU costs are higher since the modified triangular and tetrahedral elements use more than one integration point in each element and the stable time increment size is somewhat smaller than in analyses that use reduced-integration elements with the same node count. For the mesh refinements used in this problem, the CPE6M model takes about twice the CPU time as the CPE4R model, while the C3D10M model takes about 5.75 times the CPU time as the C3D8R model.

Input files

roll2d330_anl_ss.inp	Two-dimensional case (using CPE4R elements) with a mass scaling factor of 2758 and the STIFFNESS hourglass control.
roll3d330_rev_anl_css.inp	Three-dimensional case (using C3D8R elements) with a mass scaling factor of 2758, an analytical rigid surface of TYPE=REVOLUTION, and the CENTROID kinematic and STIFFNESS hourglass section control options.

roll2d66_anl_ss.inp	Two-dimensional case (using CPE4R elements) with a mass scaling factor of 110 using the STIFFNESS hourglass control.
roll2d330_anl_cs.inp	Two-dimensional case (using CPE4R elements) with a mass scaling factor of 2758 using the COMBINED hourglass control.
roll2d330_anl_enhs.inp	Two-dimensional case (using CPE4R elements) with a mass scaling factor of 2758 using the ENHANCED hourglass control.
roll2d330_cs.inp	Two-dimensional case (using CPE4R elements) with a mass scaling factor of 2758 using the COMBINED hourglass control and rigid elements.
roll3d330_css.inp	Three-dimensional case (using C3D8R elements) with a mass scaling factor of 2758, rigid elements, and the CENTROID kinematic and STIFFNESS hourglass section control options.
roll3d330_css_gcont.inp	Three-dimensional case (using C3D8R elements) with a mass scaling factor of 2758, rigid elements, the CENTROID kinematic and STIFFNESS hourglass section control options, and the general contact capability.
roll3d330_ocs.inp	Three-dimensional case (using C3D8R elements) with a mass scaling factor of 2758, rigid elements, and the ORTHOGONAL kinematic and COMBINED hourglass section control options.
roll3d330_ocs_gcont.inp	Three-dimensional case (using C3D8R elements) with a mass scaling factor of 2758, rigid elements, the ORTHOGONAL kinematic and COMBINED hourglass section control options, and the general contact capability.
roll2d1650_anl_ss.inp	Two-dimensional case (using CPE4R elements) with a mass scaling factor of 68962 using STIFFNESS hourglass control.
roll3d330_rev_anl_ocs.inp	Three-dimensional model (using C3D8R elements) with a mass scaling factor of 2758, an analytical rigid surface of TYPE=REVOLUTION, and the ORTHOGONAL kinematic and COMBINED hourglass section control options.
roll3d330_rev_anl_oenhs.inp	Three-dimensional model (using C3D8R elements) with a mass scaling factor of 2758, an analytical rigid surface of TYPE=REVOLUTION, and the ORTHOGONAL kinematic and ENHANCED hourglass section control options.

roll3d330_rev_anl_cenhs.inp	Three-dimensional model (using C3D8R elements) with a mass scaling factor of 2758, an analytical rigid surface of TYPE=REVOLUTION, and the CENTROID kinematic and ENHANCED hourglass section control options.
roll3d330_rev_anl.inp	Three-dimensional model (using C3D8R elements) with a mass scaling factor of 2758, an analytical rigid surface of TYPE=REVOLUTION, and the default section control options.
roll3d_auto_rev_anl_css.inp	Three-dimensional case (using C3D8R elements) with automatic mass scaling, an analytical rigid surface of TYPE=REVOLUTION, and the CENTROID kinematic and STIFFNESS hourglass section control options.
roll3d330_cyl_anl.inp	Three-dimensional model (using C3D8R elements) with a mass scaling factor of 2758, an analytical rigid surface of TYPE=CYLINDER, and the default section control options.
roll2d66.inp	Two-dimensional model (using CPE4R elements) with a mass scaling factor of 110 and the default section controls.
roll2d330.inp	Two-dimensional model (using CPE4R elements) with a mass scaling factor of 2758 and the default section controls.
roll2d1650.inp	Two-dimensional model (using CPE4R elements) with a mass scaling factor of 68962 and the default section controls.
roll3d330.inp	Three-dimensional model using rigid elements and the default section controls.
roll3d330_gcont.inp	Three-dimensional model using rigid elements, the default section controls, and the general contact capability.
roll2d66_anl.inp	Two-dimensional model (using CPE4R elements) with a mass scaling factor of 110, analytical rigid surfaces, and the default section controls.
roll2d330_anl.inp	Two-dimensional model (using CPE4R elements) with a mass scaling factor of 2758, analytical rigid surfaces, and the default section controls.
roll2d1650_anl.inp	Two-dimensional model (using CPE4R elements) with a mass scaling factor of 68962, analytical rigid surfaces, and the default section controls.
roll2d_impl_qs.inp	Implicit, quasi-static, two-dimensional model (using CPE4R elements) with analytical rigid surfaces.
roll2d330_anl_cpe6m.inp	Two-dimensional case (using CPE6M elements) with a mass scaling factor of 2758.

roll3d330_anl_c3d10m.inp	Three-dimensional case (using C3D10M elements) with a mass scaling factor of 2758.
roll3d_medium.inp	Additional mesh refinement case (using C3D8R elements) included for the sole purpose of testing the performance of the code.
roll3d_medium_gcont.inp	Additional mesh refinement case (using C3D8R elements) with the general contact capability.

References

- Kobayashi, S., S. I. Oh, and T. Altan, *Metal Forming and the Finite Element Method,* Oxford University Press, 1989.
- Kopp, R., and P. M. Dohmen, “Simulation und Planung von Walzprozessen mit Hilfe der Finite-Elemente-Methode (FEM),” Stahl U. Eisen, no. 7, pp. 131–136, 1990.

Table 1.3.6–1 Analysis cases and relative CPU costs. (The two-dimensional explicit analyses all use CPE4R elements and stiffness hourglass control. The three-dimensional explicit analyses use C3D8R elements and the centroidal kinematic and stiffness hourglass section controls.)

Analysis Type	Mass Scaling Factor	Effective Roll Surface Speed (m/sec)	Relative CPU Time
Explicit, plane strain	110.3	11.2	4.99
Explicit, plane strain	2758.5	56.1	1.00
Explicit, plane strain	68961.8	280.5	0.21
Implicit, plane strain		quasi-static	13.4
Explicit, 3D	2758.5	56.1	13.8
Explicit, 3D	automatic	~96	9.5

Table 1.3.6–2 Explicit section controls tested (mass scaling factor=2758.5). CPE4R and C3D8R elements are employed for the two-dimensional and three-dimensional cases, respectively. Spread values are reported for the half-model at node 24015.

Analysis Type	Section Controls		Relative CPU Time	Spread (mm)
	Kinematic	Hourglass		
Explicit, plane strain	n/a	stiffness	1.00	n/a
Explicit, plane strain	n/a	relax stiffness	1.11	n/a
Explicit, plane strain	n/a	combined	1.04	n/a
Explicit, plane strain	n/a	enhanced	1.02	n/a
Explicit, 3D	average strain	relax stiffness	20.8	2.06
Explicit, 3D	orthogonal	combined	17.1	2.07
Explicit, 3D	centroidal	stiffness	13.8	2.10
Explicit, 3D	centroidal	enhanced	14.8	2.10
Explicit, 3D	orthogonal	enhanced	17.3	2.10

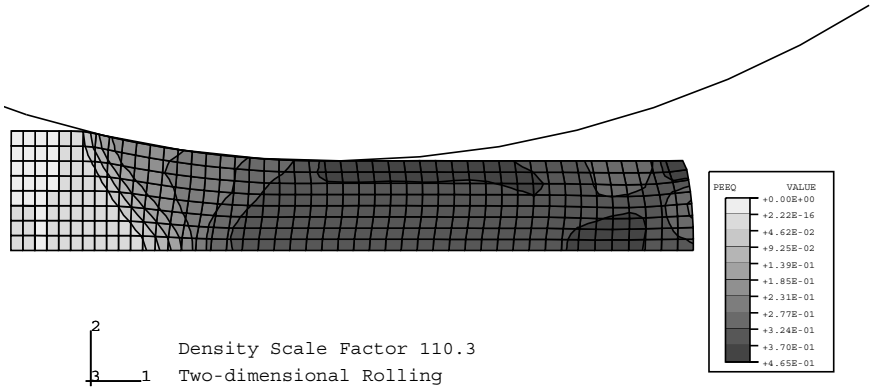


Figure 1.3.6–1 Equivalent plastic strain for the plane strain case (CPE4R) with stiffness hourglass control (mass scaling factor=110.3).

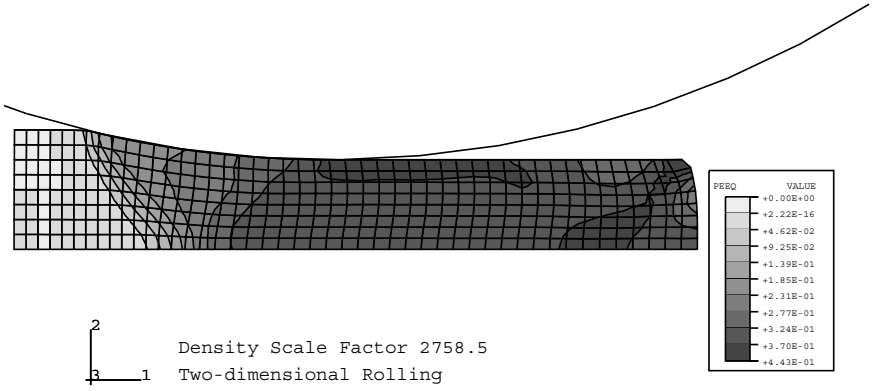


Figure 1.3.6–2 Equivalent plastic strain for the plane strain case (CPE4R) with stiffness hourglass control (mass scaling factor=2758.5).

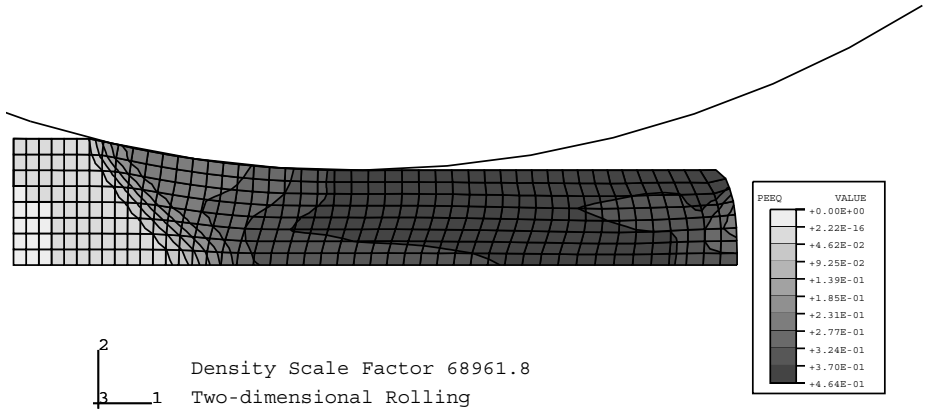


Figure 1.3.6–3 Equivalent plastic strain for the plane strain case (CPE4R) with stiffness hourglass control (mass scaling factor=68961.8).

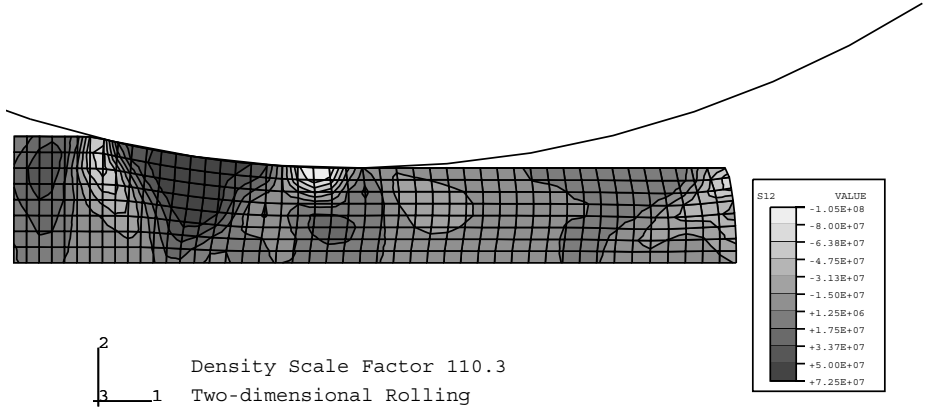


Figure 1.3.6–4 Shear stress for the plane strain case (CPE4R) with stiffness hourglass control (mass scaling factor=110.3).

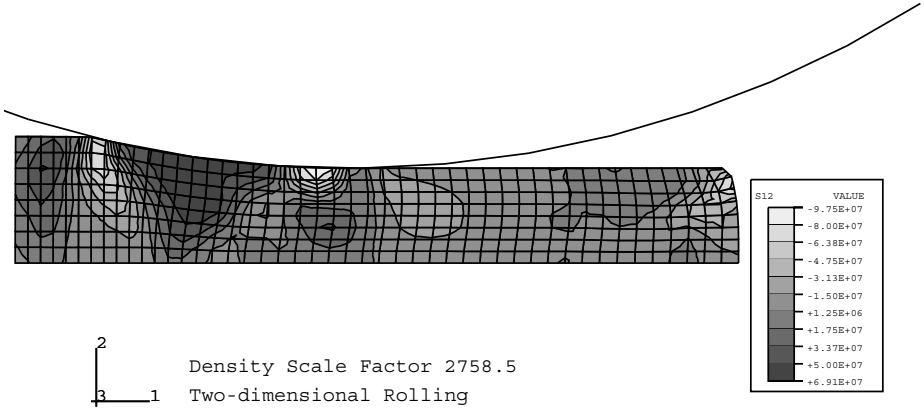


Figure 1.3.6–5 Shear stress for the plane strain case (CPE4R) with stiffness hourglass control (mass scaling factor=2758.5).

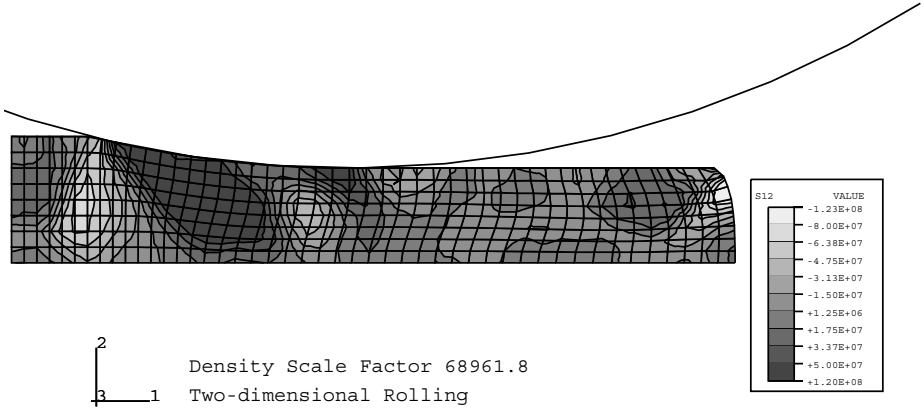


Figure 1.3.6–6 Shear stress for the plane strain case (CPE4R) with stiffness hourglass control (mass scaling factor=68961.8).

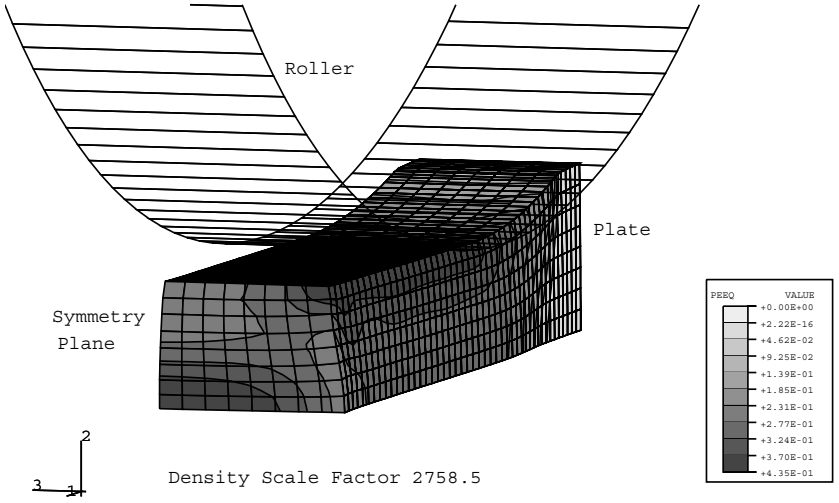


Figure 1.3.6–7 Equivalent plastic strain for the three-dimensional case (C3D8R) using the centroidal kinematic and stiffness hourglass section controls (mass scaling factor=2758.5).

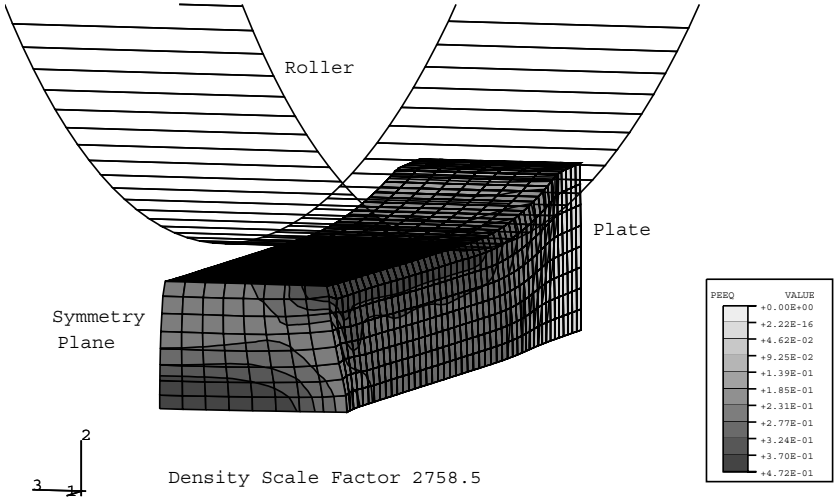


Figure 1.3.6–8 Equivalent plastic strain for the three-dimensional case (C3D8R) using the average strain kinematic and relax stiffness hourglass section controls (mass scaling factor=2758.5).

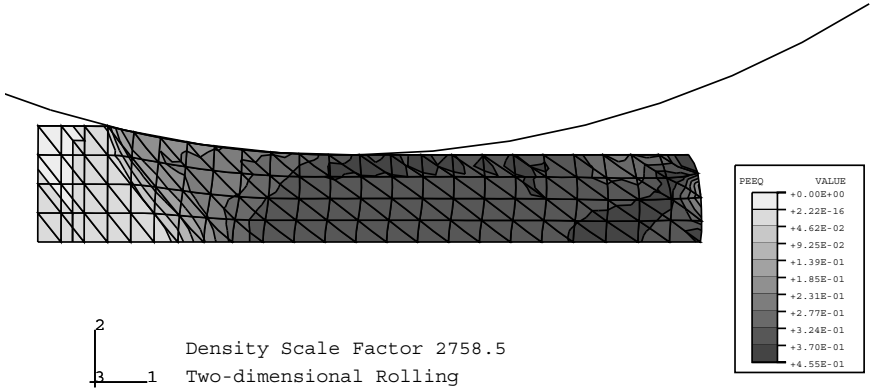


Figure 1.3.6–9 Equivalent plastic strain for the plane strain case (CPE6M) (mass scaling factor=2758.5).

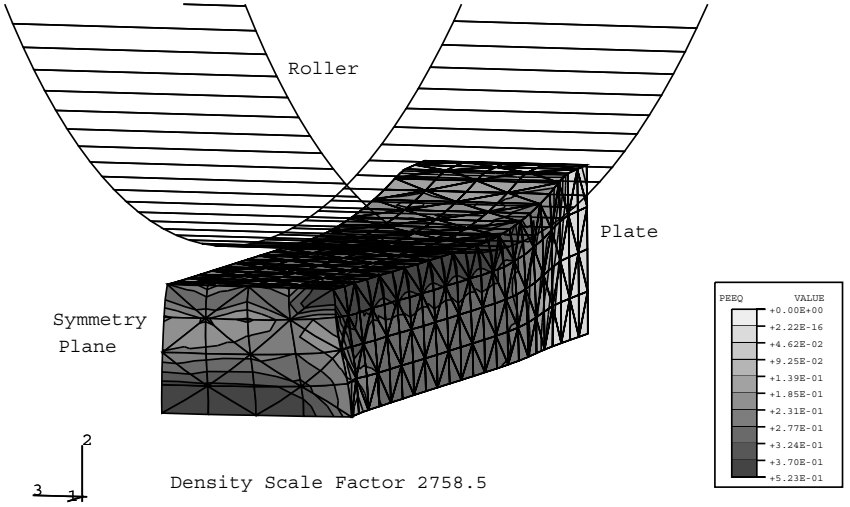


Figure 1.3.6–10 Equivalent plastic strain for the three-dimensional case (C3D10M) (mass scaling factor=2758.5).

1.3.7 AXISYMMETRIC FORMING OF A CIRCULAR CUP

Products: Abaqus/Standard Abaqus/Explicit

This example illustrates the hydroforming of a circular cup using an axisymmetric model. In this case a two-stage forming sequence is used, with annealing between the stages. Two analysis methods are used: in one the entire process is analyzed using Abaqus/Explicit; in the other the forming sequences are analyzed with Abaqus/Explicit, while the springback analyses are run in Abaqus/Standard. Here, the import capability is used to transfer results between Abaqus/Explicit and Abaqus/Standard and vice versa.

Problem description

The model consists of a deformable blank and three rigid dies. The blank has a radius of 150.0 mm, is 1.0 mm thick, and is modeled using axisymmetric shell elements, SAX1. The coefficient of friction between the blank and the dies is taken to be 0.1. Dies 1 and 2 are offset from the blank by half of the thickness of the blank, because the contact algorithm takes into account the shell thickness. To avoid pinching of the blank while die 3 is put into position for the second forming stage, the radial gap between dies 2 and 3 is set to be 20% bigger than the initial shell thickness. Figure 1.3.7–1 and Figure 1.3.7–2 show the initial geometry of the model.

The three dies are modeled with either two-dimensional analytical rigid surfaces or RAX2 rigid elements. An analytical rigid surface can yield a more accurate representation of two-dimensional curved punch geometries and result in computational savings. Contact pressure can be viewed on the specimen surface, and the reaction force is available at the rigid body reference node. In addition, both the kinematic (default) and penalty contact formulations are tested. Results for the kinematic contact formulation using rigid elements are presented here.

The blank is made of aluminum-killed steel, which is assumed to satisfy the Ramberg-Osgood relation between true stress and logarithmic strain,

$$\epsilon = (\sigma/K)^{1/n},$$

with a reference stress value (K) of 513 MPa and work-hardening exponent (n) of 0.223. Isotropic elasticity is assumed, with Young's modulus of 211 GPa and Poisson's ratio of 0.3. With these data an initial yield stress of 91.3 MPa is obtained. The stress-strain behavior is defined by piecewise linear segments matching the Ramberg-Osgood curve up to a total (logarithmic) strain level of 107%, with Mises yield, isotropic hardening, and no rate dependence.

The analysis that is performed entirely within Abaqus/Explicit consists of six steps. In the first step contact is defined between the blank and dies 1 and 2. Both dies remain fixed while a distributed load of 10 MPa in the negative z-direction is ramped onto the blank. This load is then ramped off in the second step, allowing the blank to spring back to an equilibrium state.

The third step is an annealing step. The annealing procedure in Abaqus/Explicit sets all appropriate state variables to zero. These variables include stresses, strains (excluding the thinning strain for shells,

membranes, and plane stress elements), plastic strains, and velocities. There is no time associated with an annealing step. The process occurs instantaneously.

In the fourth step contact is defined between the blank and die 3 and contact is removed between the blank and die 1. Die 3 moves down vertically in preparation for the next pressure loading.

In the fifth step another distributed load is applied to the blank in the positive z-direction, forcing the blank into die 3. This load is then ramped off in the sixth step to monitor the springback of the blank.

To obtain a quasi-static response, an investigation was conducted to determine the optimum rate for applying the pressure loads and removing them. The optimum rate balances the computational time against the accuracy of the results; increasing the loading rate will reduce the computer time but lead to less accurate quasi-static results.

The analysis that uses the import capability consists of four runs. The first run is identical to Step 1 of the Abaqus/Explicit analysis described earlier. In the second run the Abaqus/Explicit results for the first forming stage are imported into Abaqus/Standard without updating the reference configuration and with an import of the material state for the first springback analysis. The third run imports the results of the first springback analysis into Abaqus/Explicit for the subsequent annealing process and the second forming stage. By updating the reference configuration and not importing the material state, this run begins with no initial stresses or strains, effectively simulating the annealing process. The final run imports the results of the second forming stage into Abaqus/Standard for the second springback analysis.

Results and discussion

Figure 1.3.7–3 to Figure 1.3.7–5 show the results of the analysis conducted entirely within Abaqus/Explicit using the rigid element approach and the kinematic contact formulation. Figure 1.3.7–3 shows the deformed shape at the end of Step 2, after the elastic springback. Figure 1.3.7–4 shows the deformed shape at the end of the analysis, after the second elastic springback. Although it is not shown here, the amount of springback observed during the unloading steps is negligible. Figure 1.3.7–5 shows a contour plot of the shell thickness (STH) at the end of the analysis. The thickness of the material at the center of the cup has been reduced by about 20%, while the thickness at the edges of the cup has been increased by about 10%.

The results obtained using the import capability to perform the springback analyses in Abaqus/Standard are nearly identical, as are those obtained using analytical rigid surfaces and/or penalty contact formulations.

You can use the **abaqus restartjoin** execution procedure to extract data from the output database created by a restart analysis and append the data to a second output database. For more information, see "Joining output database (`.odb`) files from restarted analyses," Section 3.2.21 of the Abaqus Analysis User's Guide.

Input files

axiform.inp	Abaqus/Explicit analysis that uses rigid elements and kinematic contact. This file is also used for the first step of the analysis that uses the import capability.

axiform_anl.inp	Model using analytical rigid surfaces and kinematic contact.
axiform_pen.inp	Model using rigid elements and penalty contact.
axiform_anl_pen.inp	Model using analytical rigid surfaces and penalty contact.
axiform_sprbk1.inp	First springback analysis using the import capability.
axiform_form2.inp	Second forming analysis using the import capability.
axiform_sprbk2.inp	Second springback analysis using the import capability.
axiform_restart.inp	Restart of axiform.inp included for the purpose of testing the restart capability.
axiform_rest_anl.inp	Restart of axiform_anl.inp included for the purpose of testing the restart capability.

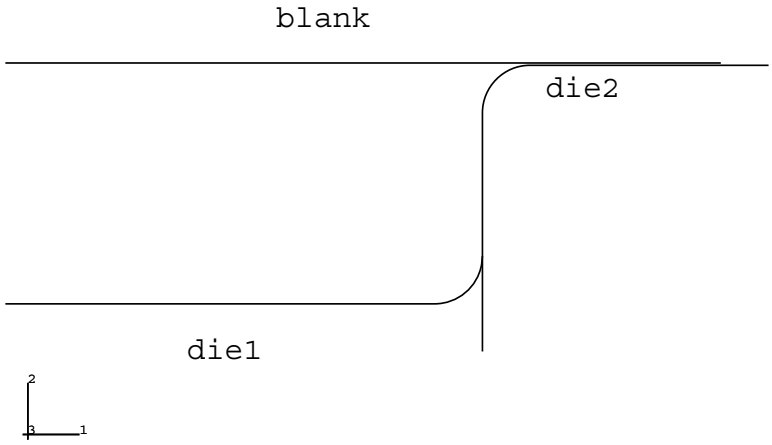


Figure 1.3.7–1 Configuration at the beginning of stage 1.

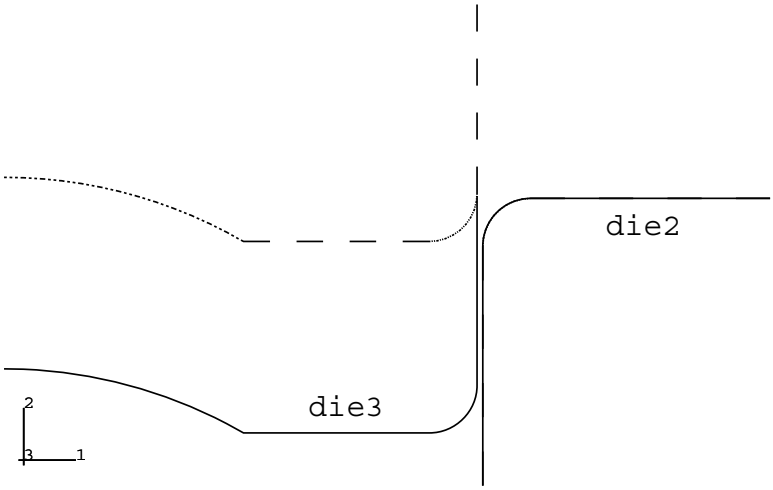


Figure 1.3.7–2 Configuration of dies in forming stage 2. (The dotted line shows the initial position of die 3.)



Figure 1.3.7–3 Deformed configuration after the first forming stage.

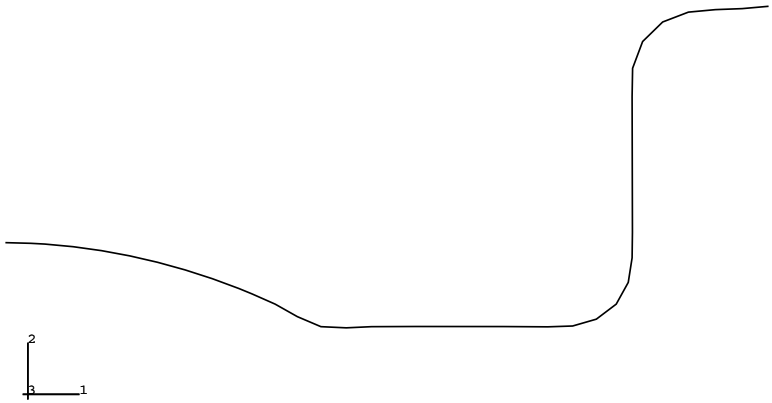


Figure 1.3.7–4 Final configuration.

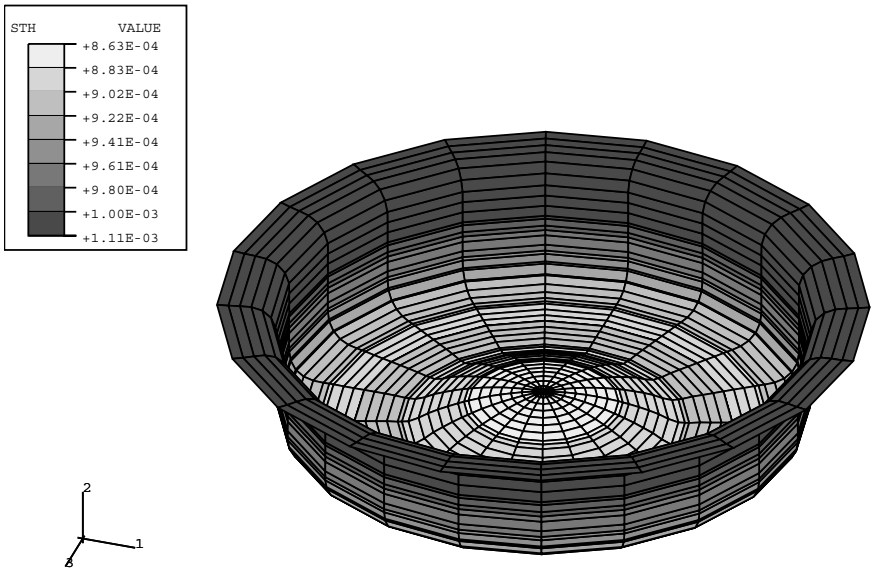


Figure 1.3.7–5 Contour plot of shell thickness.

1.3.8 CUP/TROUGH FORMING

Product: Abaqus/Explicit

This example illustrates the use of adaptive meshing in forging problems that include large amounts of shearing at the tool-blank interface; a cup and a trough are formed.

Problem description

Three different geometric models are considered, as shown in Figure 1.3.8–1. Each model consists of a rigid punch, a rigid die, and a deformable blank. The outer top and bottom edges of the blank are cambered, which facilitates the flow of material against the tools. The punch and die have semicircular cross-sections; the punch has a radius of 68.4 mm, and the die has a radius of 67.9 mm. The blank is modeled as a von Mises elastic, perfectly plastic material with a Young's modulus of 4000 MPa and a yield stress of 5 MPa. The Poisson's ratio is 0.21; the density is 1.E−4 tonne/mm^3.

In each case the punch is moved 61 mm, while the die is fully constrained. A smooth amplitude curve is used to ramp the punch velocity to a maximum, at which it remains constant. The smoothing of the velocity promotes a quasi-static response to the loading.

Case 1: Axisymmetric model for cup forming

The blank is meshed with CAX4R elements and measures 50 × 64.77 mm. The punch and the die are modeled as analytical rigid surfaces using connected line segments. Symmetry boundary conditions are prescribed at r=0. The finite element model is shown in Figure 1.3.8–2.

Case 2: Three-dimensional model for trough forming

The blank is meshed with C3D8R elements and measures 50 × 64.7 × 64.7 mm. The punch and the die are modeled as three-dimensional cylindrical analytical rigid surfaces. Symmetry boundary conditions are applied at the x=0 and z=0 planes. The finite element model of the blank is shown in Figure 1.3.8–3.

Case 3: Three-dimensional model for cup forming

The blank is meshed with C3D8R elements. A 90° wedge of the blank with a radius of 50 mm and a height of 64.7 mm is analyzed. The punch and the die are modeled as three-dimensional revolved analytical rigid surfaces. Symmetry boundary conditions are applied at the x=0 and y=0 planes. The finite element model of the blank is shown in Figure 1.3.8–4.

Adaptive meshing

A single adaptive mesh domain that incorporates the entire blank is used for each model. Symmetry planes are defined as Lagrangian boundary regions (the default), and contact surfaces are defined as sliding boundary regions (the default). Since this problem is quasi-static with relatively small amounts of deformation per increment, the default values for frequency, mesh sweeps, and other adaptive mesh parameters and controls are sufficient.

Results and discussion

Figure 1.3.8–5 through Figure 1.3.8–7 show the mesh configuration at the end of the forging simulation for Cases 1–3. In each case a quality mesh is maintained throughout the simulation. As the blank flattens out, geometric edges and corners that exist at the beginning of the analysis are broken and adaptive meshing is allowed across them. The eventual breaking of geometric edges and corners is essential for this type of problem to minimize element distortion and optimize element aspect ratios.

For comparison purposes Figure 1.3.8–8 shows the deformed mesh for a pure Lagrangian simulation of Case 1 (the axisymmetric model). The mesh is clearly better when continuous adaptive meshing is used. Several diamond-shaped elements with extremely poor aspect ratios are formed in the pure Lagrangian simulation. Adaptive meshing improves the element quality significantly, especially along the top surface of the cup where solution gradients are highest. Figure 1.3.8–9 and Figure 1.3.8–10 show contours of equivalent plastic strain at the completion of the forging for the adaptive meshing and pure Lagrangian analyses of Case 1, respectively. Overall plastic strains compare quite closely. Slight differences exist only along the upper surface, where the pure Lagrangian mesh becomes very distorted at the end of the simulation. The time histories of the vertical punch force for the adaptive and pure Lagrangian analyses agree closely for the duration of the forging, as shown in Figure 1.3.8–11.

Input files

ale_cupforming_axi.inp	Case 1.
ale_cupforming_axinodes.inp	External file referenced by Case 1.
ale_cupforming_axielements.inp	External file referenced by Case 1.
ale_cupforming_cyl.inp	Case 2.
ale_cupforming_sph.inp	Case 3.
lag_cupforming_axi.inp	Lagrangian solution of Case 1.

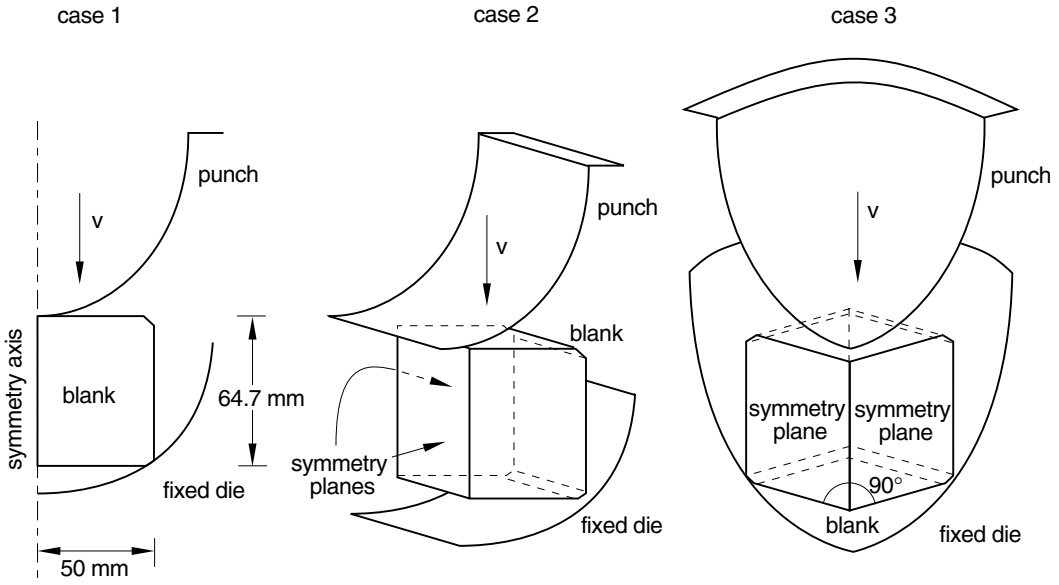


Figure 1.3.8–1 Model geometries for each case.

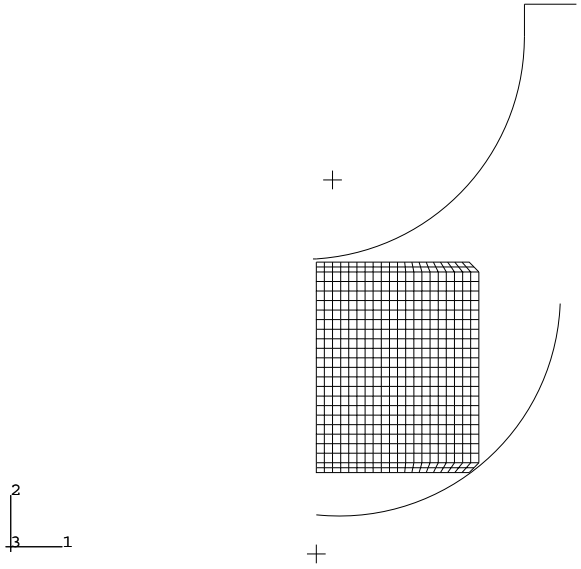


Figure 1.3.8–2 Undeformed mesh for Case 1.

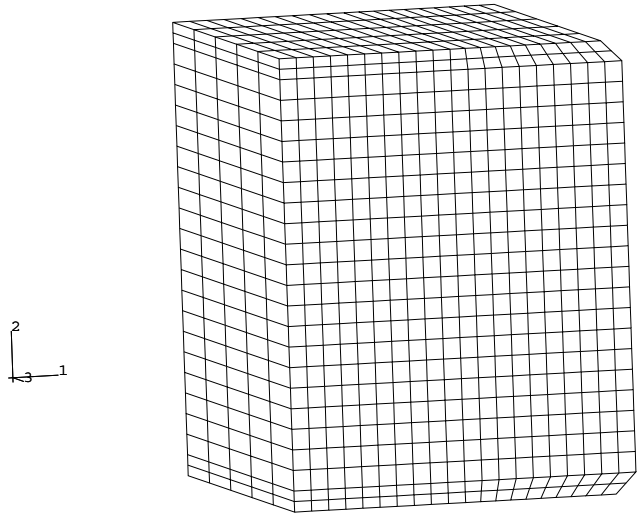


Figure 1.3.8–3 Undeformed mesh for Case 2.

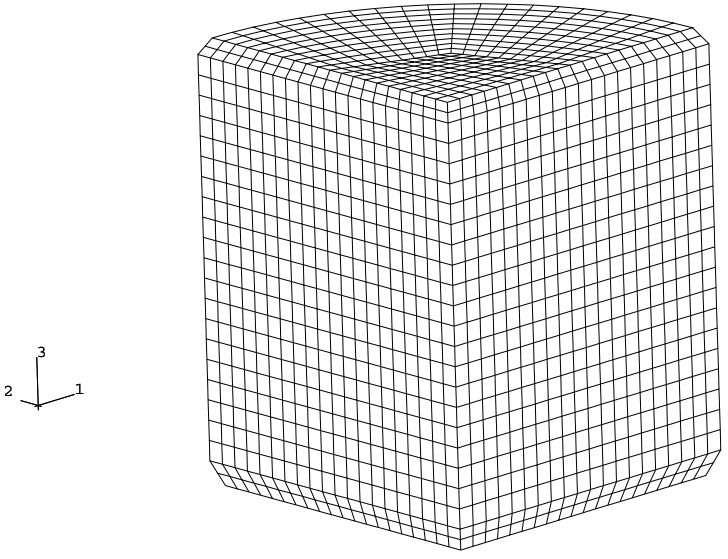


Figure 1.3.8–4 Undeformed mesh for Case 3.

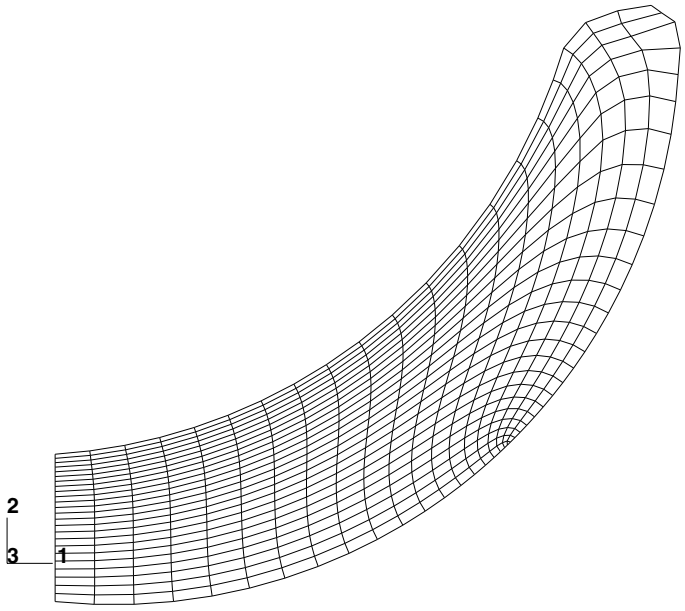


Figure 1.3.8–5 Deformed mesh for Case 1.

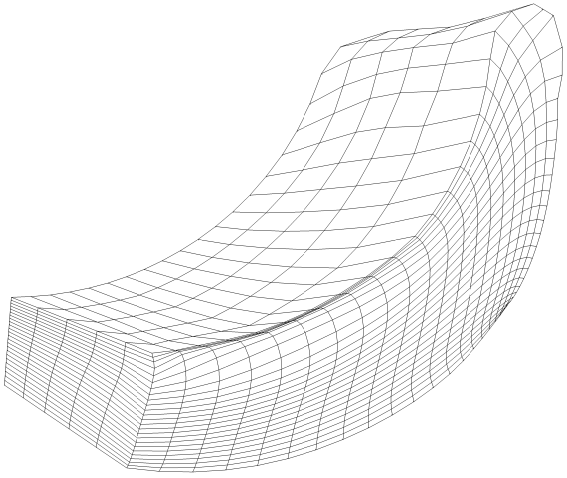


Figure 1.3.8–6 Deformed mesh for Case 2.

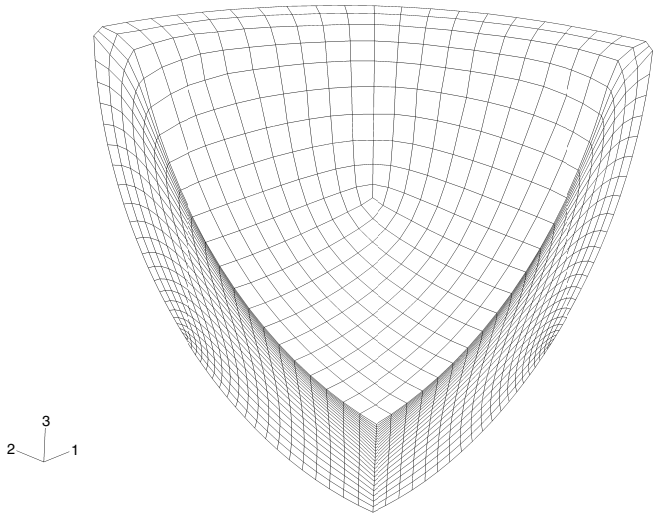


Figure 1.3.8–7 Deformed mesh for Case 3.

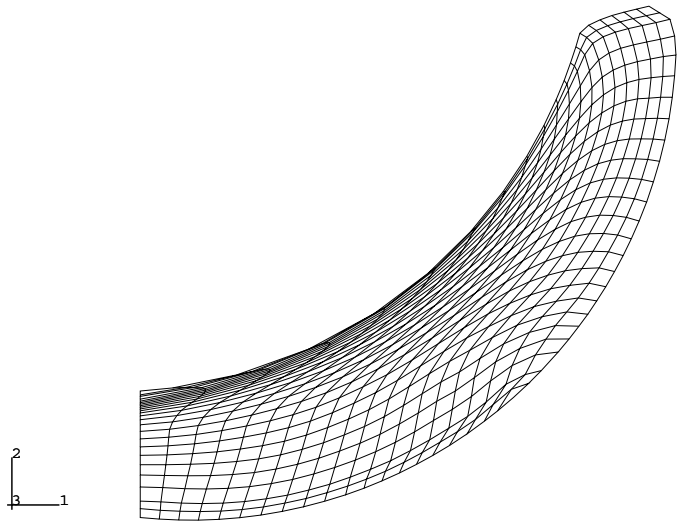


Figure 1.3.8–8 Deformed mesh for Case 1 using a pure Lagrangian formulation.

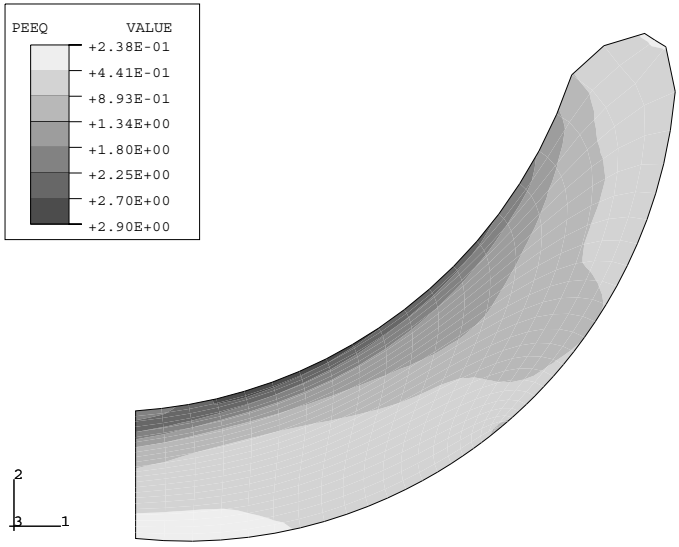


Figure 1.3.8–9 Contours of equivalent plastic strain for Case 1 using adaptive meshing.

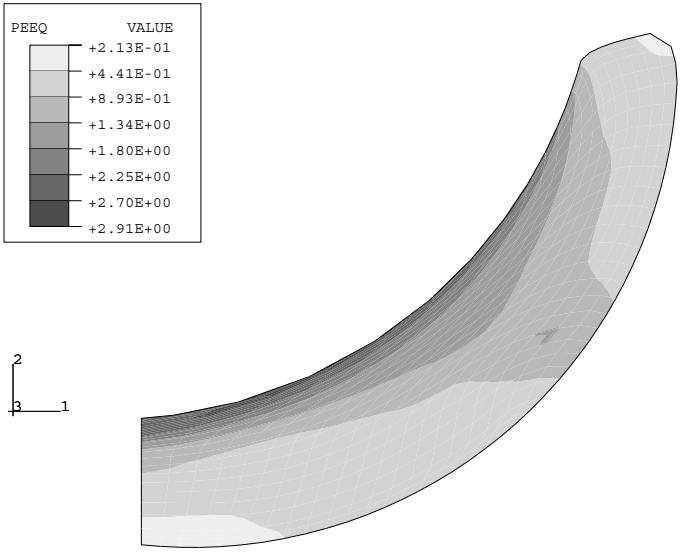


Figure 1.3.8–10 Contours of equivalent plastic strain for Case 1 using a pure Lagrangian fomulation.

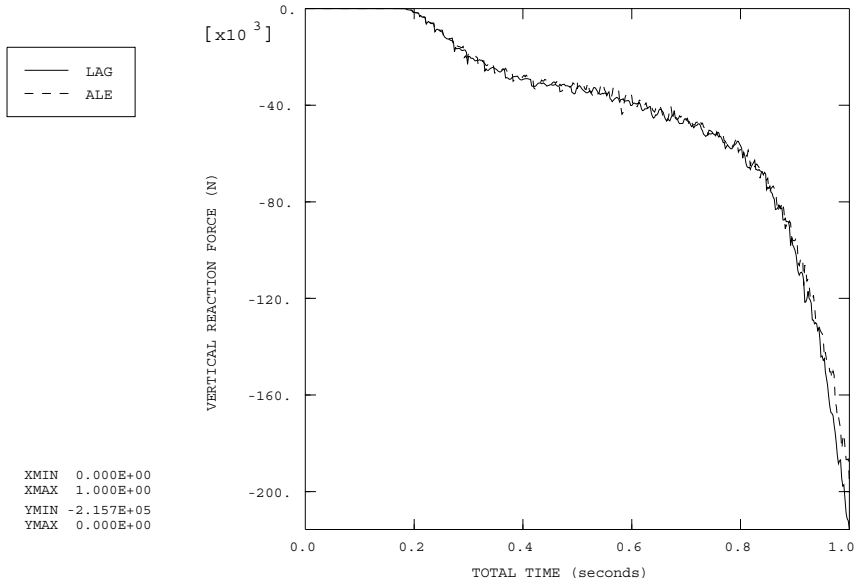


Figure 1.3.8–11 Comparison of time histories for the vertical punch force for Case 1.

1.3.9 FORGING WITH SINUSOIDAL DIES

Product: Abaqus/Explicit

This example illustrates the use of adaptive meshing in forging problems that incorporate geometrically complex dies and involve substantial material flow.

Problem description

Three different geometric models are considered, as shown in Figure 1.3.9–1. Each model consists of a rigid die and a deformable blank. The cross-sectional shape of the die is sinusoidal with an amplitude and a period of 5 and 10 mm, respectively. The blank is steel and is modeled as a von Mises elastic-plastic material with a Young's modulus of 200 GPa, an initial yield stress of 100 MPa, and a constant hardening slope of 300 MPa. Poisson's ratio is 0.3; the density is 7800 kg/m^3.

In all cases the die is moved downward vertically at a velocity of 2000 mm/sec and is constrained in all other degrees of freedom. The total die displacement is 7.6 mm for Case 1, 6.7 mm for Case 2, and 5.6 mm for Case 3. These displacements represent the maximum possible given the refinement and topology of the initial mesh (if the quality of the mesh is retained for the duration of the analysis). Although each analysis uses a sinusoidal die, the geometries and flow characteristics of the blank material are quite different for each problem.

Case 1: Axisymmetric model

The blank is meshed with CAX4R elements and measures 20 × 10 mm. The dies are modeled as analytical rigid surfaces comprised of connected line segments. The bottom of the blank is constrained in the z-direction, and symmetry boundary conditions are prescribed at r=0. The initial configuration of the blank and the die is shown in Figure 1.3.9–2.

Case 2: Three-dimensional model

The blank is meshed with C3D8R elements and measures 20 × 10 × 10 mm. The dies are modeled as three-dimensional cylindrical analytical rigid surfaces. The bottom of the blank is constrained in the y-direction, and symmetry boundary conditions are applied at the x=0 and z=10 planes. The finite element model of the blank and the die is shown in Figure 1.3.9–3.

Case 3: Three-dimensional model

The blank is meshed with C3D8R elements and measures 20 × 10 × 20 mm. The dies are modeled as three-dimensional revolved analytical rigid surfaces. The bottom of the blank is constrained in the y-direction, and symmetry boundary conditions are applied at the x=0 and z=10 planes. The finite element model of the blank and the die is shown in Figure 1.3.9–4. The revolved die is displaced upward in the figure from its initial position for clarity.

Adaptive meshing

A single adaptive mesh domain that incorporates the entire blank is used for each model. Symmetry planes are defined as Lagrangian boundary regions (the default), and contact surfaces are defined as sliding boundary regions (the default). Because the material flow for each of the geometries is substantial, the frequency and the intensity of adaptive meshing must be increased to provide an accurate solution. The frequency at which adaptive meshing is to be performed is reduced from the default of 10 to 5 for all cases. The number of mesh sweeps is increased from the default of 1 to 3 for all cases.

Results and discussion

Figure 1.3.9–5 and Figure 1.3.9–6 show the deformed mesh and contours of equivalent plastic strain at the completion of the forming step for Case 1. Adaptive meshing maintains reasonable element shapes and aspect ratios. This type of forging problem cannot typically be solved using a pure Lagrangian formulation. Figure 1.3.9–7 shows the deformed mesh for Case 2. A complex, doubly curved deformation pattern is formed on the free surface as the material spreads under the die. Element distortion appears to be reasonable. Figure 1.3.9–8 and Figure 1.3.9–9 show the deformed mesh and contours of equivalent plastic strain for Case 3. Although the die is a revolved geometry, the three-dimensional nature of the blank gives rise to fairly complex strain patterns that are symmetric with respect to the planes of quarter symmetry.

Input files

ale_sinusoid_forgingaxi.inp	Case 1.
ale_sinusoid_forgingaxisurf.inp	External file referenced by Case 1.
ale_sinusoid_forgingcyl.inp	Case 2.
ale_sinusoid_forgingrev.inp	Case 3.

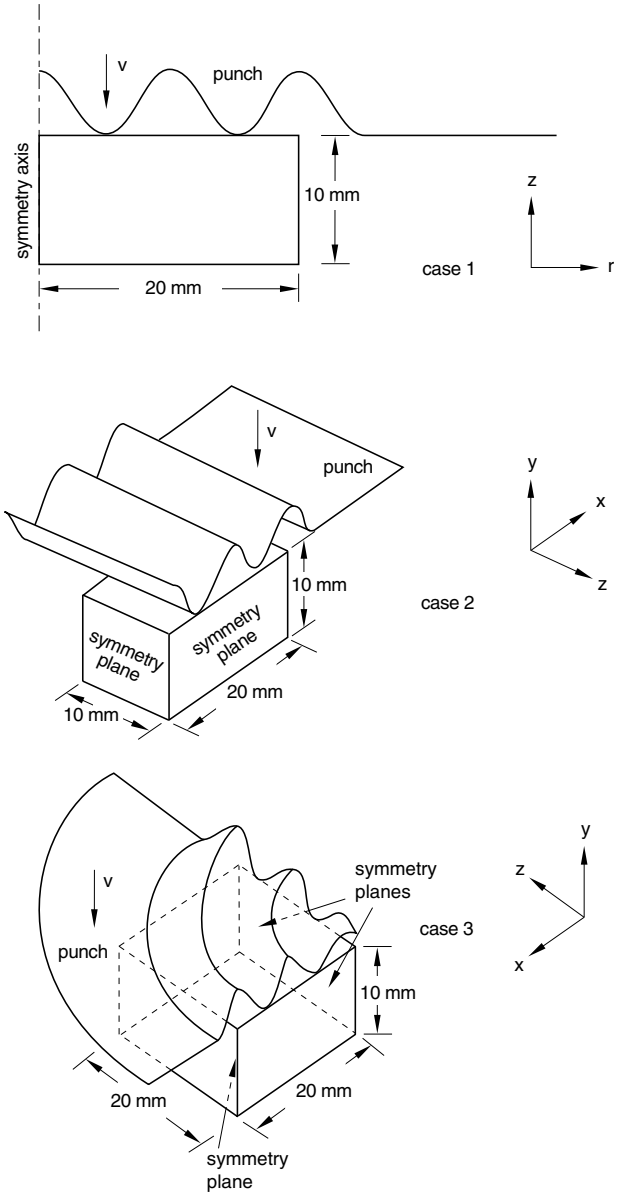


Figure 1.3.9–1 Model geometries for each of the three cases.

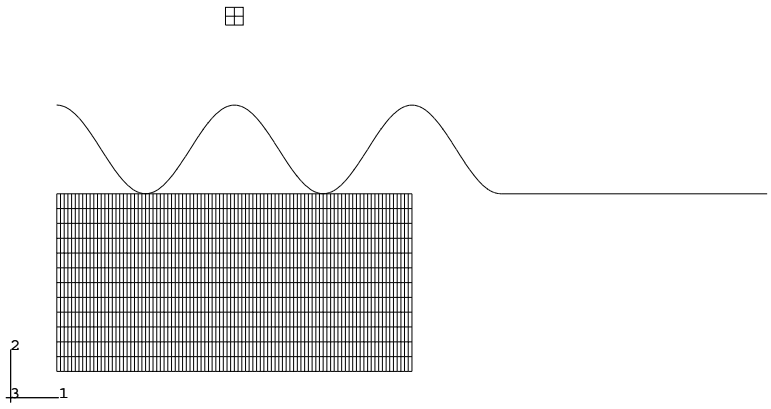


Figure 1.3.9–2 Initial configuration for Case 1.

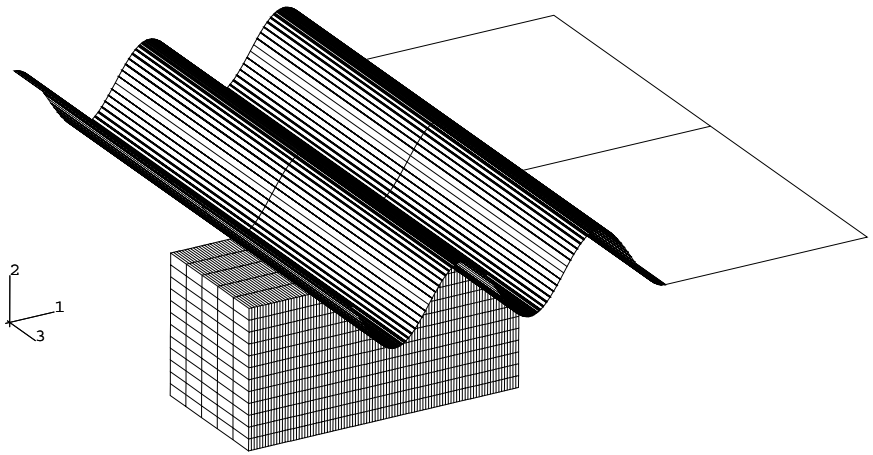


Figure 1.3.9–3 Initial configuration for Case 2.

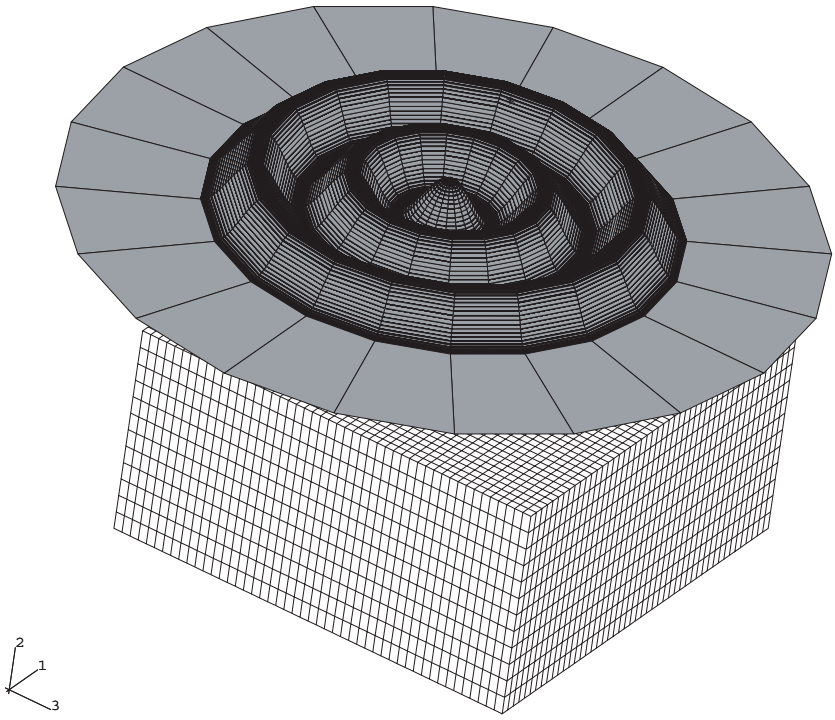


Figure 1.3.9–4 Initial configuration for Case 3.

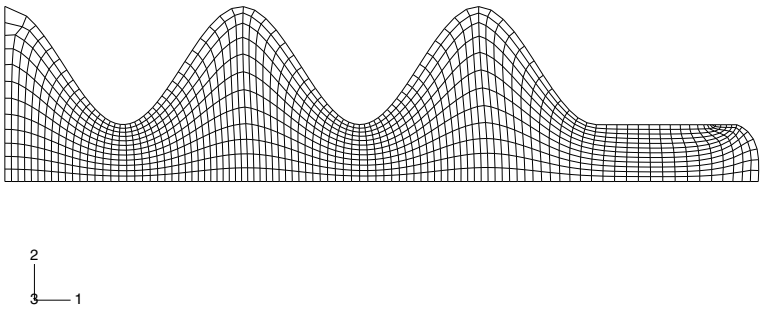


Figure 1.3.9–5 Deformed mesh for Case 1.

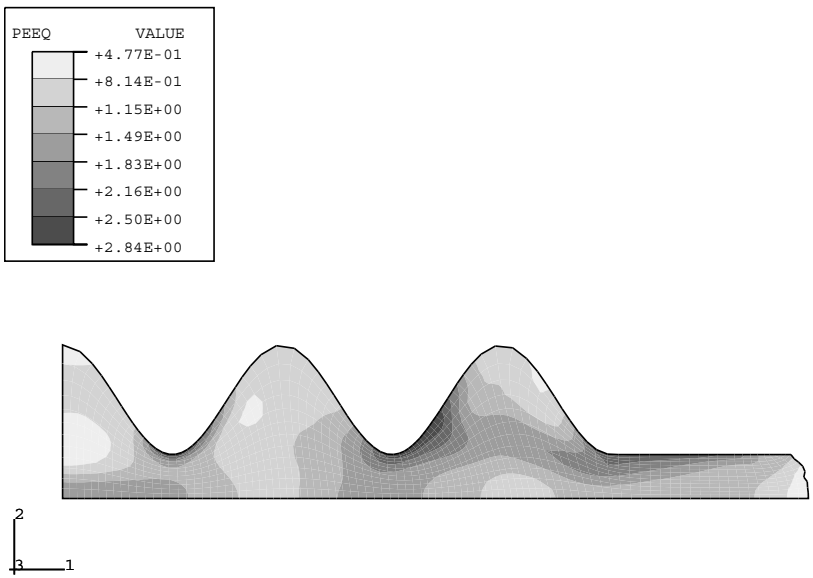


Figure 1.3.9–6 Contours of equivalent plastic strain for Case 1.

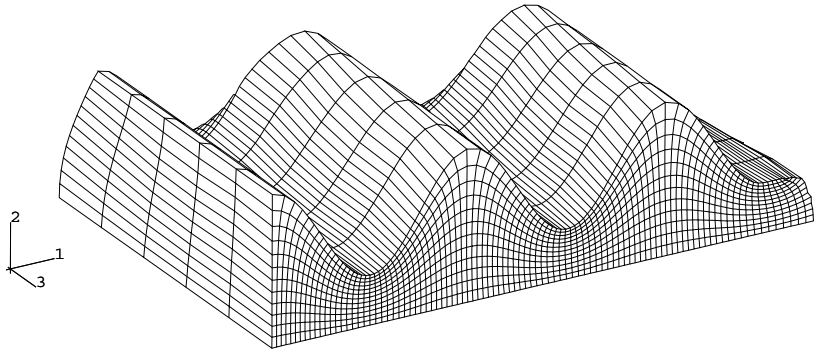


Figure 1.3.9–7 Deformed mesh for Case 2.

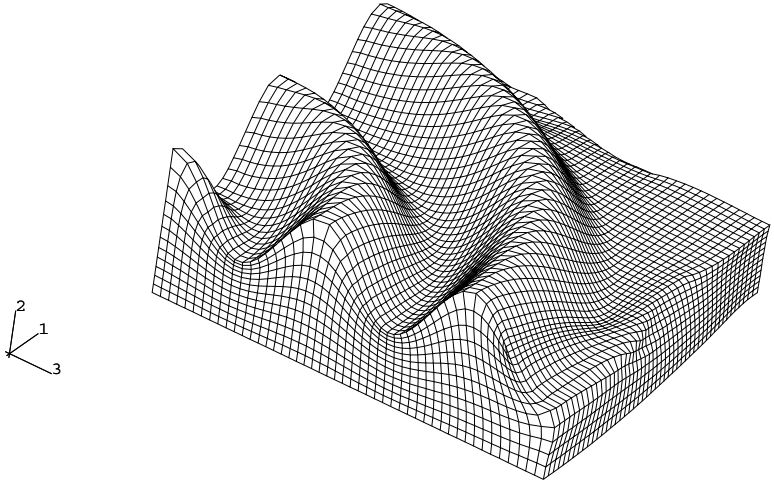


Figure 1.3.9–8 Deformed mesh for Case 3.

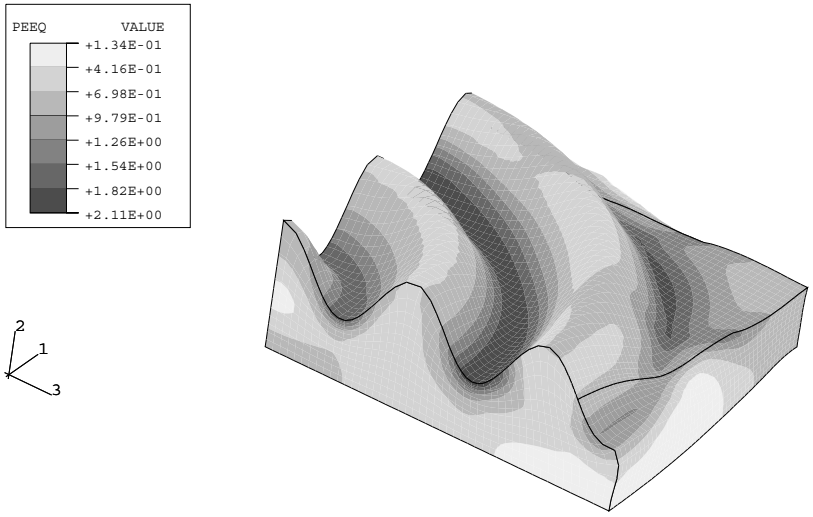


Figure 1.3.9–9 Contours of equivalent plastic strain for Case 3.

1.3.10 FORGING WITH MULTIPLE COMPLEX DIES

Product: Abaqus/Explicit

This example illustrates the use of adaptive meshing in forging problems that use multiple geometrically complex dies. The problem is based on a benchmark presented at the "FEM–Material Flow Simulation in the Forging Industry" workshop.

Problem description

The benchmark problem is an axisymmetric forging, but in this example both axisymmetric and three-dimensional geometric models are considered. For the axisymmetric models the default hourglass formulation (HOURGLASS=RELAX STIFFNESS) and the enhanced strain hourglass formulation (HOURGLASS=ENHANCED) are considered. For the three-dimensional geometric models the pure stiffness hourglass formulation (HOURGLASS=STIFFNESS) and the enhanced strain hourglass formulation with the orthogonal kinematic formulation (KINEMATIC SPLIT=ORTHOGONAL) are considered. Each model is shown in Figure 1.3.10–1. Both models consist of two rigid dies and a deformable blank. The blank's maximum radial dimension is 895.2 mm, and its thickness is 211.4 mm. The outer edge of the blank is rounded to facilitate the flow of material through the dies. The blank is modeled as a von Mises elastic-plastic material with a Young's modulus of 200 GPa, an initial yield stress of 360 MPa, and a constant hardening slope of 30 MPa. The Poisson's ratio is 0.3; the density is 7340 kg/m^3.

Both dies are fully constrained, with the exception of the top die, which is moved 183.4 mm downward at a constant velocity of 166.65 mm/s.

Case 1: Axisymmetric model

The blank is meshed with CAX4R elements. A fine discretization is required in the radial direction because of the geometric complexity of the dies and the large amount of material flow that occurs in that direction. Symmetry boundary conditions are prescribed at r=0. The dies are modeled as TYPE=SEGMENTS analytical rigid surfaces. The initial configuration is shown in Figure 1.3.10–2.

Case 2: Three-dimensional model

The blank is meshed with C3D8R elements. A 90° wedge of the blank is analyzed. The level of mesh refinement is the same as that used in the axisymmetric model. Symmetry boundary conditions are applied at the x=0 and z=0 planes. The dies are modeled as TYPE=REVOLUTION analytical rigid surfaces. The initial configuration of the blank only is shown in Figure 1.3.10–3. Although the tools are not shown in the figure, they are originally in contact with the blank.

Adaptive meshing

A single adaptive mesh domain that incorporates the entire blank is used for each model. Symmetry planes are defined as Lagrangian boundary regions (the default), and contact surfaces are defined as

sliding boundary regions (the default). Since this problem is quasi-static with relatively small amounts of deformation per increment, the defaults for frequency, mesh sweeps, and other adaptive mesh parameters and controls are sufficient.

Results and discussion

Figure 1.3.10–4 and Figure 1.3.10–5 show the deformed mesh for the axisymmetric case using the default hourglass control formulation (HOURGLASS=RELAX STIFFNESS) at an intermediate stage ($t = 0.209$ s) and in the final configuration ($t = 0.35$ s), respectively. The elements remain well shaped throughout the entire simulation, with the exception of the elements at the extreme radius of the blank, which become very coarse as material flows radially during the last 5% of the top die's travel. Figure 1.3.10–6 shows contours of equivalent plastic strain at the completion of forming. Figure 1.3.10–7 and Figure 1.3.10–8 show the deformed mesh for the three-dimensional case using the pure stiffness hourglass control (HOURGLASS=STIFFNESS) and the orthogonal kinematic formulation (KINEMATIC SPLIT=ORTHOGONAL) at $t = 0.209$ and $t = 0.35$, respectively. Although the axisymmetric and three-dimensional mesh smoothing algorithms are not identical, the elements in the three-dimensional model also remain well shaped until the end of the analysis, when the same behavior that is seen in the two-dimensional model occurs. Contours of equivalent plastic strain for the three-dimensional model (not shown) are virtually identical to those shown in Figure 1.3.10–6.

Input files

ale_duckshape_forgingaxi.inp	Case 1 using the default hourglass formulation (HOURGLASS=RELAX STIFFNESS).
ale_duckshape_forgingaxi_enhs.inp	Case 1 using the enhanced strain hourglass formulation (HOURGLASS=ENHANCED).
ale_duckshape_forg_axind.inp	External file referenced by the Case 1 analyses.
ale_duckshape_forg_axiel.inp	External file referenced by the Case 1 analyses.
ale_duckshape_forg_axiset.inp	External file referenced by the Case 1 analyses.
ale_duckshape_forg_axirs.inp	External file referenced by the Case 1 analyses.
ale_duckshape_forgingrev.inp	Case 2 using the pure stiffness hourglass formulation (HOURGLASS=STIFFNESS) and the orthogonal kinematic formulation (KINEMATIC SPLIT=ORTHOGONAL).
ale_duckshape_forgingrev_oenhs.inp	Case 2 using the enhanced strain hourglass formulation (HOURGLASS=ENHANCED) and the orthogonal kinematic formulation (KINEMATIC SPLIT=ORTHOGONAL).

Reference

- Industrieverband Deutscher Schmieden e.V. (IDS), "Forging of an Axisymmetric Disk," FEM–Material Flow Simulation in the Forging Industry, Hagen, Germany, October 1997.

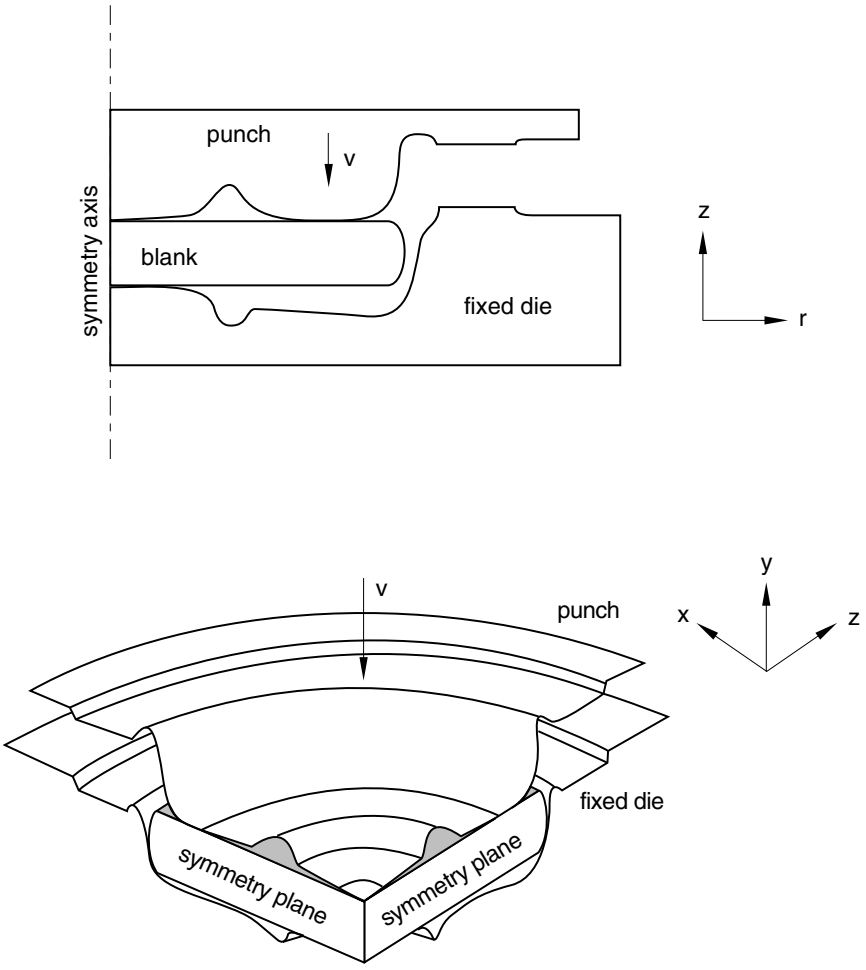


Figure 1.3.10–1 Axisymmetric and three-dimensional model geometries.

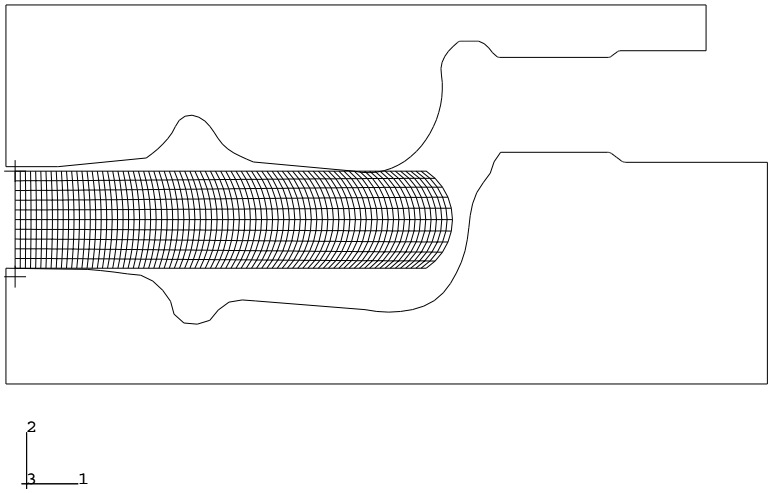


Figure 1.3.10–2 Initial configuration for the axisymmetric model.

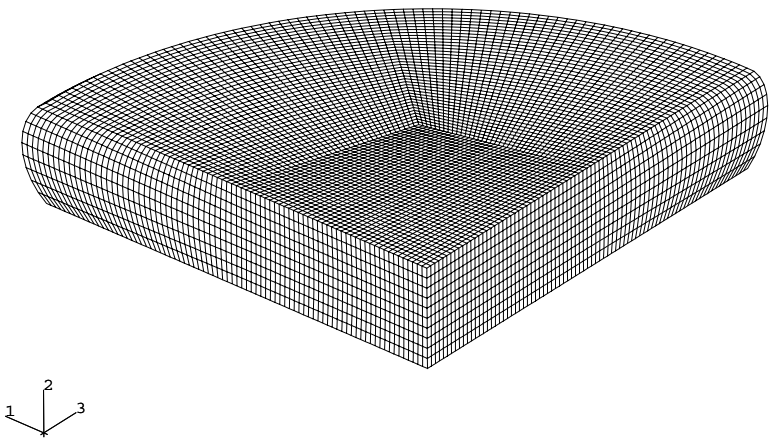


Figure 1.3.10–3 Initial configuration mesh for the three-dimensional model.

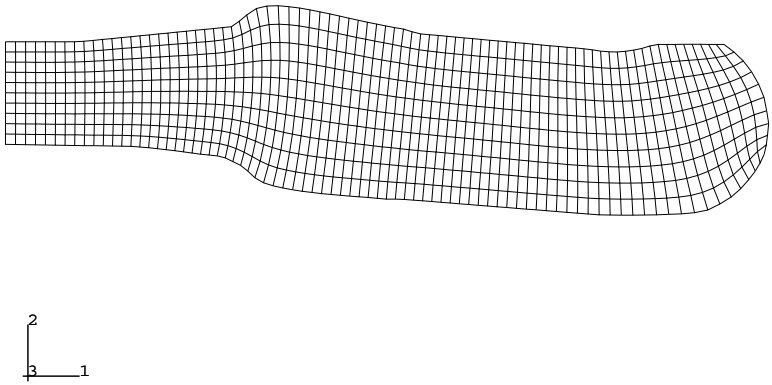


Figure 1.3.10–4 The deformed mesh for the axisymmetric model using the default hourglass formulation (HOURGLASS=RELAX STIFFNESS) at an intermediate stage.

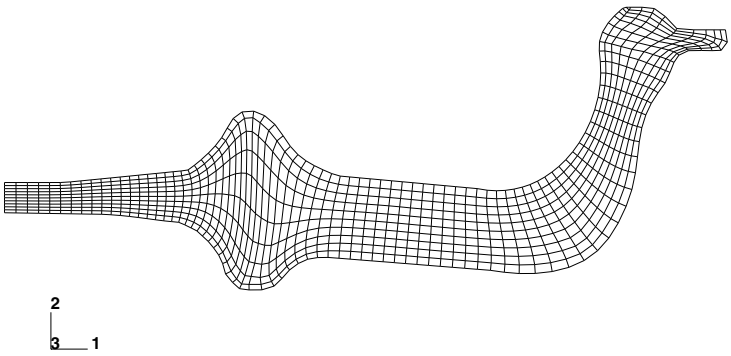


Figure 1.3.10–5 The deformed mesh for the axisymmetric model using the default hourglass formulation (HOURGLASS=RELAX STIFFNESS) at the end of forming.

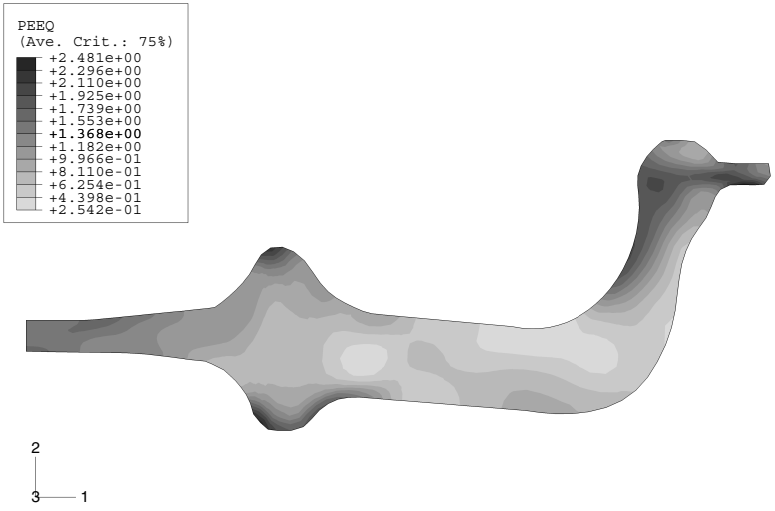


Figure 1.3.10–6 Contours of equivalent plastic strain for the axisymmetric model using the default hourglass formulation (HOURGLASS=RELAX STIFFNESS) at the end of forming.

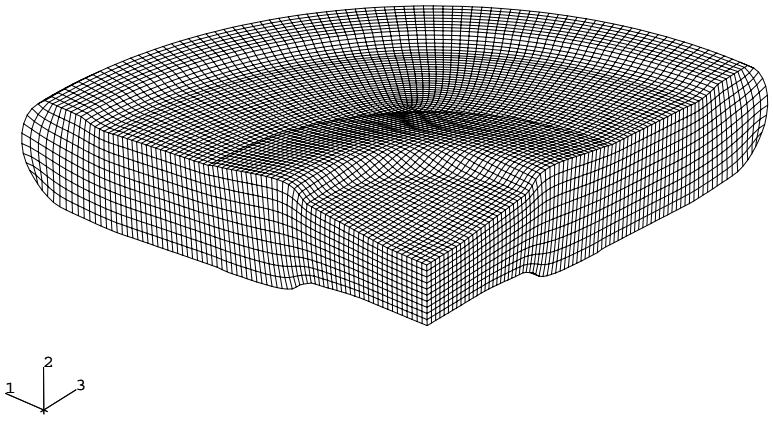


Figure 1.3.10–7 The deformed mesh for the three-dimensional model using the pure stiffness hourglass formulation (HOURGLASS=STIFFNESS) and the orthogonal kinematic formulation (KINEMATIC SPLIT=ORTHOGONAL) at an intermediate stage.

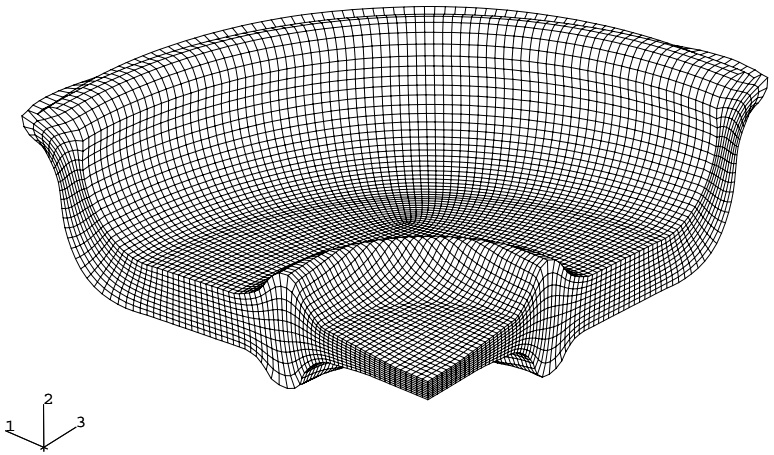


Figure 1.3.10–8 The deformed mesh for the three-dimensional model using the pure stiffness hourglass formulation (HOURGLASS=STIFFNESS) and the orthogonal kinematic formulation (KINEMATIC SPLIT=ORTHOGONAL) at the end of forming.

1.3.11 FLAT ROLLING: TRANSIENT AND STEADY-STATE

Product: Abaqus/Explicit

This example illustrates the use of adaptive meshing to simulate a rolling process using both transient and steady-state approaches, as shown in Figure 1.3.11–1. A transient flat rolling simulation is performed using three different methods: a "pure" Lagrangian approach, an adaptive meshing approach using a Lagrangian domain, and a mixed Eulerian-Lagrangian adaptive meshing approach in which material upstream from the roller is drawn from an Eulerian inflow boundary but the downstream end of the blank is handled in a Lagrangian manner. In addition, a steady-state flat rolling simulation is performed using an Eulerian adaptive mesh domain as a control volume and defining inflow and outflow Eulerian boundaries. Solutions using each approach are compared.

Problem description

For each analysis case quarter symmetry is assumed; the model consists of a rigid roller and a deformable blank. The blank is meshed with C3D8R elements. The cylindrical roller is modeled as an analytical rigid surface. The radius of the cylinder is 175 mm. Symmetry boundary conditions are prescribed on the right (z=0 plane) and bottom (y=0 plane) faces of the blank.

Coulomb friction with a friction coefficient of 0.3 is assumed between the roller and the plate. All degrees of freedom are constrained on the roller except rotation about the z-axis, where a constant angular velocity of 6.28 rad/sec is defined. For each analysis case the blank is given an initial velocity of 0.3 m/s in the x-direction to initiate contact.

The blank is steel and is modeled as a von Mises elastic-plastic material with isotropic hardening. The Young's modulus is 150 GPa, and the initial yield stress is 168.2 MPa. The Poisson's ratio is 0.3; the density is 7800 kg/m^3. The masses of all blank elements are scaled by a factor of 2750 at the beginning of the step so that the analysis can be performed more economically. This scaling factor represents an approximate upper bound on the mass scaling possible for this problem, above which significant inertial effects would be generated.

The criteria for stopping the rolling analyses based on the achievement of a steady-state condition is defined. The criteria used require the satisfaction of the steady-state detection norms of equivalent plastic strain, spread, force, and torque within the default tolerances. The exit plane for each norm is defined as the plane passing through the center of the roller with the normal to the plane coincident with the rolling direction. For Case 1 through Case 3 the steady-state detection norms are evaluated as each plane of elements passes the exit plane. Case 4 requires uniform sampling since the initial mesh is roughly stationary due to the initial geometry and the inflow and outflow Eulerian boundaries.

The finite element models used for each analysis case are shown in Figure 1.3.11–2. A description of each model and the adaptive meshing techniques used follows:

Case 1: Transient simulation—pure Lagrangian approach

The blank is initially rectangular and measures 224 × 20 × 50 mm. No adaptive meshing is performed. The analysis is run until steady-state conditions are achieved.

Case 2: Transient simulation—Lagrangian adaptive mesh domain

The finite element model for this case is identical to that used for Case 1, with the exception that a single adaptive mesh domain that incorporates the entire blank is defined to allow continuous adaptive meshing. Symmetry planes are defined as Lagrangian surfaces (the default), and the contact surface on the blank is defined as a sliding surface (the default). The analysis is run until steady-state conditions are achieved.

Case 3: Transient simulation—mixed Eulerian-Lagrangian approach

This analysis is performed on a relatively short initial blank measuring 65 × 20 × 50 mm. Material is continuously drawn by the action of the roller on the blank through an inflow Eulerian boundary defined on the upstream end. The blank is meshed with the same number of elements as in Cases 1 and 2 so that similar aspect ratios are obtained as the blank lengthens and steady-state conditions are achieved.

An adaptive mesh domain is defined that incorporates the entire blank. Because it contains at least one Eulerian surface, this domain is considered Eulerian for the purpose of setting parameter defaults. However, the analysis model has both Lagrangian and Eulerian aspects. The amount of material flow with respect to the mesh will be large at the inflow end and small at the downstream end of the domain. To account for the Lagrangian motion of the downstream end, change the adaptive mesh controls for this problem so that adaptive meshing is performed based on the positions of the nodes at the start of the current adaptive mesh increment. To mesh the inflow end accurately and to perform the analysis economically, the frequency is set to 5 and the number of mesh sweeps is set to 5.

As in Case 2, symmetry planes are defined as Lagrangian boundary regions (the default), and the contact surface on the blank is defined as a sliding boundary region (the default). In addition, the boundary on the upstream end is defined as an Eulerian surface. Adaptive mesh constraints are defined on the Eulerian surface using adaptive mesh constraints to hold the inflow surface mesh completely fixed while material is allowed to enter the domain normal to the surface. An equation constraint is used to ensure that the velocity normal to the inflow boundary is uniform across the surface. The velocity of nodes in the direction tangential to the inflow boundary surface is constrained.

Case 4: Steady-state simulation—Eulerian adaptive mesh domain

This analysis employs a control volume approach in which material is drawn from an inflow Eulerian boundary and is pushed out through an outflow boundary by the action of the roller. The blank geometry for this analysis case is defined such that it approximates the shape corresponding to the steady-state solution: this geometry can be thought of as an "initial guess" to the solution. The blank initially measures 224 mm in length and 50 mm in width and has a variable thickness such that it conforms to the shape of the roller. The surface of the blank transverse to the rolling direction is not adjusted to account for the eventual spreading that will occur in the steady-state solution. Actually, any reasonable initial geometry will reach a steady state, but geometries that are closer to the steady-state geometry often allow a solution to be obtained in a shorter period of time.

As in the previous two cases an adaptive mesh domain is defined on the blank, symmetry planes are defined as Lagrangian surfaces (the default), and the contact surface is defined as a sliding surface (the default). Inflow and outflow Eulerian surfaces are defined on the ends of the blank using the same

techniques as in Case 3, except that for the outflow boundary adaptive mesh constraints are applied only normal to the boundary surface and no material constraints are applied tangential to the boundary surface.

To improve the computational efficiency of the analysis, the frequency of adaptive meshing is increased to every fifth increment because the Eulerian domain undergoes very little overall deformation and the material flow speed is much less than the material wave speed. This frequency will cause the mesh at Eulerian boundaries to drift slightly. However, the amount of drift is extremely small and does not accumulate. There is no need to increase the mesh sweeps because this domain is relatively stationary and, by default, adaptive meshing is performed based on the nodal positions of the original mesh. Very little mesh smoothing is required.

Results and discussion

The final deformed configurations of the blank for each of the three transient cases are shown in Figure 1.3.11–3. The transient cases have reached a steady-state solution and have been terminated based on the criteria given in the steady-state detection definition. Steady-state conditions are determined to have been reached when the reaction forces and moments on the roller have stabilized and the cross-sectional shape and distribution of equivalent plastic strain under the roller become constant over time. When using a steady-state detection definition, these conditions imply that the force, moment, spread, and equivalent plastic strain norms have stabilized such that the changes in the norms over three consecutive sampling intervals have fallen below the user-prescribed tolerances. See "Steady-state detection," Section 11.8.1 of the Abaqus Analysis User's Guide, for a detailed discussion on the definition of the norms. Contours of equivalent plastic strain for each of the three transient cases are in good agreement and are shown in the final configuration of each blank in Figure 1.3.11–4. Figure 1.3.11–5 shows the initial and final mesh configurations at steady state. With the exception of Case 3 all analyses were terminated using the default steady-state norm tolerances. Case 3 required that the force and torque norm tolerances be increased from .005 to .01 due to the force and torque at the roller being rather noisy.

To compare the results from the transient and steady-state approaches, the steady-state detection norms are summarized for each case in Table 1.3.11–1. The table shows a comparison of the values of the steady-state detection norms after the analyses have been terminated. The only significant difference is in the value of the spread norm for Case 4, which is higher than the others. The spread norm is defined as the largest of the second principle moments of inertia of the workpiece's cross-section. Since the spread norm is a cubic function of the lateral deformation of the workpiece, rather small differences in displacements between the test cases can lead to significant differences in the spread norms.

Time history plots of the steady-state detection norms are also shown. Figure 1.3.11–9 and Figure 1.3.11–10 show time history plots of the steady-state force and torque norms, respectively, for all cases. The force and torque norms are essentially running averages of the force and moment on the roller and show good agreement for all four test cases. Figure 1.3.11–7 and Figure 1.3.11–8 show time history plots of the steady-state equivalent plastic strain and spread norms, respectively, for all cases. The equivalent plastic strains norms are in good agreement for all cases.

Input files

lag_flatrolling.inp	Case 1 with contact pairs.
lag_flatrolling_gcont.inp	Case 1 with general contact.
ale_flatrolling_noeuler.inp	Case 2.
ale_flatrolling_inlet.inp	Case 3.
ale_flatrolling_inletoutlet.inp	Case 4.

Table 1.3.11–1 Comparison of steady-state detection norms.

Formulation	Spread norm	Effective plastic strain norm	Force norm	Torque norm
Case 1	1.349 E−7	.8037	−1.43 E6	3.59 E4
Case 2	1.369 E−7	.8034	−1.43 E6	3.55 E4
Case 3	1.365 E−7	.8018	−1.43 E6	3.61 E4
Case 4	1.485 E−7	.8086	−1.40 E6	3.65 E4

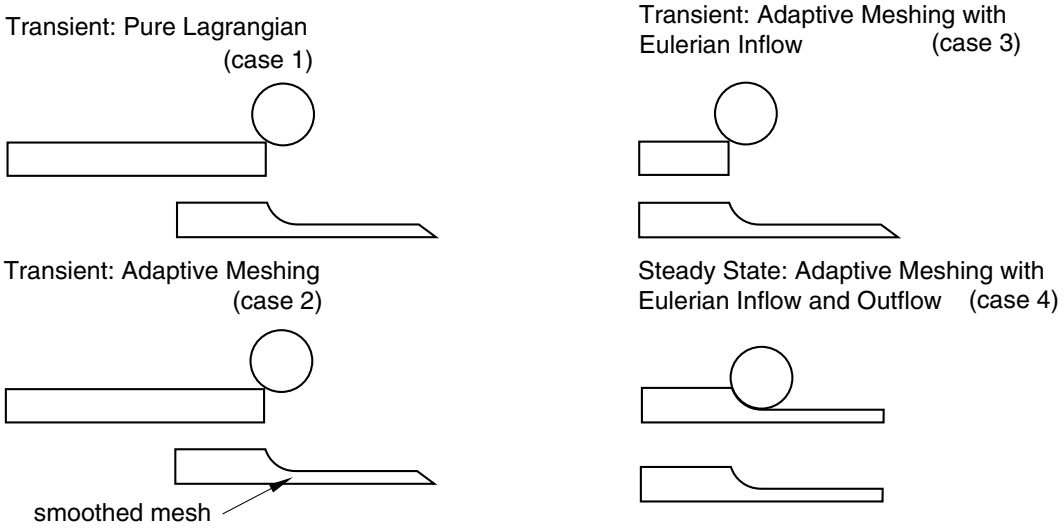


Figure 1.3.11–1 Diagram illustrating the four analysis approaches used in this problem.

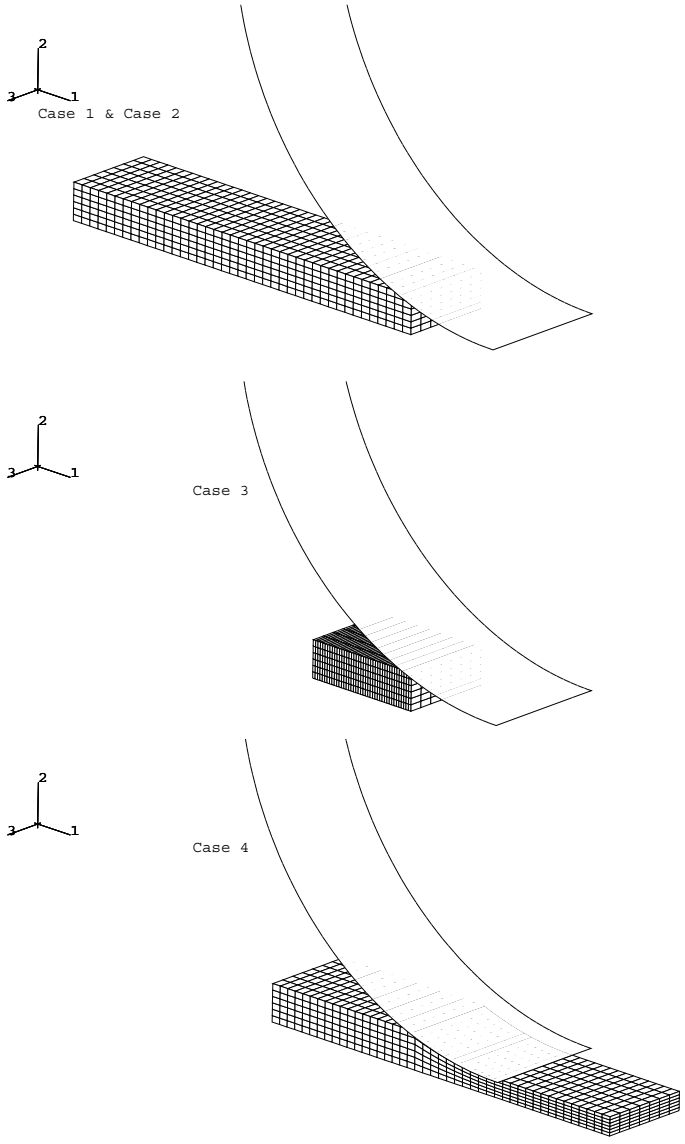


Figure 1.3.11–2 Initial configurations for each case.

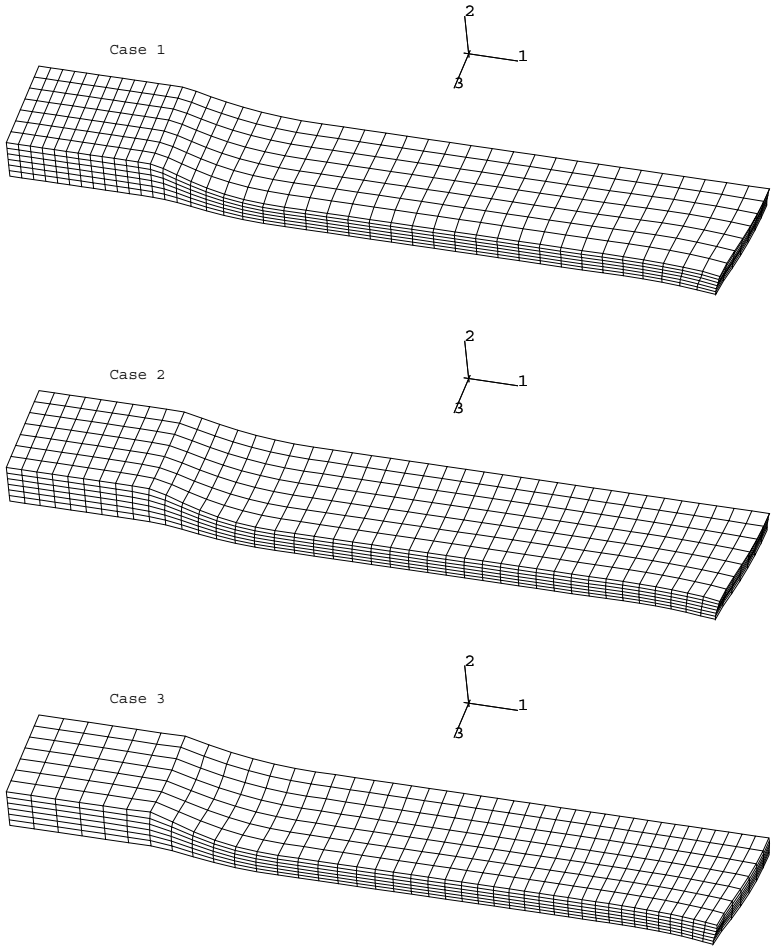


Figure 1.3.11–3 Deformed mesh for Cases 1–3.

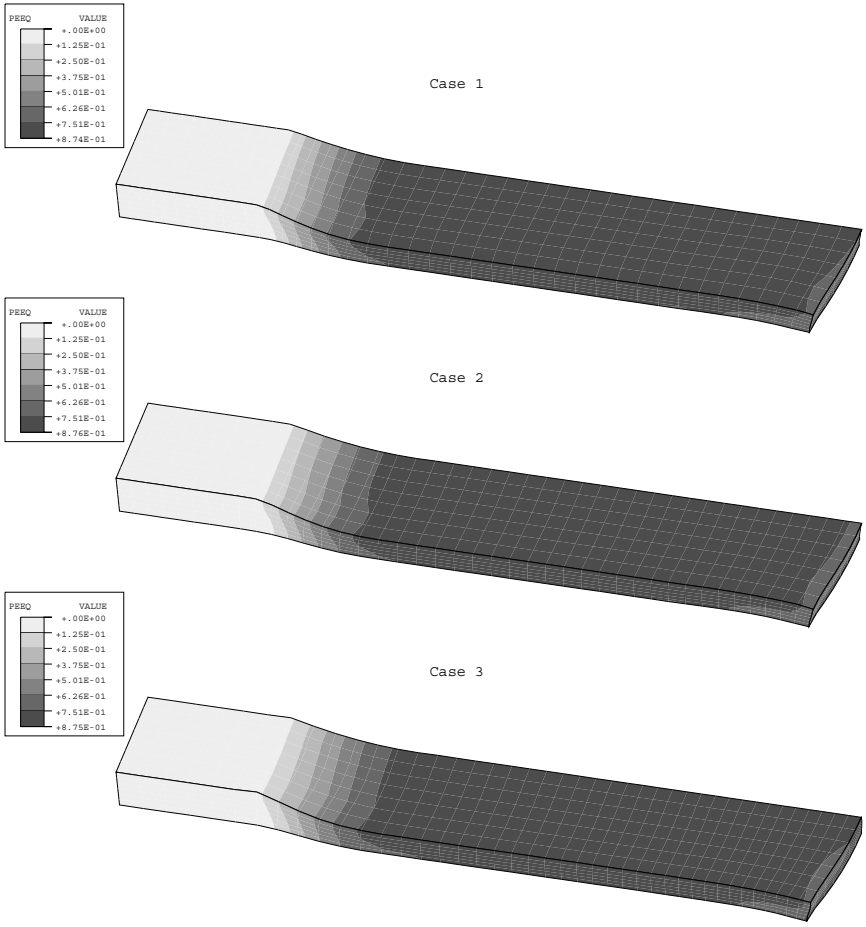


Figure 1.3.11–4 Contours of equivalent plastic strain for Cases 1–3.

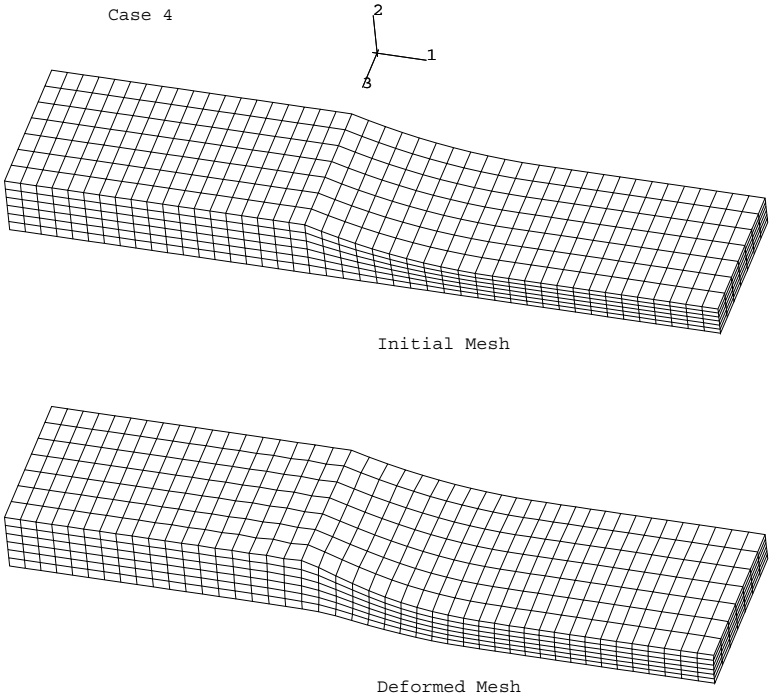


Figure 1.3.11–5 Deformed mesh for Case 4 (shown with initial mesh for comparison).

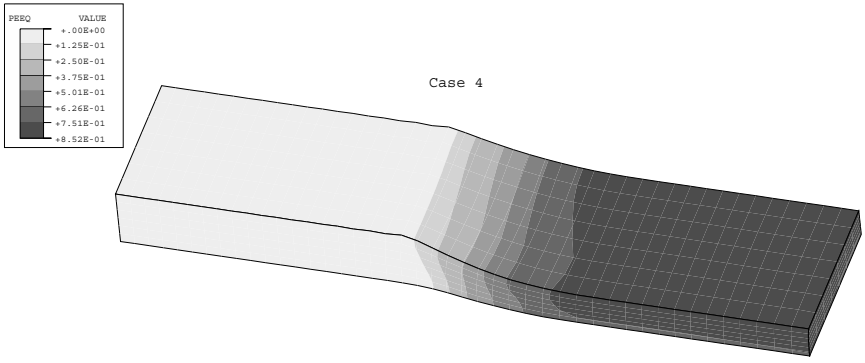


Figure 1.3.11–6 Contours of equivalent plastic strain for Case 4.

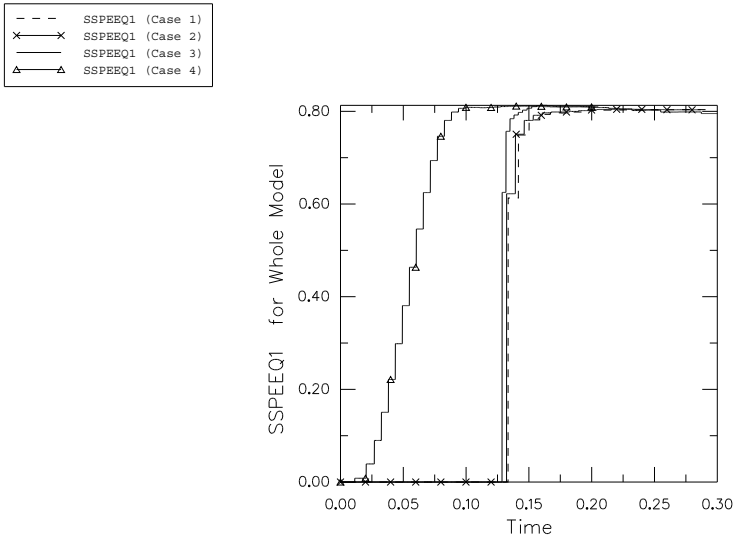


Figure 1.3.11–7 Comparison of equivalent plastic strain norm versus time for all cases.

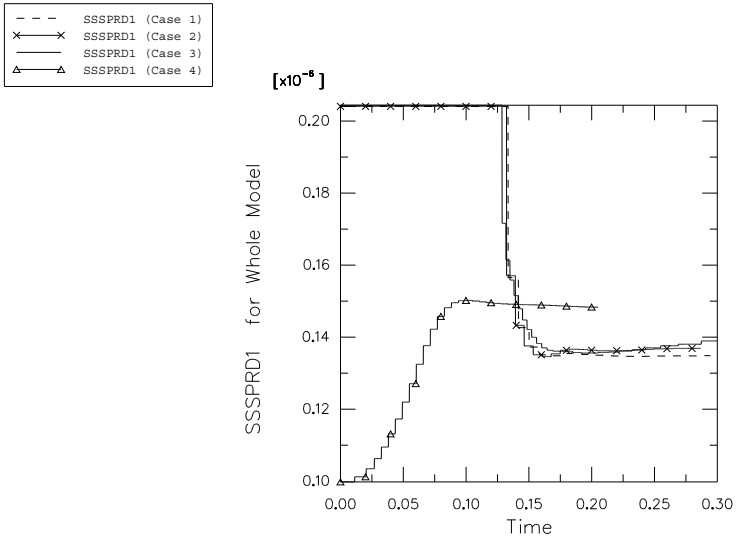


Figure 1.3.11–8 Comparison of spread norm versus time for all cases.

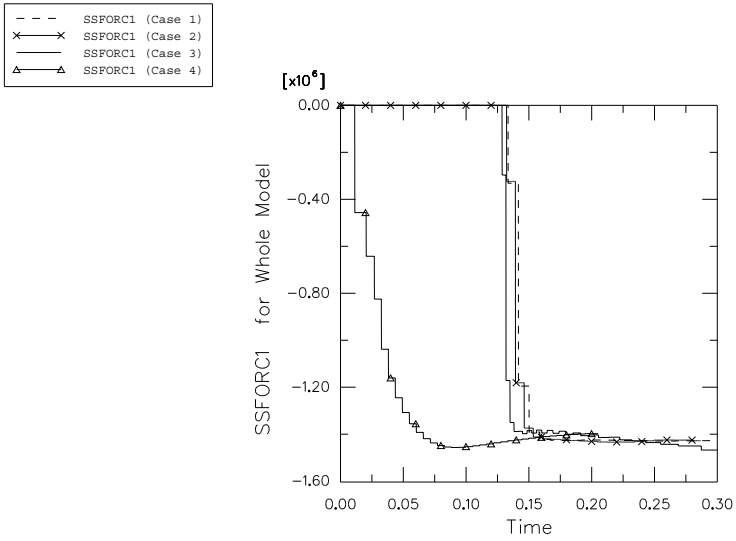


Figure 1.3.11–9 Comparison of force norm versus time for all cases.

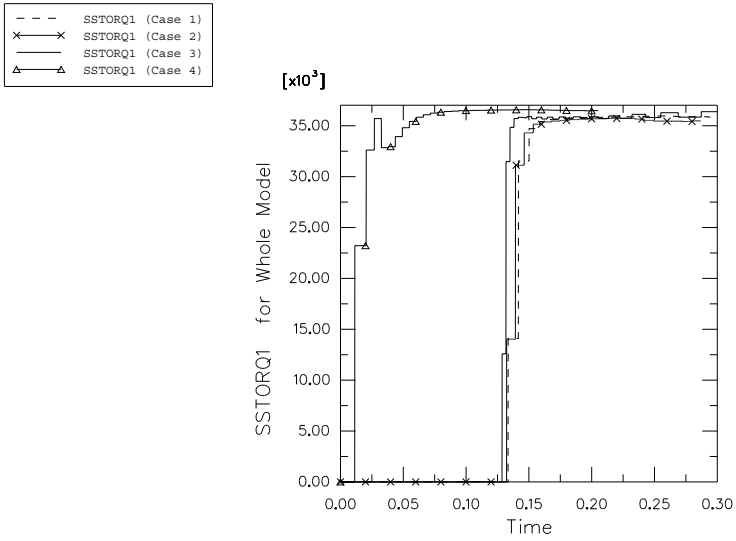


Figure 1.3.11–10 Comparison of torque norm versus time for all cases.

1.3.12 SECTION ROLLING

Product: Abaqus/Explicit

This example illustrates the use of adaptive meshing in a transient simulation of section rolling. Results are compared to a pure Lagrangian simulation.

Problem description

This analysis shows a stage in the rolling of a symmetric I-section. Because of the cross-sectional shape of the I-section, two planes of symmetry exist and only a quarter of the section needs to be modeled. The quarter-symmetry model, shown in Figure 1.3.12–1, consists of two rigid rollers and a blank. Roller 1 has a radius of 747 mm, and roller 2 has a radius of 452 mm. The blank has a length of 850 mm, a web half-width of 176.7 mm, a web half-thickness of 24 mm, and a variable flange thickness.

The finite element model is shown in Figure 1.3.12–2. The blank is meshed with C3D8R elements. Symmetry boundary conditions are applied on the y and z symmetry planes of the blank. The rollers are modeled as three-dimensional revolved analytical rigid surfaces. Roller 1 has all degrees of freedom constrained except rotation about the z-axis, where a constant angular velocity of 5 rad/sec is specified. Roller 2 has all degrees of freedom constrained except rotation about the y-axis. An initial velocity of 4187.0 mm/sec in the negative x-direction is applied to the blank to initiate contact between the blank and the rollers. This velocity corresponds to the velocity of the rollers at the point of initial contact.

Variable mass scaling is used to scale the masses of all the blank elements so that a desired minimum stable time increment is achieved initially and the stable time increment does not fall below this minimum throughout the analysis. The loading rates and mass scaling definitions are such that a quasi-static solution is generated.

The blank is steel and is modeled as a von Mises elastic-plastic material with a Young's modulus of 212 GPa, an initial yield stress of 80 MPa, and a constant hardening slope of 258 MPa. Poisson's ratio is 0.281; the density is 7833 kg/m^3. Coulomb friction with a friction coefficient of 0.3 is assumed between the rollers and the blank.

Adaptive meshing

Adaptive meshing can improve the solution and mesh quality for section rolling problems that involve large deformations. A single adaptive mesh domain that incorporates the entire blank is defined. Symmetry planes are defined as Lagrangian boundary regions (the default), and the contact surface on the blank is defined as a sliding boundary region (the default). The default values are used for all adaptive mesh parameters and controls.

Results and discussion

Figure 1.3.12–3 shows the deformed configuration of the blank when continuous adaptive meshing is used. For comparison purposes a pure Lagrangian simulation is performed. Figure 1.3.12–4 shows the deformed configuration for a pure Lagrangian simulation. The mesh at the flange-web interface is

distorted in the Lagrangian simulation, but the mesh remains nicely proportioned in the adaptive mesh analysis. A close-up view of the deformed configuration of the blank is shown for each analysis in Figure 1.3.12–5 and Figure 1.3.12–6 to highlight the differences in mesh quality. Contours of equivalent plastic strain for each analysis are shown in Figure 1.3.12–7 and Figure 1.3.12–8. The plastic strain distributions are very similar.

Figure 1.3.12–9 and Figure 1.3.12–10 show time history plots for the *y*-component of reaction force and the reaction moment about the *z*-axis, respectively, for roller 1. The results for the adaptive mesh simulation compare closely to those for the pure Lagrangian simulation.

Input files

ale_rolling_section.inp	Analysis that uses adaptive meshing.
ale_rolling_sectionnode.inp	External file referenced by the adaptive mesh analysis.
ale_rolling_sectionelem.inp	External file referenced by the adaptive mesh analysis.
ale_rolling_sectionnelset.inp	External file referenced by the adaptive mesh analysis.
ale_rolling_sectionsurf.inp	External file referenced by the adaptive mesh analysis.
lag_rolling_section.inp	Lagrangian analysis using contact pairs.
lag_rolling_section_gcont.inp	Lagrangian analysis using general contact.

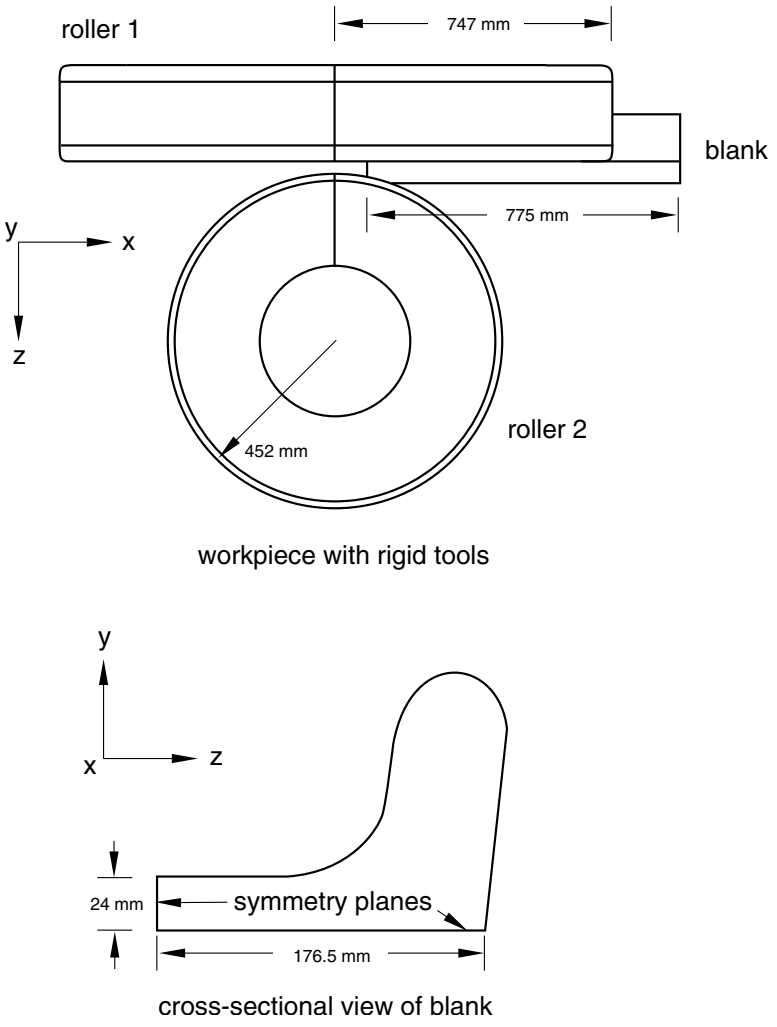


Figure 1.3.12–1 Geometry of the quarter-symmetry blank and the rollers.

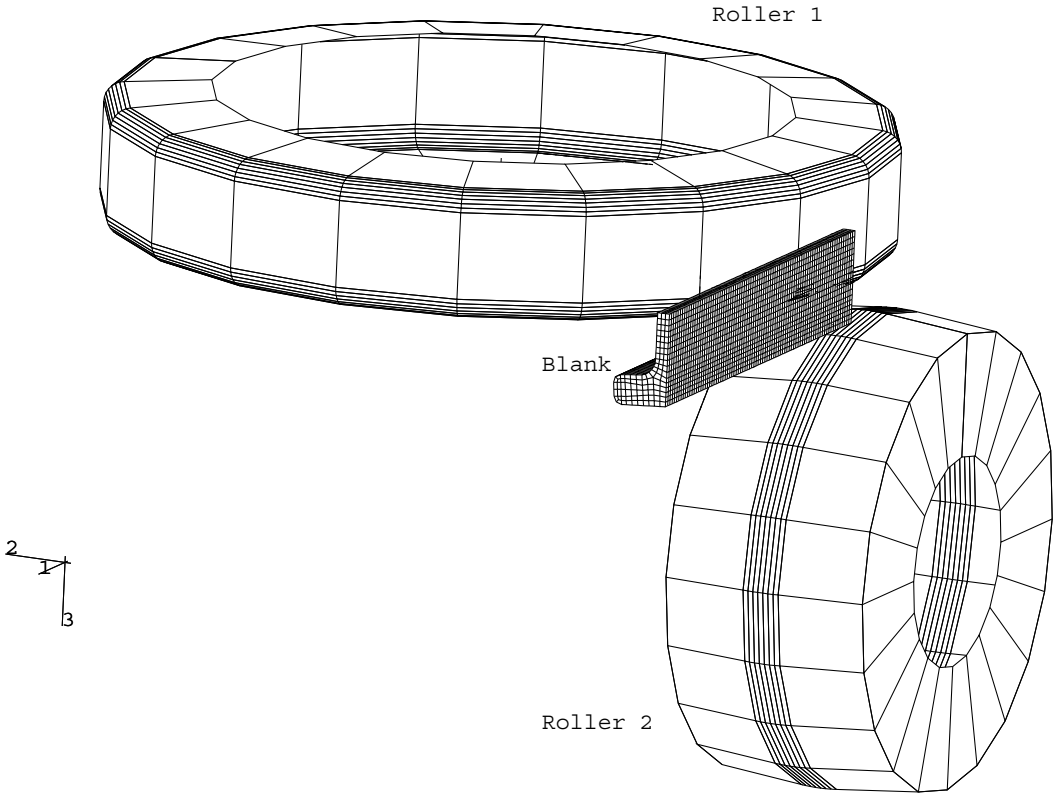


Figure 1.3.12–2 Quarter-symmetry finite element model.

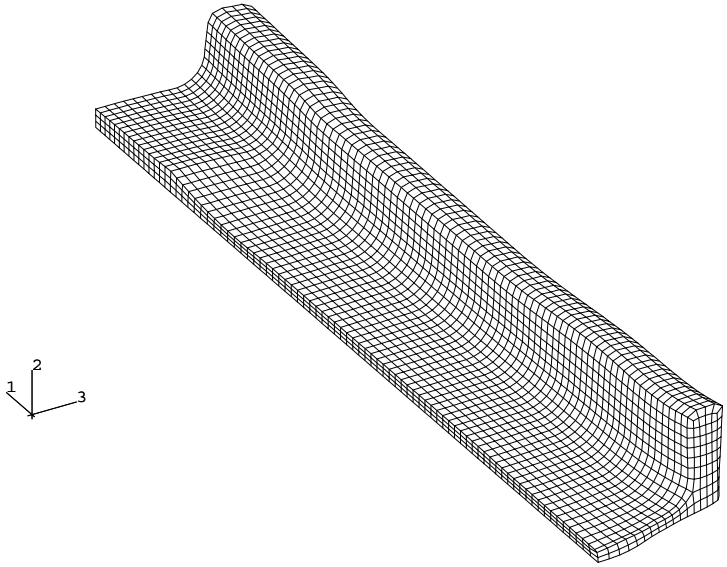


Figure 1.3.12–3 Deformed blank for the adaptive mesh simulation.

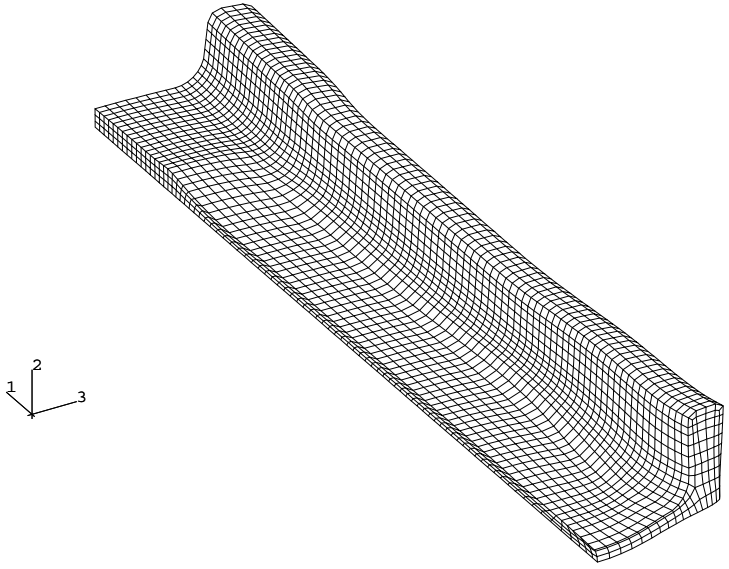


Figure 1.3.12–4 Deformed blank for the pure Lagrangian simulation.

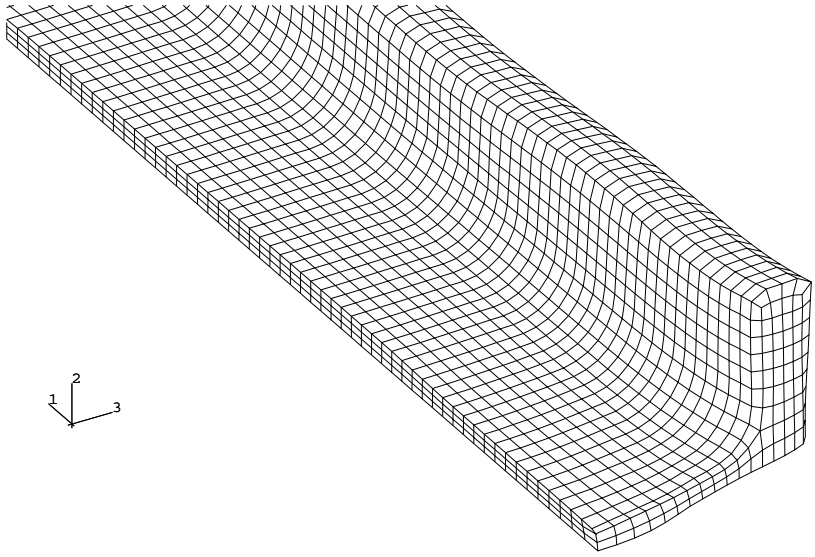


Figure 1.3.12–5 Close-up of the deformed blank for the adaptive mesh simulation.

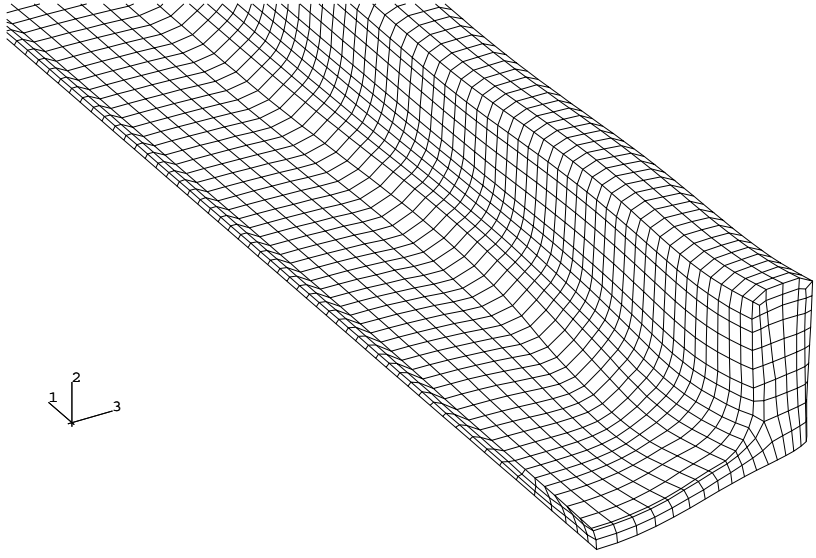


Figure 1.3.12–6 Close-up of the deformed blank for the pure Lagrangian simulation.

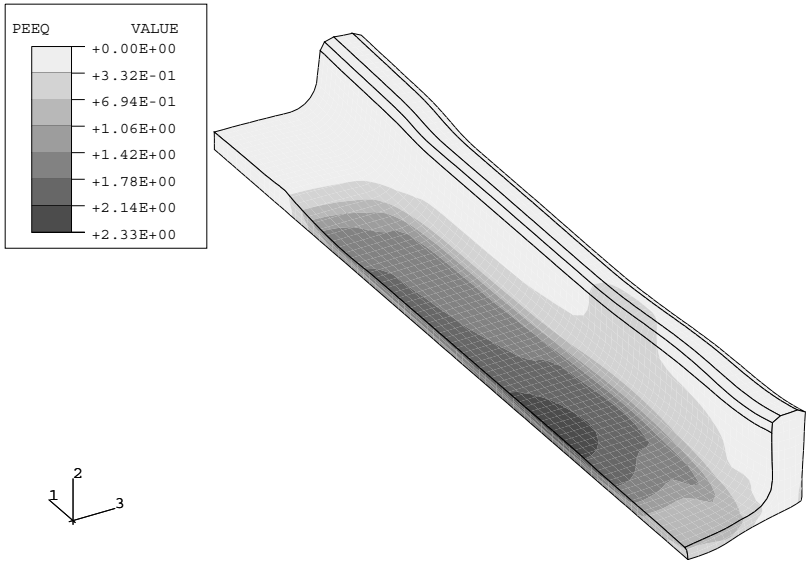


Figure 1.3.12–7 Contours of equivalent plastic strain for the adaptive mesh simulation.

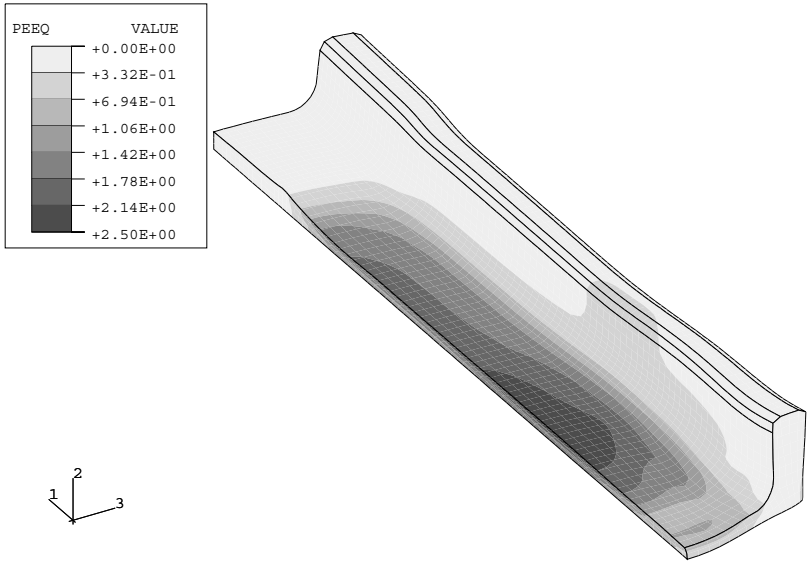


Figure 1.3.12–8 Contours of equivalent plastic strain for the pure Lagrangian simulation.

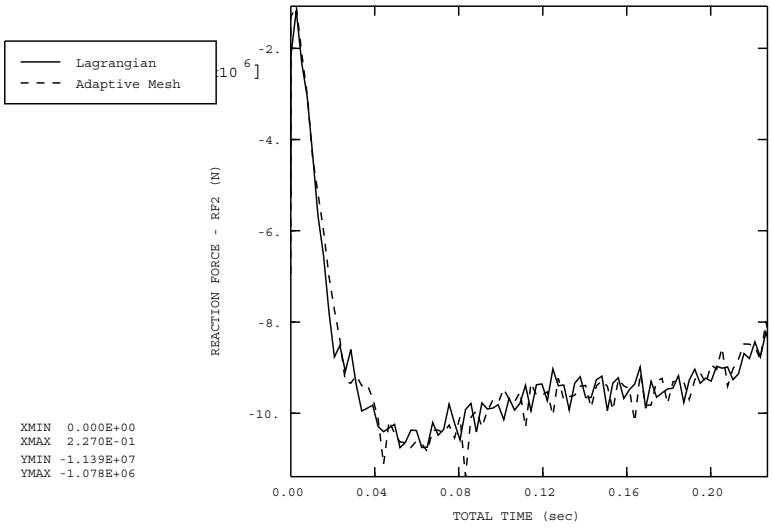


Figure 1.3.12–9 Time history of the reaction force in the *y*-direction at the reference node of Roller 1.

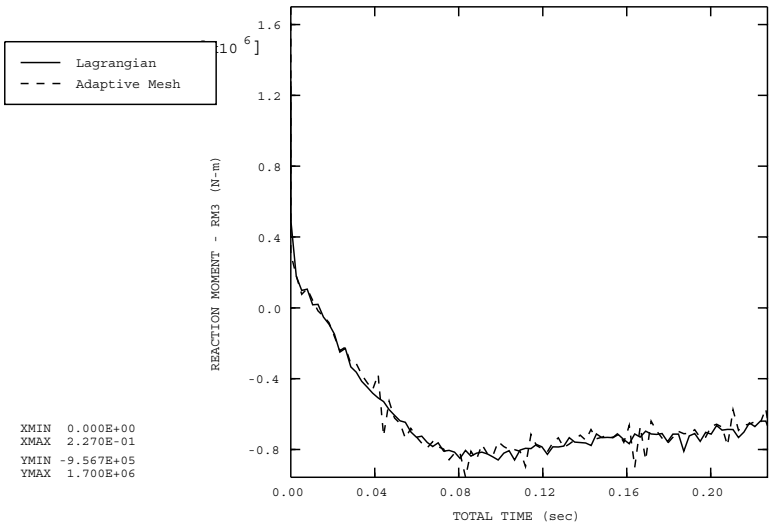


Figure 1.3.12–10 Time history of the reaction moment about the *z*-axis at the reference node of Roller 1.

1.3.13 RING ROLLING

Product: Abaqus/Explicit

This example illustrates the use of adaptive meshing in a two-dimensional rolling simulation. Results are compared to those obtained using a pure Lagrangian approach.

Problem description

Ring rolling is a specialized process typically used to manufacture parts with revolved geometries such as bearings. The three-dimensional rolling setup usually includes a freely mounted, idle roll; a continuously rotating driver roll; and guide rolls in the rolling plane. Transverse to the rolling plane, conical rolls are used to stabilize the ring and provide a forming surface in the out-of-plane direction. In this example a two-dimensional, plane stress idealization is used that ignores the effect of the conical rolls. A schematic diagram of the ring and the surrounding tools is shown in Figure 1.3.13–1.

The driver roll has a diameter of 680 mm, and the idle and guide rolls have diameters of 102 mm. The ring has an initial inner diameter of 127.5 mm and a thickness of 178.5 mm. The idle and driver rolls are arranged vertically and are in contact with the inner and outer surfaces of the ring, respectively. The driver roll is rotated around its stationary axis, while the idle roll is moved vertically downward at a specified feed rate. For this simulation the x–y motion of the guide rolls is determined *a priori* and is prescribed so that the rolls remain in contact with the ring throughout the analysis but do not exert appreciable force on it. In practice the guide rolls are usually connected through linkage systems, and their motion is a function of both force and displacement.

The ring is meshed with CPS4R elements, as shown in Figure 1.3.13–2. The ring is steel and is modeled as a von Mises elastic-plastic material with a Young's modulus of 150 GPa, an initial yield stress of 168.7 MPa, and a constant hardening slope of 884 MPa. The Poisson's ratio is 0.3; the density is 7800 kg/m^3.

The analysis is run so that the ring completes approximately 20 revolutions (16.5 seconds). The rigid rolls are modeled as analytical rigid surfaces using connected line segments. The driver roll is rotated at a constant angular velocity of 3.7888 rad/sec about the z-axis, while the idle roll has a constant feed rate of 4.9334 mm/sec and is free to rotate about the z-axis. All other degrees of freedom for the driver and idle rolls are constrained. A friction coefficient of 0.5 is defined at the blank-idle roll and blank-drive roll interfaces. Frictionless contact is used between the ring and guide rolls, and the rotation of the guide rolls is constrained since the actual guide rolls are free to rotate and exert negligible torque on the ring.

To obtain an economical solution, the masses of all elements in the ring are scaled by a factor of 2500. This scaling factor represents a reasonable upper limit on the mass scaling possible for this problem, above which significant inertial effects would be generated. Furthermore, since the two-dimensional model does not contain the conical rolls, the ring oscillates from side to side even under the action of the guide rolls. An artificial viscous pressure of 300 MPa sec/m is applied on the inner and outer surfaces of the ring to assist the guide rolls in preserving the circular shape of the ring. The pressure value was chosen by trial and error.

Adaptive meshing

A single adaptive mesh domain that incorporates the ring is defined. Contact surfaces on the ring are defined as sliding boundary regions (the default). Because of the large number of increments required to simulate 20 revolutions, the deformation per increment is very small. Therefore, the frequency of adaptive meshing is changed from the default of 10 to every 50 increments. The cost of adaptive meshing at this frequency is negligible compared to the underlying analysis cost.

Results and discussion

Figure 1.3.13–3 shows the deformed configuration of the ring after completing 20 revolutions with continuous adaptive meshing. High-quality element shapes and aspect ratios are maintained throughout the simulation. Figure 1.3.13–4 shows the deformed configuration of the ring when a pure Lagrangian simulation is performed. The pure Lagrangian mesh is distorted, especially at the inner radius where elements become skewed and very small in the radial direction.

Figure 1.3.13–5 and Figure 1.3.13–6 show time history plots for the *y*-component of reaction force on the idle roll and the reaction moment about the *z*-axis for the driver roll, respectively, for both the adaptive mesh and pure Lagrangian approaches. Although the final meshes are substantially different, the roll force and torque match reasonably well.

For both the adaptive and pure Lagrangian solutions the plane stress idealization used here results in very localized through-thickness straining at the inner and outer radii of the ring. This specific type of localized straining is unique to plane stress modeling and does not occur in ring rolling processes. It is also not predicted by a three-dimensional finite element model. If adaptivity is used and refined meshing is desired to capture strong gradients at the inner and outer extremities, the initially uniform mesh can be replaced with a graded mesh. Although not shown here, a graded mesh concentrates element refinement in areas of strong gradients. You can specify in the adaptive mesh controls that the initial mesh gradation should be preserved while distortions are reduced as the analysis evolves.

Input files

ale_ringroll_2d.inp	Analysis that uses adaptive meshing.
ale_ringroll_2dnode.inp	External file referenced by the adaptive mesh analysis.
ale_ringroll_2delem.inp	External file referenced by the adaptive mesh analysis.
guideamp.inp	External file referenced by the adaptive mesh analysis.
lag_ringroll_2d.inp	Lagrangian analysis.

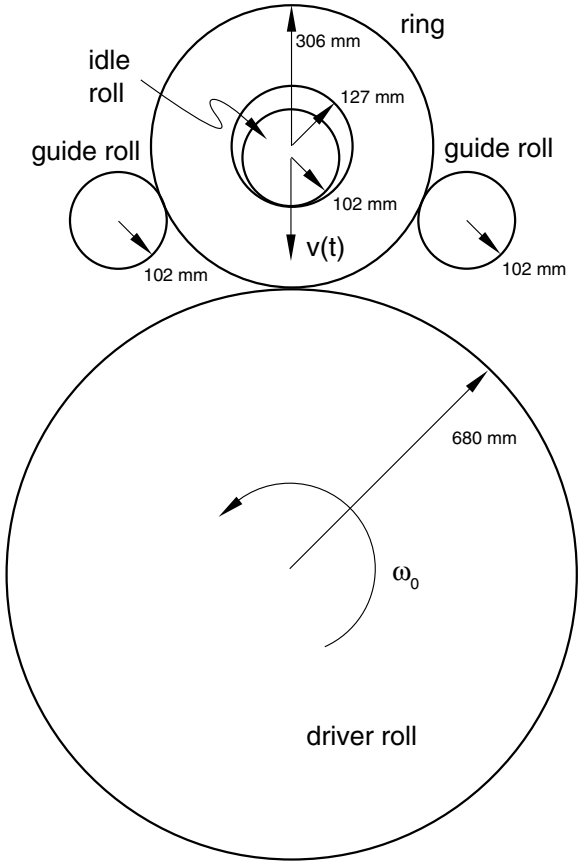


Figure 1.3.13–1 Model geometry for the two-dimensional ring rolling analysis.

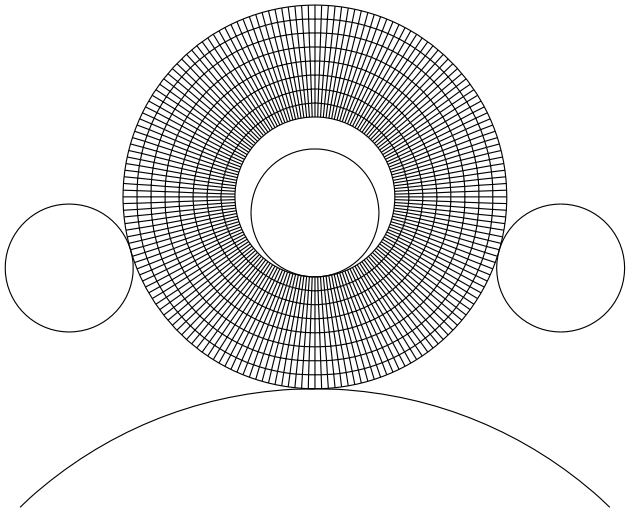


Figure 1.3.13–2 Initial mesh configuration.

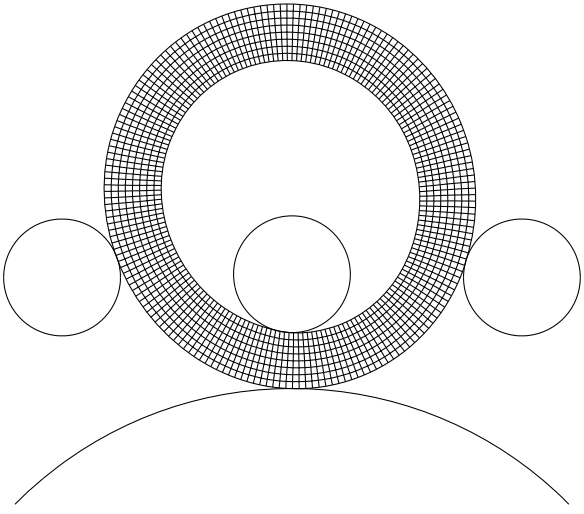


Figure 1.3.13–3 Deformed configuration after 20 revolutions using adaptive meshing.

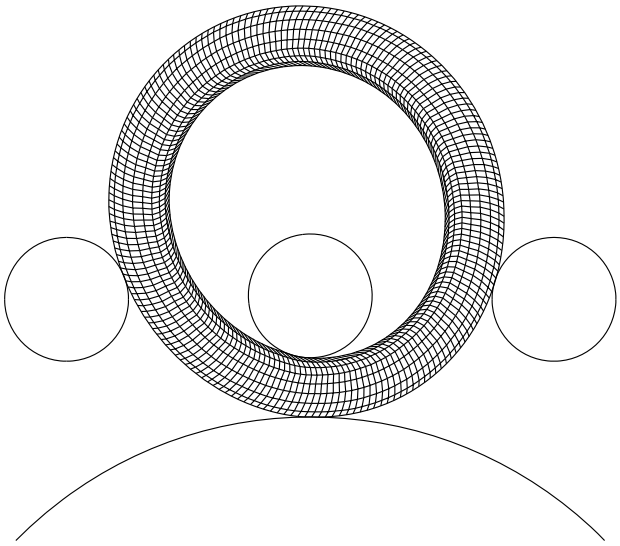


Figure 1.3.13–4 Deformed configuration after 20 revolutions using a pure Lagrangian approach.

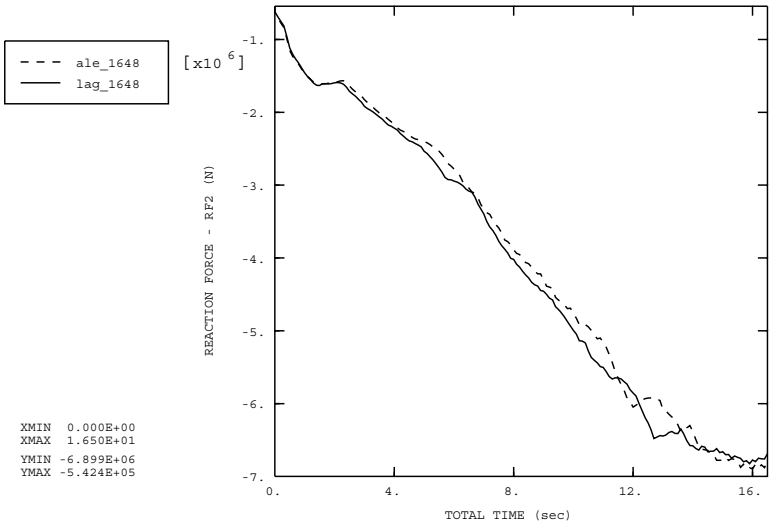


Figure 1.3.13–5 Time history of the reaction force in the *y*-direction for the idle roll.

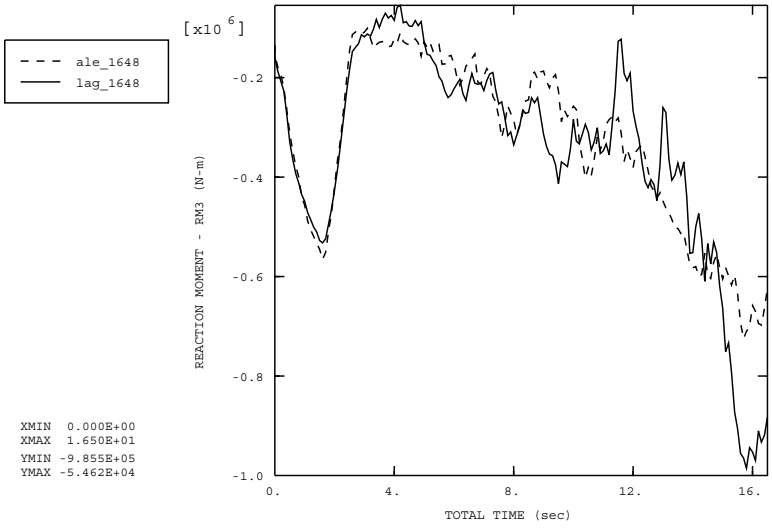


Figure 1.3.13–6 Time history of the reaction moment about the *z*-axis for the driver roll.

1.3.14 AXISYMMETRIC EXTRUSION: TRANSIENT AND STEADY-STATE

Product: Abaqus/Explicit

This example illustrates the use of adaptive meshing in simulations of extrusion processes with three axisymmetric analysis cases. First, a transient simulation is performed for a backward, flat-nosed die, extrusion geometry using adaptivity on a Lagrangian mesh domain. Second, a transient simulation is performed on the analogous forward, square die, extrusion geometry, also using adaptivity on a Lagrangian mesh domain. Finally, a steady-state simulation is performed for the forward extrusion geometry using adaptivity on an Eulerian mesh domain.

Problem description

The model configurations for the three analysis cases are shown in Figure 1.3.14–1. Each of the models is axisymmetric and consists of one or more rigid tools and a deformable blank. The rigid tools are modeled as analytical rigid surfaces of connected line segments. All contact surfaces are assumed to be well-lubricated and, thus, are treated as frictionless. The blank is made of aluminum and is modeled as a von Mises elastic-plastic material with isotropic hardening. The Young's modulus is 38 GPa, and the initial yield stress is 27 MPa. The Poisson's ratio is 0.33; the density is 2672 kg/m^3.

Case 1: Transient analysis of a backward extrusion

The model geometry consists of a rigid die, a rigid punch, and a blank. The blank is meshed with CAX4R elements and measures 28 × 89 mm. The blank is constrained along its base in the z-direction and at the axis of symmetry in the r-direction. Radial expansion is prevented by contact between the blank and the die. The punch and the die are fully constrained, with the exception of the prescribed vertical motion of the punch. The punch is moved downward 82 mm to form a tube with wall and endcap thicknesses of 7 mm each. The punch velocity is specified using a smooth amplitude so that the response is essentially quasi-static.

The deformation that occurs in extrusion problems, especially in those that involve flat-nosed die geometries, is extreme and requires adaptive meshing. Since adaptive meshing in Abaqus/Explicit works with the same mesh topology throughout the step, the initial mesh must be chosen such that the mesh topology will be suitable for the duration of the simulation. A simple meshing technique has been developed for extrusion problems such as this. In two dimensions it uses a four-sided, mapped mesh domain that can be created with nearly all finite element mesh preprocessors. The vertices for the four-sided, mapped mesh are shown in Figure 1.3.14–1 and are denoted A, B, C, and D. Two vertices are located on either side of the extrusion opening, the third is in the corner of the dead material zone (the upper left corner of the blank), and the fourth vertex is located in the diagonally opposite corner. A 10 × 60 element mesh using this meshing technique is created for this analysis case and is shown in Figure 1.3.14–2. The mesh refinement is oriented such that the fine mesh along sides AB and DC will move up along the extruded walls as the punch is moved downward.

An adaptive mesh domain is defined that incorporates the entire blank. Because of the extremely large distortions expected in the backward extrusion simulation, three mesh sweeps, instead of the default

value of one, are specified for each adaptive mesh increment. The default adaptive meshing frequency of 10 is used. Alternatively, a higher frequency could be specified to perform one mesh sweep per adaptive mesh increment. However, this method would result in a higher computational cost because of the increased number of advection sweeps it would require.

A substantial amount of initial mesh smoothing is performed by increasing the number of mesh sweeps to be performed at the beginning of the step to 100. The initially smoothed mesh is shown in Figure 1.3.14–2. Initial smoothing reduces the distortion of the mapped mesh by rounding out corners and easing sharp transitions before the analysis is performed; therefore, it allows the best mesh to be used throughout the analysis.

Case 2: Transient analysis of a forward extrusion

The model geometry consists of a rigid die and a blank. The blank geometry and the mesh are identical to those described for Case 1, except that the mapped mesh is reversed with respect to the vertical plane so that the mesh lines are oriented toward the forward extrusion opening. The blank is constrained at the axis of symmetry in the *r*-direction. Radial expansion is prevented by contact between the blank and the die. The die is fully constrained. The blank is pushed up 19 mm by prescribing a constant velocity of 5 m/sec for the nodes along the bottom of the blank. As the blank is pushed up, material flows through the die opening to form a solid rod with a 7 mm radius.

Adaptive meshing for Case 2 is defined in a similar manner as for Case 1. The undeformed mesh configurations, before and after initial mesh smoothing, are shown in Figure 1.3.14–3.

Case 3: Steady-state analysis of a forward extrusion

The model geometry consists of a rigid die, identical to the die used for Case 2, and a blank. The blank geometry is defined such that it closely approximates the shape corresponding to the steady-state solution: this geometry can be thought of as an "initial guess" to the solution. As shown in Figure 1.3.14–4, the blank is discretized with a simple graded pattern that is most refined near the die fillet. No special mesh is required for the steady-state case since minimal mesh motion is expected during the simulation. The blank is constrained at the axis of symmetry in the *r*-direction. Radial expansion of the blank is prevented by contact between it and the die.

An adaptive mesh domain is defined that incorporates the entire blank. Because the Eulerian domain undergoes very little overall deformation and the material flow speed is much less than the material wave speed, the frequency of adaptive meshing is changed to 5 from the default value of 1 to improve the computational efficiency of the analysis.

The outflow boundary is assumed to be traction-free and is located far enough downstream to ensure that a steady-state solution can be obtained. This boundary is cast as an Eulerian boundary region. A multi-point constraint is defined on the outflow boundary to keep the velocity normal to the boundary uniform. The inflow boundary is defined using an Eulerian boundary condition to prescribe a velocity of 5 m/sec in the vertical direction. Adaptive mesh constraints are defined on both the inflow and outflow boundaries to fix the mesh in the vertical direction. This effectively creates a stationary control volume with respect to the inflow and outflow boundaries through which material can pass.

Results and discussion

The results for each analysis case are described below.

Case 1

The use of the mapped meshing technique along with adaptive meshing allows the backward extrusion analysis to run to completion, creating the long tube with an endcap. Three plots of the deformed mesh at various times are shown in Figure 1.3.14–5. These plots clearly show how the quality of the mesh is preserved for the majority of the simulation. Despite the large amount of deformation involved, the mesh remains smooth and concentrated in the areas of high strain gradients. Extreme deformation and thinning at the punch fillet occurs near the end of the analysis. This thinning can be reduced by increasing the fillet radius of the punch. Corresponding contours of equivalent plastic strain are plotted in Figure 1.3.14–6. The plastic strains are highest along the inner surface of the tube.

Case 2

Adaptive meshing enables the transient forward extrusion simulation to proceed much further than would be possible using a pure Lagrangian approach. After pushing the billet 19 mm through the die, the analysis cannot be continued because the elements become too distorted. Since the billet material is essentially incompressible and the cross-sectional area of the die opening at the top is 1/16 of the original cross-sectional area of the billet, a rod measuring approximately 304 mm (three times the length of the original billet) is formed.

Three plots of the deformed mesh at various times in the transient forward extrusion are shown in Figure 1.3.14–7. As in the backward extrusion case, the plots show that the quality of the mesh is preserved for a majority of the simulation. The last deformed shape has been truncated for clarity because the extruded column becomes very long and thin. Contours of equivalent plastic strain at similar times are shown in Figure 1.3.14–8. The plastic strain distribution developing in the vertical column does not reach a steady-state value, even at a height of 304 mm. The steady-state results reported in the discussion for Case 3 show that a steady-state solution based on the equivalent plastic strain distribution is not reached until much later. An absolute steady-state solution cannot be reached until the material on the upstream side of the dead material zone first passes along that zone and through the die opening. The dead material zone is roughly the shape of a triangle and is located in the upper right-hand corner of the die.

Case 3

The steady-state solution to the forward extrusion analysis is obtained at an extruded column height of 800 mm, which corresponds to pushing the billet 50 mm through the die. Thus, this analysis runs 2.5 times longer than Case 2.

Contours of equivalent plastic strain in the middle and at the end of the simulation are shown in Figure 1.3.14–9. Time histories of the equivalent plastic strains on the outer edge of the extruded column at the outflow boundary and 27.5 mm below the outflow boundary are shown in Figure 1.3.14–10. The plastic strains at both locations converge to the same value by the end of the simulation, which indicates

that the solution has reached a steady state. The final mesh configuration is shown in Figure 1.3.14–11. The mesh undergoes very little change from the beginning to the end of the analysis because of the accurate initial guess made for the steady-state domain shape and the ability of the adaptive meshing capability in Abaqus/Explicit to retain the original mesh gradation.

As a further check on the accuracy of the steady-state simulation and the conservation properties of adaptive meshing, a time history of the velocity at the outflow boundary is shown in Figure 1.3.14–12. The velocity reaches a steady value of approximately 80 m/s, which is consistent with the incompressible material assumption and the 1/16 ratio of the die opening to the billet size.

Input files

ale_extrusion_back.inp	Case 1.
ale_extrusion_backnode.inp	Node data for Case 1.
ale_extrusion_backelem.inp	Element data for Case 1.
ale_extrusion_forward.inp	Case 2.
ale_extrusion_forwardnode.inp	Node data for Case 2.
ale_extrusion_forwardelem.inp	Element data for Case 2.
ale_extrusion_eulerian.inp	Case 3.
ale_extrusion_euleriannode.inp	Node data for Case 3.
ale_extrusion_eulerianelem.inp	Element data for Case 3.

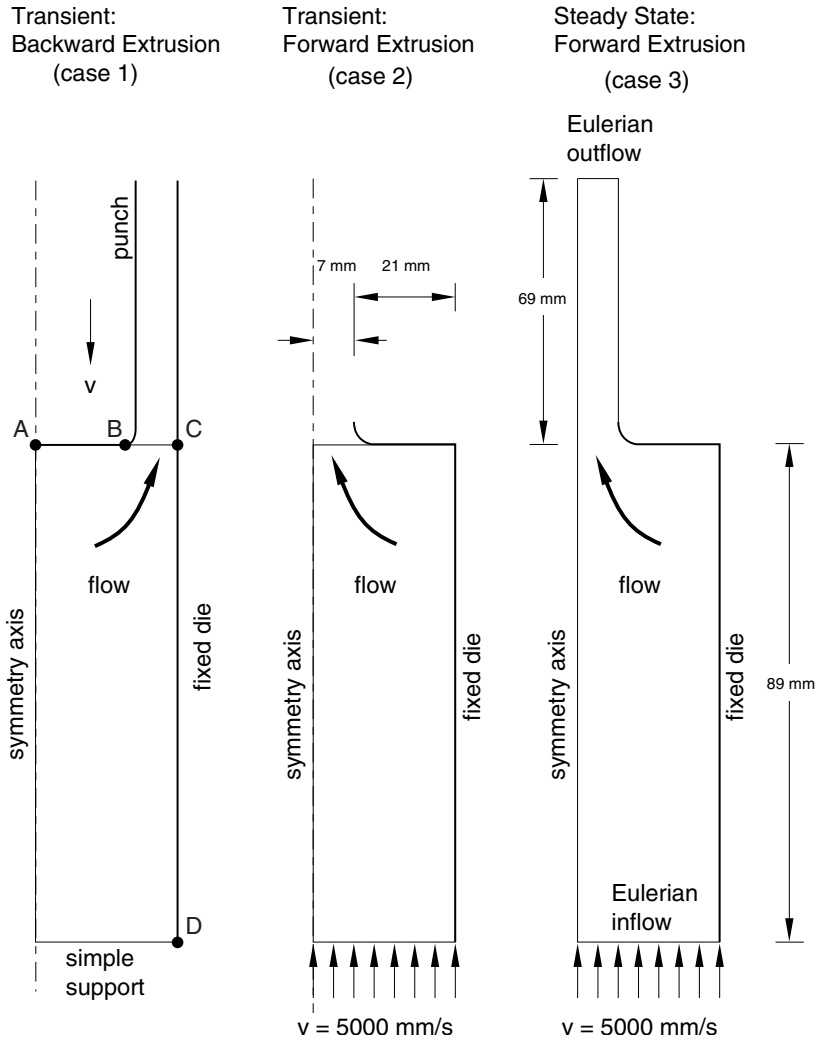


Figure 1.3.14–1 Axisymmetric model geometries used in the extrusion analysis.

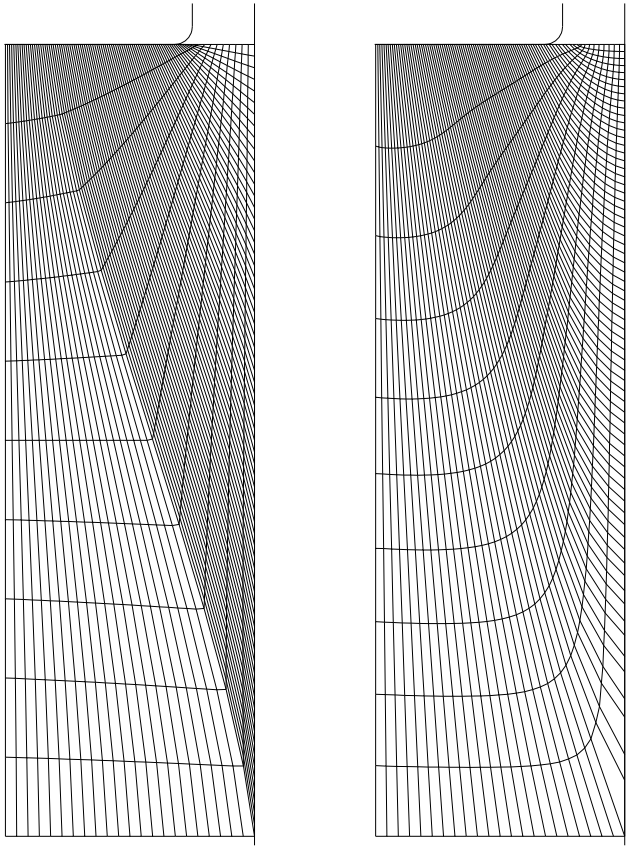


Figure 1.3.14–2 Undeformed configuration for Case 1, before and after initial smoothing.

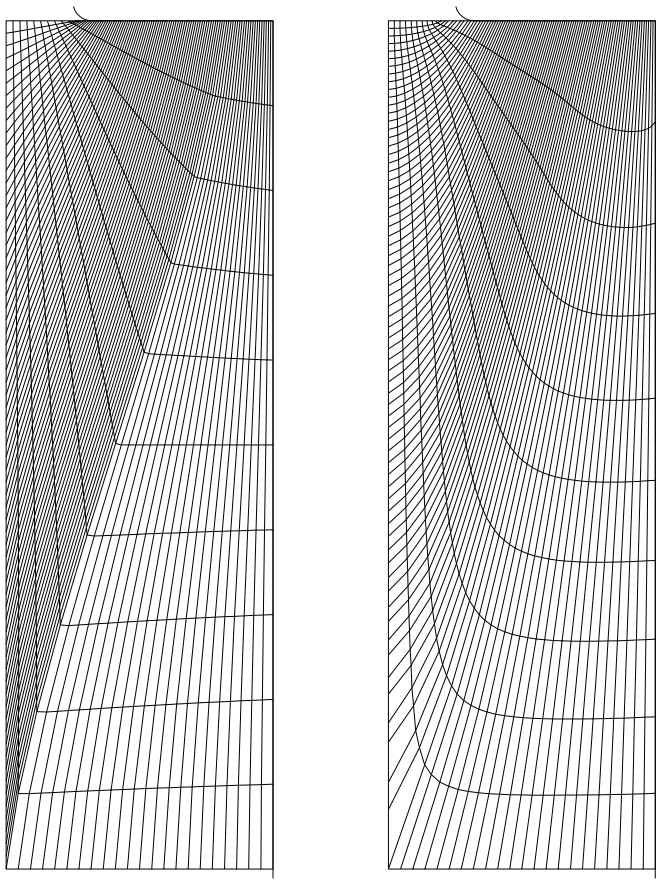


Figure 1.3.14–3 Undeformed configuration for Case 2, before and after initial smoothing.

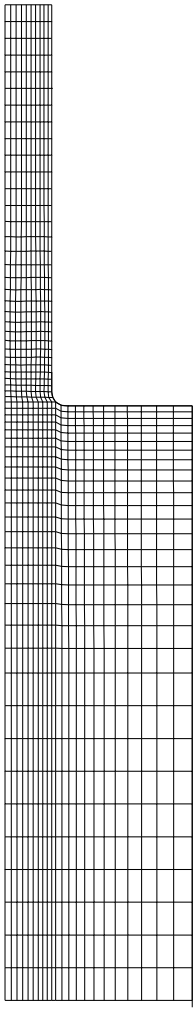


Figure 1.3.14–4 Undeformed configuration for Case 3.

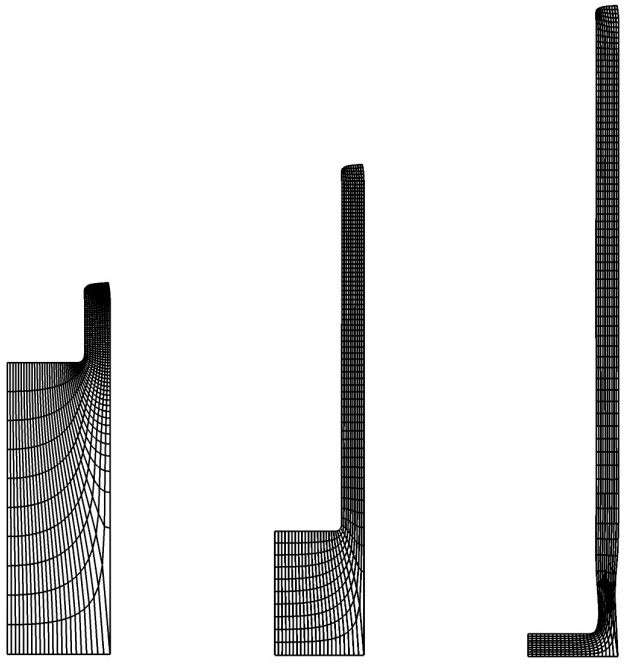


Figure 1.3.14–5 Deformed mesh at various times for Case 1.

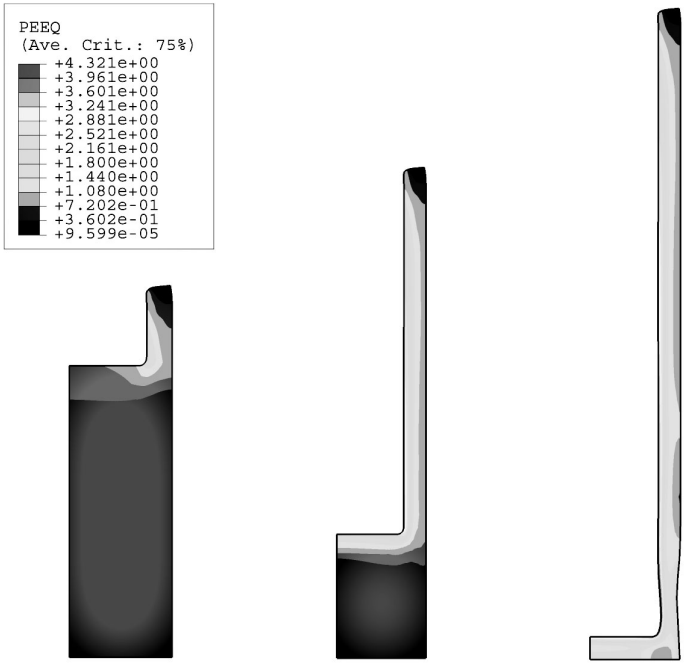


Figure 1.3.14–6 Contours of equivalent plastic strain at various times for Case 1.

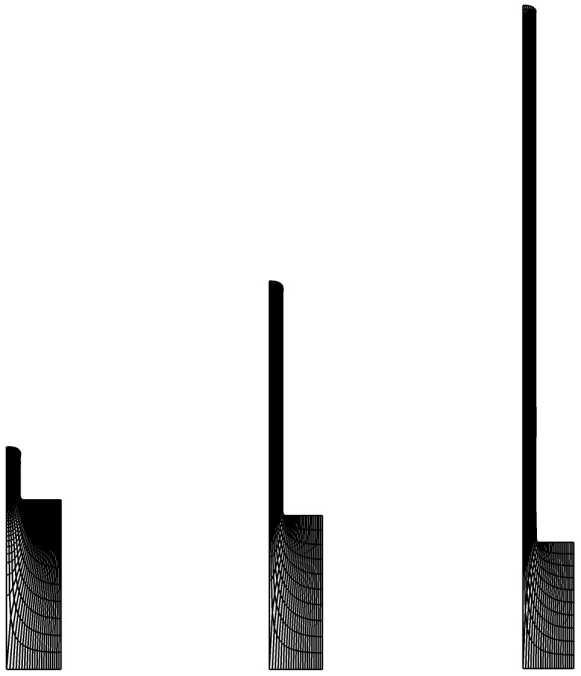


Figure 1.3.14–7 Deformed mesh at various times for Case 2.

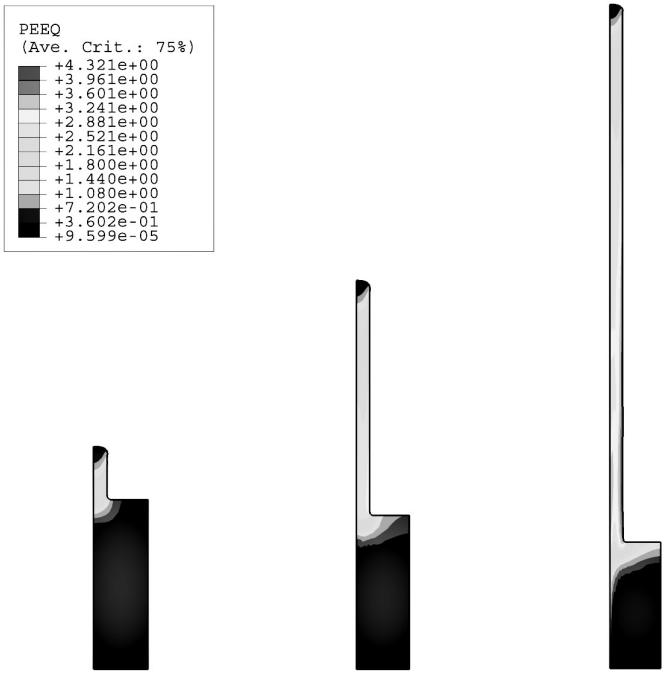


Figure 1.3.14–8 Contours of equivalent plastic strain at various times for Case 2.

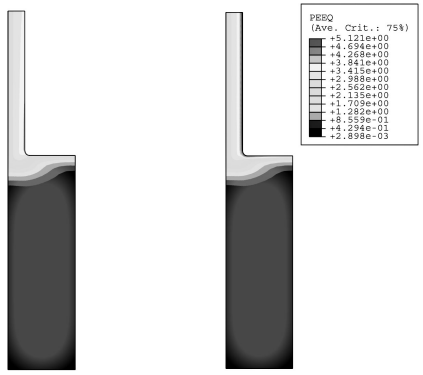


Figure 1.3.14–9 Contours of equivalent plastic strain at an intermediate stage and at the end of the analysis for Case 3.

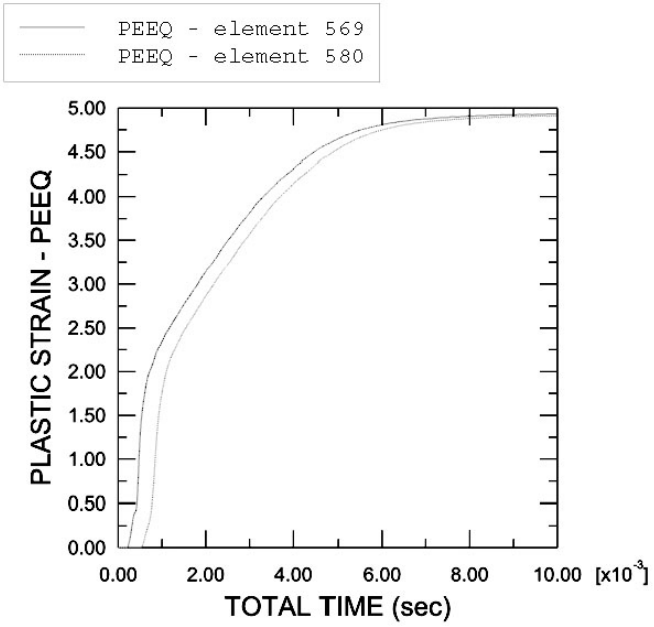


Figure 1.3.14–10 Time history of equivalent plastic strain along the outer edge of the extruded column for Case 3.

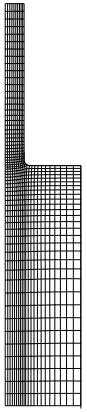


Figure 1.3.14–11 Final deformed mesh for Case 3.

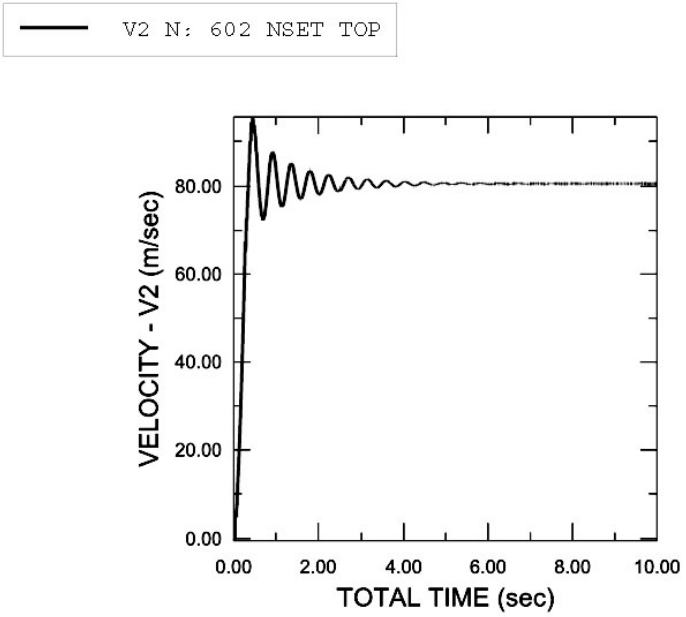


Figure 1.3.14–12 Time history of material velocity at the outflow boundary for Case 3.

1.3.15 TWO-STEP FORMING SIMULATION

Product: Abaqus/Explicit

This example illustrates the use of adaptive meshing in simulations of a two-step, bulk metal forming process. The problem is based on a benchmark problem presented at the Metal Forming Process Simulation in Industry conference.

Problem description

The model consists of two sets of rigid forming tools (one set for each forming step) and a deformable blank. The blank and forming die geometries used in the simulation are shown in Figure 1.3.15–1. The initial configurations of the blank and the tools for each step are shown in Figure 1.3.15–2 and Figure 1.3.15–4. All forming tools are modeled as discrete rigid bodies and meshed with R3D4 and R3D3 elements. The blank, which is meshed with C3D8R elements, is cylindrical and measures 14.5 × 21 mm. A half model is constructed, so symmetry boundary conditions are prescribed at the y=0 plane.

The blank is made of a steel alloy that is assumed to satisfy the Ramberg-Osgood relation for true stress and logarithmic strain,

$$\epsilon = (\sigma/K)^{1/n},$$

with a reference stress value (K) of 763 MPa and a work-hardening exponent (n) of 0.245. Isotropic elasticity is assumed, with a Young's modulus of 211 GPa and a Poisson's ratio of 0.3. An initial yield stress of 200 MPa is obtained with these data. The stress-strain behavior is defined by piecewise linear segments matching the Ramberg-Osgood curve up to a total (logarithmic) strain level of 140%, with von Mises yield and isotropic hardening.

The analysis is conducted in two steps. For the first step the rigid tools consist of a planar punch, a planar base, and a forming die. The initial configuration for this step is shown in Figure 1.3.15–2. The base, which is not shown, is placed at the opening of the forming die to prevent material from passing through the die. The motion of the tools is fully constrained, with the exception of the prescribed displacement in the z-direction for the punch, which is moved 12.69 mm toward the blank at a constant velocity of 30 m/sec consistent with a quasi-static response. The deformed configuration of the blank at the completion of the first step is shown in Figure 1.3.15–3.

In the second step the original punch and die are removed from the model and replaced with a new punch and die, as shown in Figure 1.3.15–4. The removal of the tools is accomplished by deleting the contact pairs between them and the blank. Although not shown in the figure, the base is retained; both it and the new die are fully constrained. The punch is moved 10.5 mm toward the blank at a constant velocity of 30 m/sec consistent with a quasi-static response. The deformed configuration of the blank at the completion of the second step is shown in Figure 1.3.15–5.

Adaptive meshing

A single adaptive mesh domain that incorporates the entire blank is used for both steps. A Lagrangian boundary region type (the default) is used to define the constraints on the symmetry plane, and a sliding boundary region type (the default) is used to define all contact surfaces. The frequency of adaptive meshing is increased to 5 for this problem since material flows quickly near the end of the step.

Results and discussion

Figure 1.3.15–6 shows the deformed mesh at the completion of forming for an analysis in which a pure Lagrangian mesh is used. Comparing Figure 1.3.15–5 and Figure 1.3.15–6, the resultant mesh for the simulation in which adaptive meshing is used is clearly better than that obtained with a pure Lagrangian mesh.

In Figure 1.3.15–7 through Figure 1.3.15–9 path plots of equivalent plastic strain in the blank are shown using the pure Lagrangian and adaptive mesh domains for locations in the y=0 symmetry plane at an elevation of z=10 mm. The paths are defined in the positive x-direction (from left to right in Figure 1.3.15–4 to Figure 1.3.15–6). As shown in Figure 1.3.15–7, the results are in good agreement at the end of the first step. At the end of the second step the path is discontinuous. Two paths are considered: one that spans the left-hand side and another that spans the right-hand side of the U-shaped cross-section along the symmetry plane. The left- and right-hand paths are shown in Figure 1.3.15–8 and Figure 1.3.15–9, respectively. The solutions from the second step compare qualitatively. Small differences can be attributed to the increased mesh resolution and reduced mesh distortion for the adaptive mesh domain.

Input files

ale_forging_steelpart.inp	Analysis with adaptive meshing.
ale_forging_steelpartnode1.inp	External file referenced by the adaptive mesh analysis.
ale_forging_steelpartnode2.inp	External file referenced by the adaptive mesh analysis.
ale_forging_steelpartnode3.inp	External file referenced by the adaptive mesh analysis.
ale_forging_steelpartnode4.inp	External file referenced by the adaptive mesh analysis.
ale_forging_steelpartelem1.inp	External file referenced by the adaptive mesh analysis.
ale_forging_steelpartelem2.inp	External file referenced by the adaptive mesh analysis.
ale_forging_steelpartelem3.inp	External file referenced by the adaptive mesh analysis.
ale_forging_steelpartelem4.inp	External file referenced by the adaptive mesh analysis.
ale_forging_steelpartelem5.inp	External file referenced by the adaptive mesh analysis.
ale_forging_steelpartsets.inp	External file referenced by the adaptive mesh analysis.
lag_forging_steelpart.inp	Pure Lagrangian analysis.
lag_forging_steelpart_gcont.inp	Pure Lagrangian general contact analysis.

Reference

- Hermann, M., and A. Ruf, "Forming of a Steel Part," Metal Forming Process Simulation in Industry, Stuttgart, Germany, September 1994.

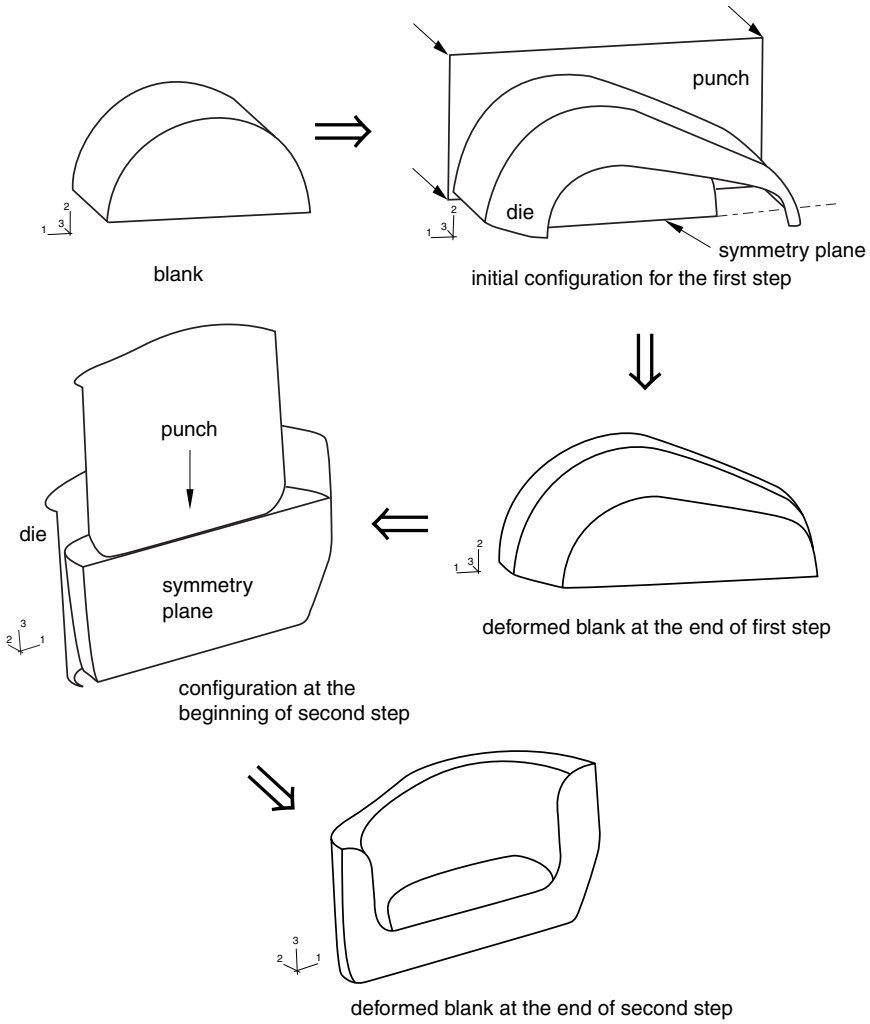


Figure 1.3.15–1 Two-step forging process.

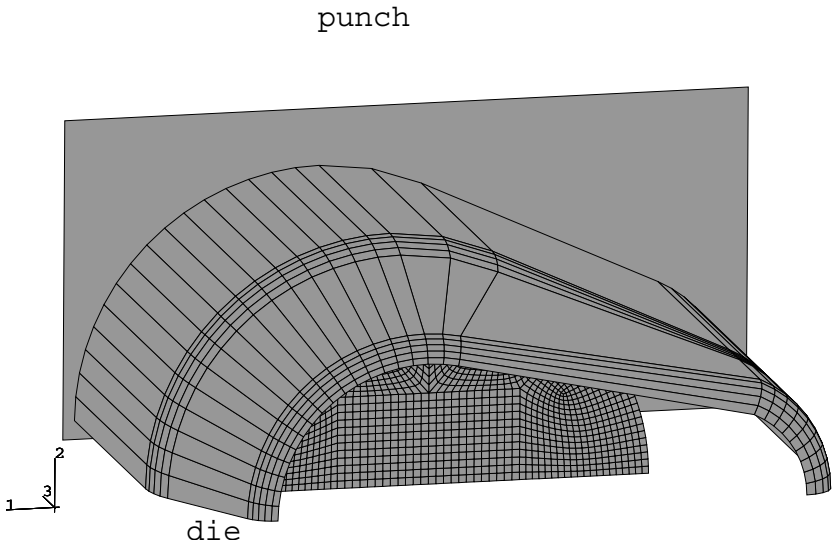


Figure 1.3.15–2 Initial configuration for the first step.

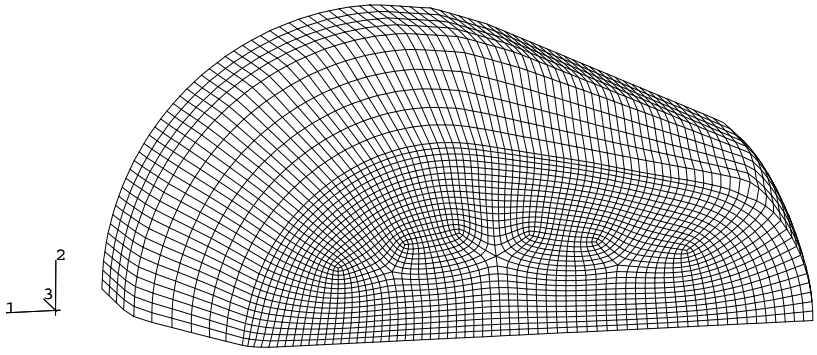


Figure 1.3.15–3 Deformed blank at the end of the first step.

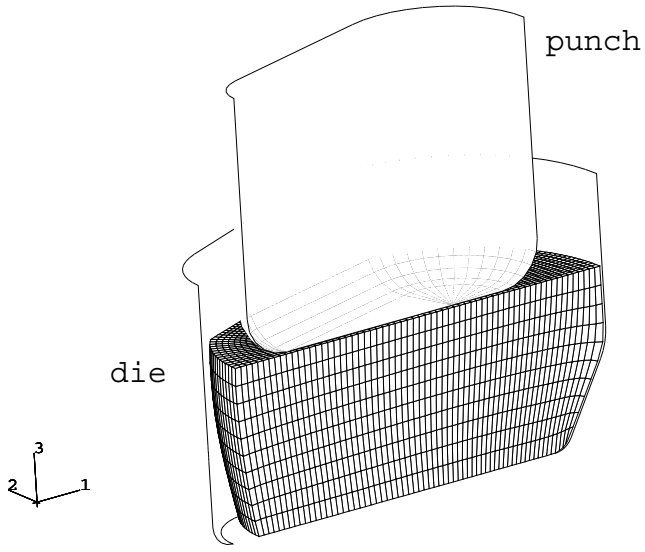


Figure 1.3.15–4 Configuration at the beginning of the second step.

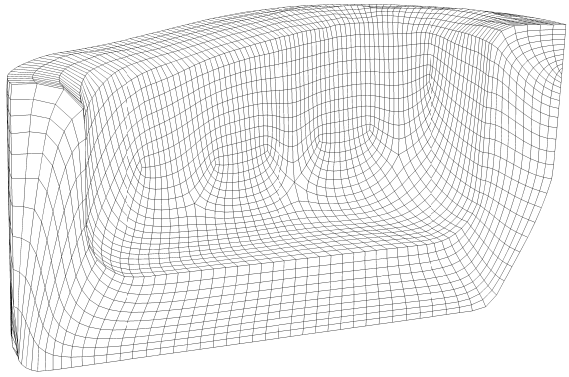


Figure 1.3.15–5 Deformed blank at the end of the second step for the adaptive mesh analysis.

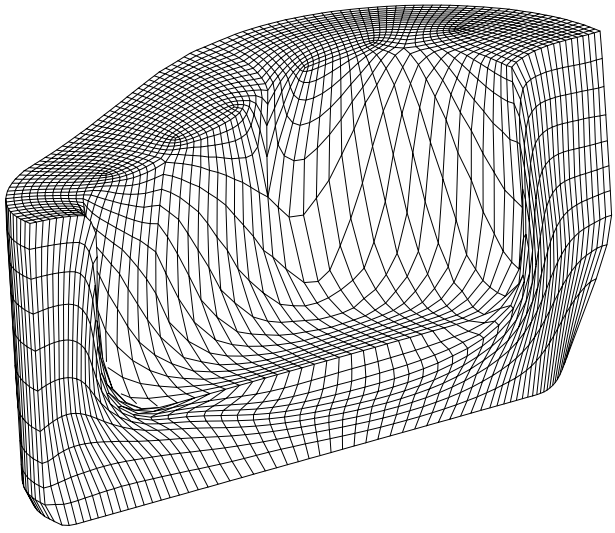


Figure 1.3.15–6 Deformed blank at the end of the second step for the pure Lagrangian analysis.

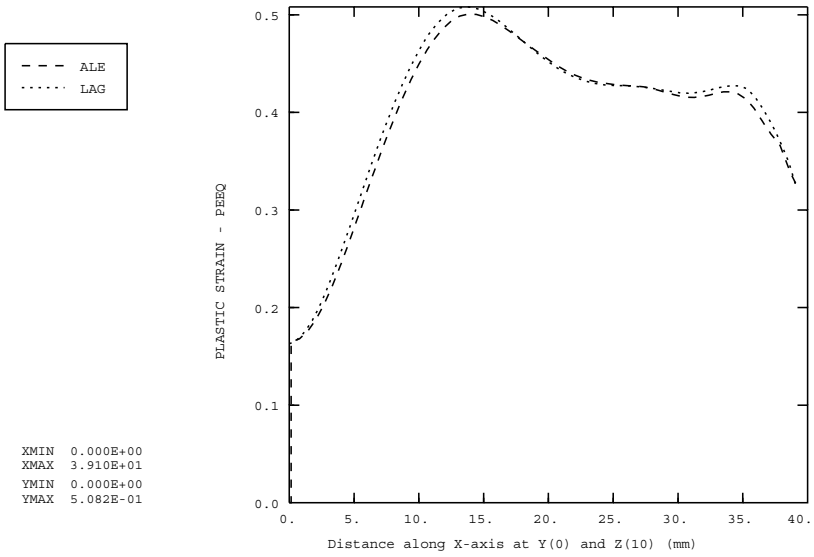


Figure 1.3.15–7 Path plot of equivalent plastic strain at the end of the first step.

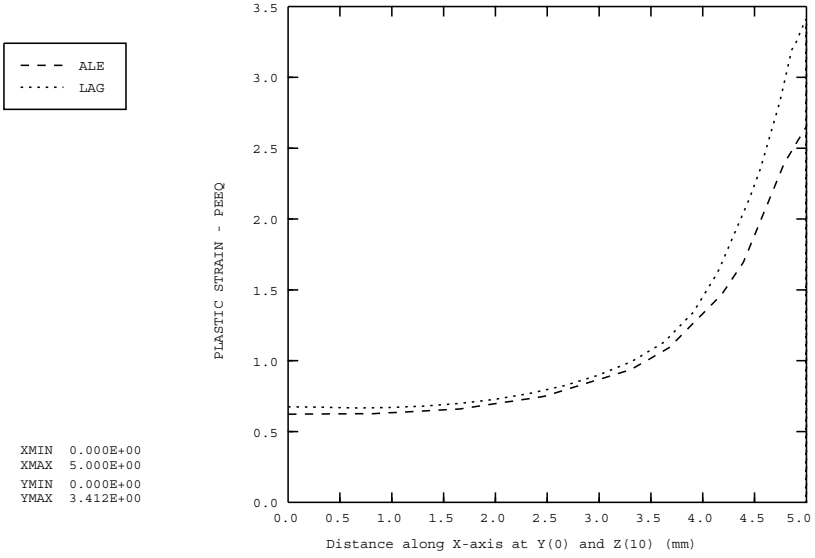


Figure 1.3.15–8 Path plot of equivalent plastic strain along the left side at the end of the second step.

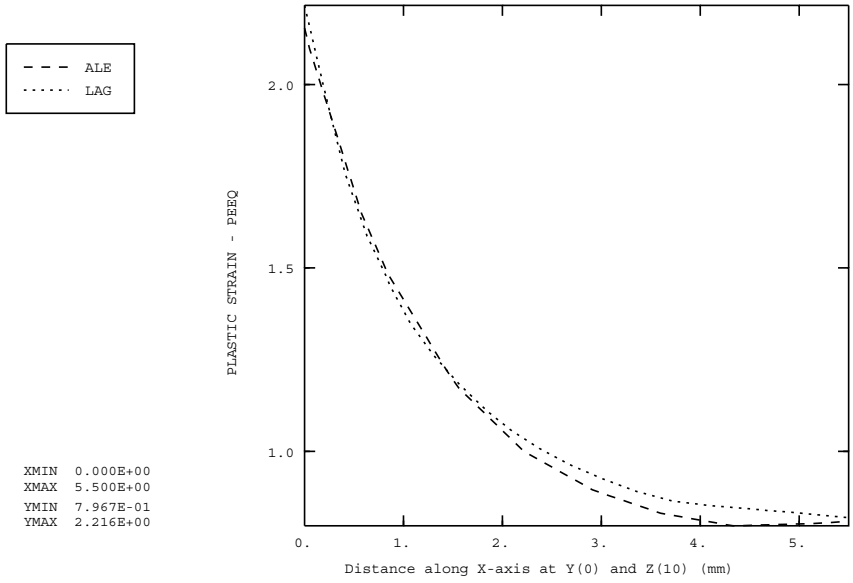


Figure 1.3.15–9 Path plot of equivalent plastic strain along the right side at the end of the second step.

1.3.16 UPSETTING OF A CYLINDRICAL BILLET: COUPLED TEMPERATURE-DISPLACEMENT AND ADIABATIC ANALYSIS

Products: Abaqus/Standard Abaqus/Explicit

This example illustrates coupled temperature-displacement analysis in a metal forming application. The case studied is an extension of the standard test case that is defined in Lippmann (1979); thus, some verification of the results is available by comparison with the numerical results presented in that reference. The example is that of a small, circular billet of metal that is reduced in length by 60%. Here the problem is analyzed as a viscoplastic case, including heating of the billet by plastic work. Such analysis is often important in manufacturing processes, especially when significant temperature rises degrade the material. The problem is also analyzed in Abaqus/Standard using a porous metal material model. The same problem is used in "Upsetting of a cylindrical billet: quasi-static analysis with mesh-to-mesh solution mapping (Abaqus/Standard) and adaptive meshing (Abaqus/Explicit)," Section 1.3.1, to illustrate mesh rezoning in Abaqus/Standard and adaptive meshing in Abaqus/Explicit.

Geometry and model

The specimen is shown in Figure 1.3.16–1: a circular billet, 30 mm long, with a radius of 10 mm, compressed between flat, rough, rigid dies. All surfaces of the billet are assumed to be fully insulated: this thermal boundary condition is chosen to maximize the temperature rise.

The finite element model is axisymmetric and includes the top half of the billet only since the middle surface of the billet is a plane of symmetry. In Abaqus/Standard elements of type CAX8RT, 8-node quadrilaterals with reduced integration that allow for fully coupled temperature-displacement analysis, are used. A regular mesh with six elements in each direction is used, as shown in Figure 1.3.16–1. In addition, the billet is modeled with CAX4RT elements in a 12 × 12 mesh for both Abaqus/Standard and Abaqus/Explicit analyses.

The contact between the top and the lateral exterior surfaces of the billet and the rigid die is modeled with a contact pair. The billet surface is defined by means of an element-based surface. The rigid die is modeled as an analytical rigid surface or as an element-based rigid surface. The mechanical interaction between the contact surfaces is assumed to be nonintermittent, rough frictional contact in Abaqus/Standard. Therefore, the contact property includes two additional specifications: a no-separation contact pressure-overclosure relationship to ensure that separation does not occur once contact has been established and rough friction to enforce a no-slip constraint once contact has been established. In Abaqus/Explicit the friction coefficient between the billet and the rigid die is 1.0.

The problem is also solved in Abaqus/Standard with the first-order fully coupled temperature-displacement CAX4T elements in a 12 × 12 mesh. Similarly, the problem is solved using CAX8RT elements and user subroutines **UMAT** and **UMATHT** to illustrate the use of these subroutines.

No mesh convergence studies have been performed, but the comparison with results given in Lippmann (1979) suggests that these meshes provide accuracy similar to the best of those analyses.

The Abaqus/Explicit simulations are performed both with and without adaptive meshing.

Material

The material definition is basically that given in Lippmann (1979), except that the metal is assumed to be rate dependent. The thermal properties are added, with values that correspond to a typical steel, as well as the data for the porous metal plasticity model. The material properties are then as follows:

Young's modulus:	200 GPa
Poisson's ratio:	0.3
Thermal expansion coefficient:	1.2×10^{-5} per °C
Initial static yield stress:	700 MPa
Work hardening rate:	300 MPa
Strain rate dependence:	$\dot{\varepsilon}^{pl} = D\big((\sigma/\sigma^{\circ}) - 1\big)^{p}$; $D = 40$/s, $p = 5$
Specific heat:	586 J/(kg°C)
Density:	7833 kg/m^3
Conductivity:	52 J/(m-s-°C)
Porous material parameters:	$q_1 = q_2 = q_3 = 1.0$
Initial relative density:	0.95 ($f_0 = 0.05$)

Since the problem definition in Abaqus/Standard assumes that the dies are completely rough, no tangential slipping is allowed wherever the metal contacts the die.

Boundary conditions and loading

The kinematic boundary conditions are symmetry on the axis (nodes at $r = 0$, in node set **AXIS**, have $u_r = 0$ prescribed) and symmetry about $z = 0$ (all nodes at $z = 0$, in node set **MIDDLE**, have $u_z = 0$ prescribed). To avoid overconstraint, the node on the top surface of the billet that lies on the symmetry axis is not part of the node set **AXIS**: the radial motion of this node is already constrained by a no-slip frictional constraint (see "Common difficulties associated with contact modeling in Abaqus/Standard," Section 39.1.2 of the Abaqus Analysis User's Guide, and "Common difficulties associated with contact modeling using contact pairs in Abaqus/Explicit," Section 39.2.2 of the Abaqus Analysis User's Guide). The rigid body reference node for the rigid surface that defines the die is constrained to have no rotation or u_r-displacement, and its u_z-displacement is prescribed to move 9 mm down the axis at constant velocity. The reaction force at the rigid reference node corresponds to the total force applied by the die.

The thermal boundary conditions are that all external surfaces are insulated (no heat flux allowed). This condition is chosen because it is the most extreme case: it must provide the largest temperature rises possible, since no heat can be removed from the specimen.

One of the controls for the automatic time incrementation scheme in Abaqus/Standard is the limit on the maximum temperature change allowed to occur in any increment. It is set to 100°C, which is a large value and indicates that we are not restricting the time increments because of accuracy considerations in integrating the heat transfer equations. In fact, the automatic time incrementation scheme will choose fairly small increments because of the severe nonlinearity present in the problem and the resultant need for several iterations per increment even with a relatively large number of increments. The maximum

allowable temperature change in an increment is set to a large value to obtain a reasonable solution at low cost.

In Abaqus/Explicit the automatic time incrementation scheme is used to ensure numerical stability and to advance the solution in time. Mass scaling is used to reduce the computational cost of the analysis.

The amplitude is applied linearly over the step because the default amplitude variation for a transient, coupled temperature-displacement analysis is a step function, but here we want the die to move down at a constant velocity.

Two versions of the analysis are run: a slow upsetting, where the upsetting occurs in 100 seconds, and a fast upsetting, where the event takes 0.1 second. Both versions are analyzed with the coupled temperature-displacement procedure. The fast upsetting is also run in Abaqus/Standard as an adiabatic static stress analysis. The time period values are specified with the respective procedure options. The adiabatic stress analysis is performed in the same time frame as the fast upsetting case. In all cases analyzed with Abaqus/Standard an initial time increment of 1.5% of the time period is used; that is, 1.5 seconds in the slow case and 0.0015 second in the fast case. This value is chosen because it will result in a nominal axial strain of about 1% per increment, and experience suggests that such increment sizes are generally suitable for cases like this.

Results and discussion

The results of the Abaqus/Standard simulations are discussed first, beginning with the results for the viscoplastic fully dense material. The results of the slow upsetting are illustrated in Figure 1.3.16–2 to Figure 1.3.16–4. The results for the fast upsetting coupled temperature-displacement analysis are illustrated in Figure 1.3.16–5 to Figure 1.3.16–7; those for the adiabatic static stress analysis are shown in Figure 1.3.16–8 and Figure 1.3.16–9. Figure 1.3.16–2 and Figure 1.3.16–5 show the configuration that is predicted at 60% upsetting. The configuration for the adiabatic analysis is not shown since it is almost identical to the fast upsetting coupled case. Both the slow and the fast upsetting cases show the folding of the top outside surface of the billet onto the die, as well as the severe straining of the middle of the specimen. The second figure in each series (Figure 1.3.16–3 for the slow case, Figure 1.3.16–6 for the fast case, and Figure 1.3.16–8 for the adiabatic case) shows the equivalent plastic strain in the billet. Peak strains of around 180% occur in the center of the specimen. The third figure in each series (Figure 1.3.16–4 for the slow case, Figure 1.3.16–7 for the fast case, and Figure 1.3.16–9 for the adiabatic case) shows the temperature distributions, which are noticeably different between the slow and fast upsetting cases. In the slow case there is time for the heat to diffuse (the 60% upsetting takes place in 100 sec, on a specimen where a typical length is 10 mm), so the temperature distribution at 100 sec is quite uniform, varying only between 180°C and 185°C through the billet. In contrast, the fast upsetting occurs too quickly for the heat to diffuse. In this case the middle of the top surface of the specimen remains at 0°C at the end of the event, while the center of the specimen heats up to almost 600°C. There is no significant difference in temperatures between the fast coupled case and the adiabatic case. In the outer top section of the billet there are differences that are a result of the severe distortion of the elements in that region and the lack of dissipation of generated heat. The temperature in the rest of the billet compares well. This example illustrates the advantage of an adiabatic analysis, since a good representation of the results is obtained in about 60% of the computer time required for the fully coupled analysis.

The results of the slow and fast upsetting of the billet modeled with the porous metal plasticity model are shown in Figure 1.3.16–10 to Figure 1.3.16–15. The deformed configuration is identical to that of Figure 1.3.16–2 and Figure 1.3.16–5. The extent of growth/closure of the voids in the specimen at the end of the analysis is shown in Figure 1.3.16–10 and Figure 1.3.16–13. The porous material is almost fully compacted near the center of the billet because of the compressive nature of the stress field in that region; on the other hand, the corner element is folded up and stretched out near the outer top portion of the billet, increasing the void volume fraction to almost 0.1 (or 10%) and indicating that tearing of the material is likely. The equivalent plastic strain is shown in Figure 1.3.16–11 (slow upsetting) and Figure 1.3.16–14 (fast upsetting) for the porous material; Figure 1.3.16–12 and Figure 1.3.16–15 show the temperature distribution for the slow and the fast upsetting of the porous metal. The porous metal needs less external work to achieve the same deformation compared to a fully dense metal. Consequently, there is less plastic work being dissipated as heat; hence, the temperature increase is not as much as that of fully dense metal. This effect is more pronounced in the fast upsetting problem, where the specimen heats up to only 510°C, compared to about 600°C for fully dense metal.

Figure 1.3.16–16 to Figure 1.3.16–18 show predictions of total upsetting force versus displacement of the die. In Figure 1.3.16–16 the slow upsetting viscoplastic and porous plasticity results are compared with several elastic-plastic and rigid-plastic results that were collected by Lippmann (1979) and slow viscoplastic results obtained by Taylor (1981). There is general agreement between all the rate-independent results, and these correspond to the slow viscoplastic results of the present example and of those found by Taylor (1981). In Figure 1.3.16–17 rate dependence of the yield stress is investigated. The fast viscoplastic and porous plasticity results show significantly higher force values throughout the event than the slow results. This effect can be estimated easily. A nominal strain rate of 6 sec is maintained throughout the event. With the viscoplastic model that is used, this effect increases the yield stress by 68%. This factor is very close to the load amplification factor that appears in Figure 1.3.16–17. Figure 1.3.16–18 shows that the force versus displacement prediction of the fast viscoplastic adiabatic analysis agrees well with the fully coupled results.

Two cases using an element-based rigid surface to model the die are also considered in Abaqus/Standard. To define the element-based rigid surface, the elements are assigned to rigid bodies using an isothermal rigid body constraint. The results agree very well with the case when the analytical rigid surface is used.

The automatic load incrementation results suggest that overall nominal strain increments of about 2% per increment were obtained, which is slightly better than what was anticipated in the initial time increment suggestion. These values are typical for problems of this class and are useful guidelines for estimating the computational effort required for such cases.

The results obtained with Abaqus/Explicit compare well with those obtained with Abaqus/Standard, as illustrated in Figure 1.3.16–19, which compares the results obtained with Abaqus/Explicit (without adaptive meshing) for the total upsetting force versus the displacement of the die against the same results obtained with Abaqus/Standard. The agreement between the two solutions is excellent. Similar agreement is obtained with the results obtained from the Abaqus/Explicit simulation using adaptive meshing. The mesh distortion is significantly reduced in this case, as illustrated in Figure 1.3.16–20.

Input files

Abaqus/Standard input files

cylbillet_cax4t_slow_dense.inp	Slow upsetting case with 144 CAX4T elements, using the fully dense material.
cylbillet_cax4t_fast_dense.inp	Fast upsetting case with 144 CAX4T elements, using the fully dense material.
cylbillet_cax4rt_slow_dense.inp	Slow upsetting case with 144 CAX4RT elements, using the fully dense material.
cylbillet_cax4rt_fast_dense.inp	Fast upsetting case with 144 CAX4RT elements, using the fully dense material.
cylbillet_cax8rt_slow_dense.inp	Slow upsetting case with CAX8RT elements, using the fully dense material.
cylbillet_cax8rt_rb_s_dense.inp	Slow upsetting case with CAX8RT elements, using the fully dense material and an element-based rigid surface for the die.
cylbillet_cax8rt_fast_dense.inp	Fast upsetting case with CAX8RT elements, using the fully dense material.
cylbillet_cax8rt_slow_por.inp	Slow upsetting case with CAX8RT elements, using the porous material.
cylbillet_cax8rt_fast_por.inp	Fast upsetting case with CAX8RT elements, using the porous material.
cylbillet_cgax4t_slow_dense.inp	Slow upsetting case with 144 CGAX4T elements, using the fully dense material.
cylbillet_cgax4t_fast_dense.inp	Fast upsetting case with 144 CGAX4T elements, using the fully dense material.
cylbillet_cgax4t_rb_f_dense.inp	Fast upsetting case with 144 CGAX4T elements, using the fully dense material and an element-based rigid surface for the die.
cylbillet_cgax4t_rb_f_dense_surf.inp	Fast upsetting case with 144 CGAX4T elements, using the fully dense material and an element-based rigid surface for the die with surface-to-surface contact.
cylbillet_cgax8rt_slow_dense.inp	Slow upsetting case with CGAX8RT elements, using the fully dense material.
cylbillet_cgax8rt_fast_dense.inp	Fast upsetting case with CGAX8RT elements, using the fully dense material.
cylbillet_c3d10m_adiab_dense.inp	Adiabatic static analysis with fully dense material modeled with C3D10M elements.
cylbillet_c3d10m_adiab_dense_surf.inp	Adiabatic static analysis with fully dense material modeled with C3D10M elements using surface-to-surface contact.

cylbillet_c3d10m_adiab_dense_po.inp	*POST OUTPUT analysis of cylbillet_c3d10m_adiab_dense.inp.
cylbillet_cax6m_adiab_dense.inp	Adiabatic static analysis with fully dense material modeled with CAX6M elements.
cylbillet_cax8r_adiab_dense.inp	Adiabatic static analysis with fully dense material modeled with CAX8R elements.
cylbillet_postoutput.inp	*POST OUTPUT analysis using the fully dense material.
cylbillet_slow_usr_umat_umatht.inp	Slow upsetting case with the material behavior defined in user subroutines **UMAT** and **UMATHT**.
cylbillet_slow_usr_umat_umatht.f	User subroutines **UMAT** and **UMATHT** used in cylbillet_slow_usr_umat_umatht.inp.

Abaqus/Explicit input files

cylbillet_x_cax4rt_slow.inp	Slow upsetting case with fully dense material modeled with CAX4RT elements and without adaptive meshing; kinematic mechanical contact.
cylbillet_x_cax4rt_fast.inp	Fast upsetting case with fully dense material modeled with CAX4RT elements and without adaptive meshing; kinematic mechanical contact.
cylbillet_x_cax4rt_slow_adap.inp	Slow upsetting case with fully dense material modeled with CAX4RT elements and with adaptive meshing; kinematic mechanical contact.
cylbillet_x_cax4rt_fast_adap.inp	Fast upsetting case with fully dense material modeled with CAX4RT elements and with adaptive meshing; kinematic mechanical contact.
cylbillet_xp_cax4rt_fast.inp	Fast upsetting case with fully dense material modeled with CAX4RT elements and without adaptive meshing; penalty mechanical contact.

References

- Lippmann, H., *Metal Forming Plasticity,* Springer-Verlag, Berlin, 1979.
- Taylor, L. M., “A Finite Element Analysis for Large Deformation Metal Forming Problems Involving Contact and Friction,” Ph.D. Thesis, U. of Texas at Austin, 1981.

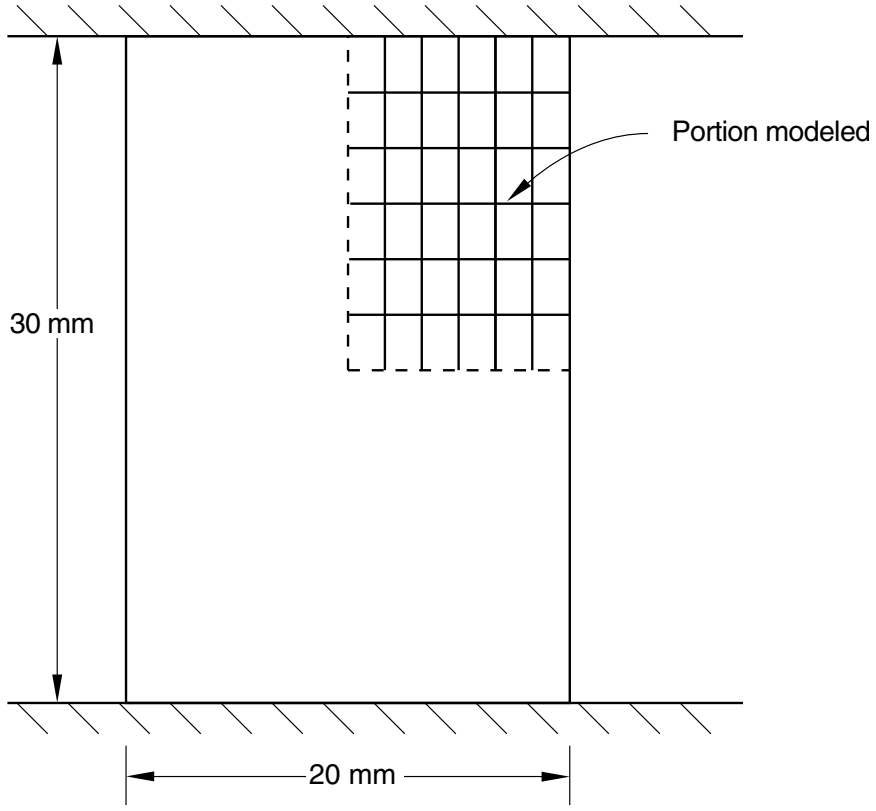


Figure 1.3.16–1 Axisymmetric upsetting example: geometry and mesh (element type CAX8RT).

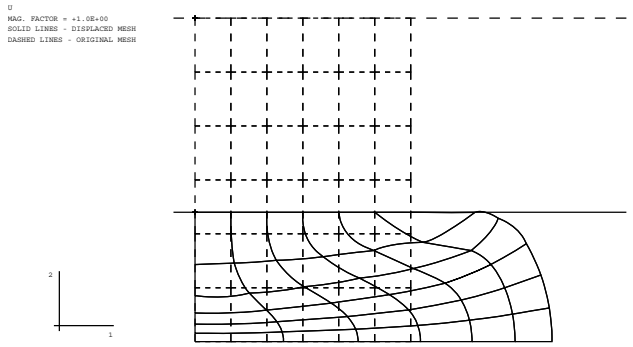


Figure 1.3.16–2 Deformed configuration at 60% upsetting: slow case, coupled temperature-displacement analysis, Abaqus/Standard.

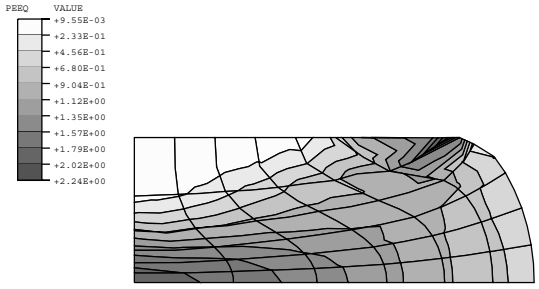


Figure 1.3.16–3 Plastic strain at 60% upsetting: slow case, coupled temperature-displacement analysis, Abaqus/Standard.

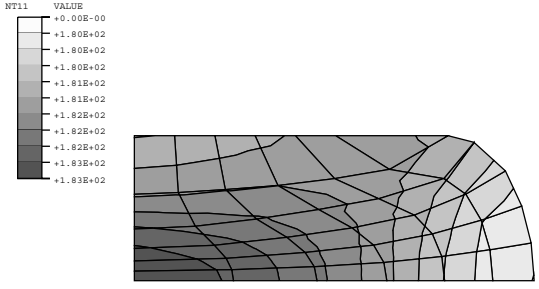


Figure 1.3.16–4 Temperature at 60% upsetting: slow case, coupled temperature-displacement analysis, Abaqus/Standard.

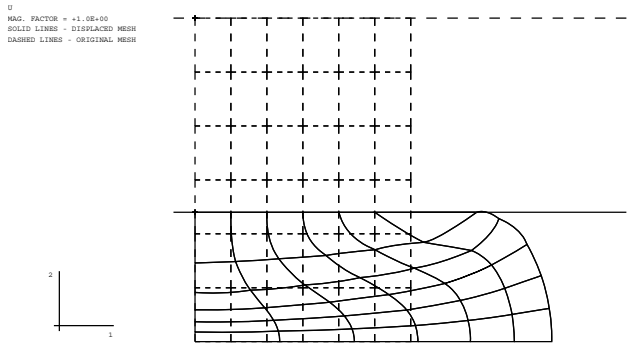


Figure 1.3.16–5 Deformed configuration at 60% upsetting: fast case, coupled temperature-displacement analysis, Abaqus/Standard.

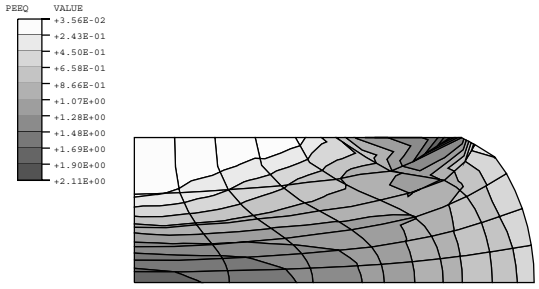


Figure 1.3.16–6 Plastic strain at 60% upsetting: fast case, coupled temperature-displacement analysis, Abaqus/Standard.

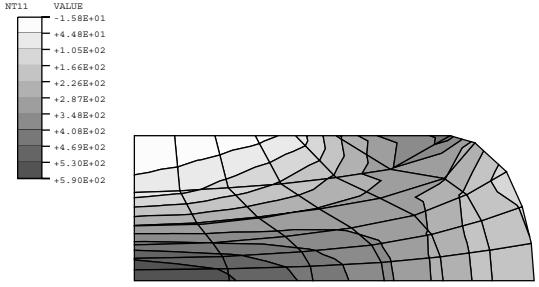


Figure 1.3.16–7 Temperature at 60% upsetting: fast case, coupled temperature-displacement analysis, Abaqus/Standard.

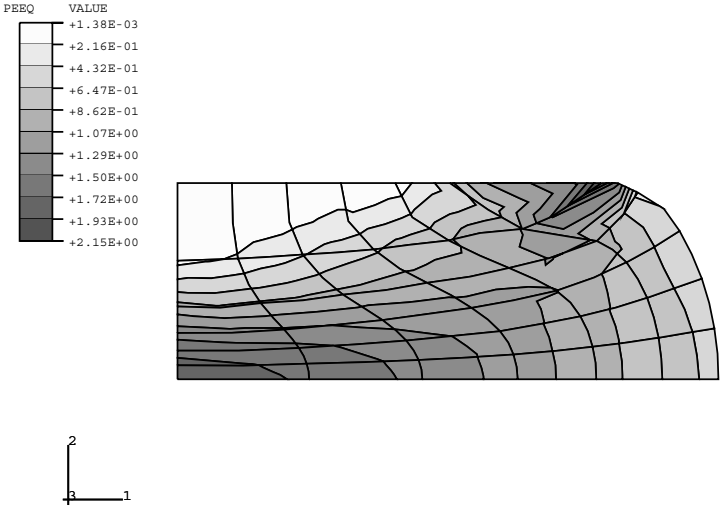


Figure 1.3.16–8 Plastic strain at 60% upsetting: fast case, adiabatic stress analysis, Abaqus/Standard.

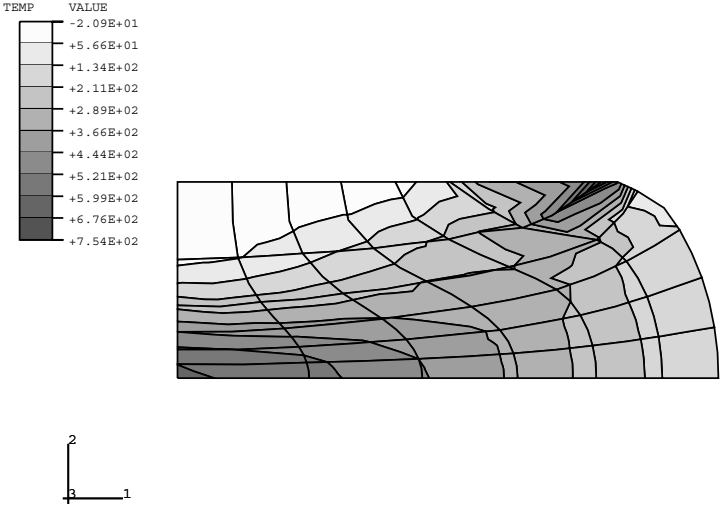


Figure 1.3.16–9 Temperature at 60% upsetting: fast case, adiabatic stress analysis, Abaqus/Standard.

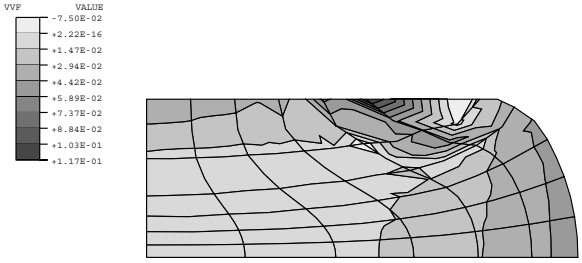


Figure 1.3.16–10 Void volume fraction at 60% upsetting: porous material, slow coupled temperature-displacement analysis, Abaqus/Standard.

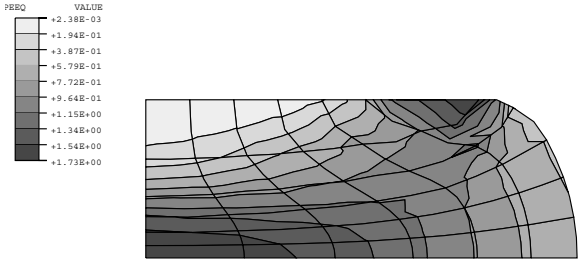


Figure 1.3.16–11 Plastic strain at 60% upsetting: porous material, slow coupled temperature-displacement analysis, Abaqus/Standard.

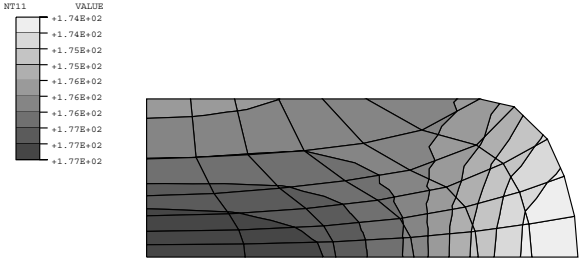


Figure 1.3.16–12 Temperature at 60% upsetting: porous material, slow coupled temperature-displacement analysis, Abaqus/Standard.

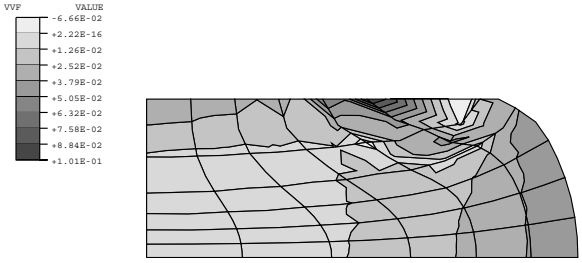


Figure 1.3.16–13 Void volume fraction at 60% upsetting: porous material, fast coupled temperature-displacement analysis, Abaqus/Standard.

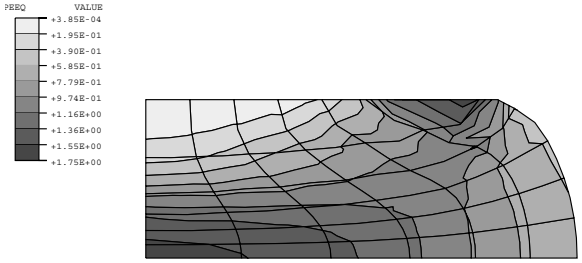


Figure 1.3.16–14 Plastic strain at 60% upsetting: porous material, fast coupled temperature-displacement analysis, Abaqus/Standard.

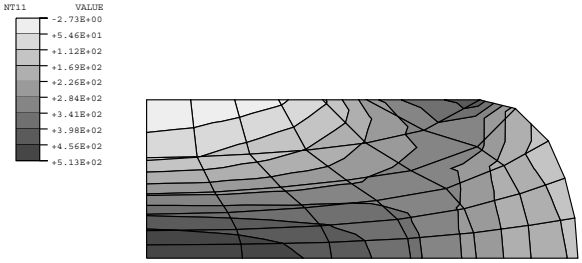


Figure 1.3.16–15 Temperature at 60% upsetting: porous material, fast coupled temperature-displacement analysis, Abaqus/Standard.

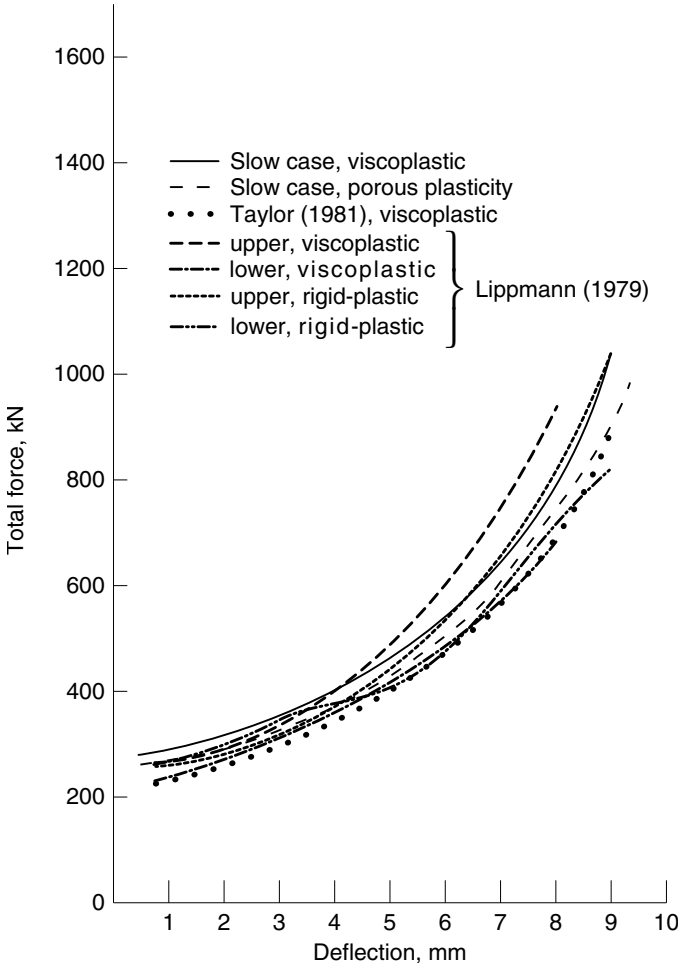


Figure 1.3.16–16 Force-deflection response for slow cylinder upsetting, Abaqus/Standard.

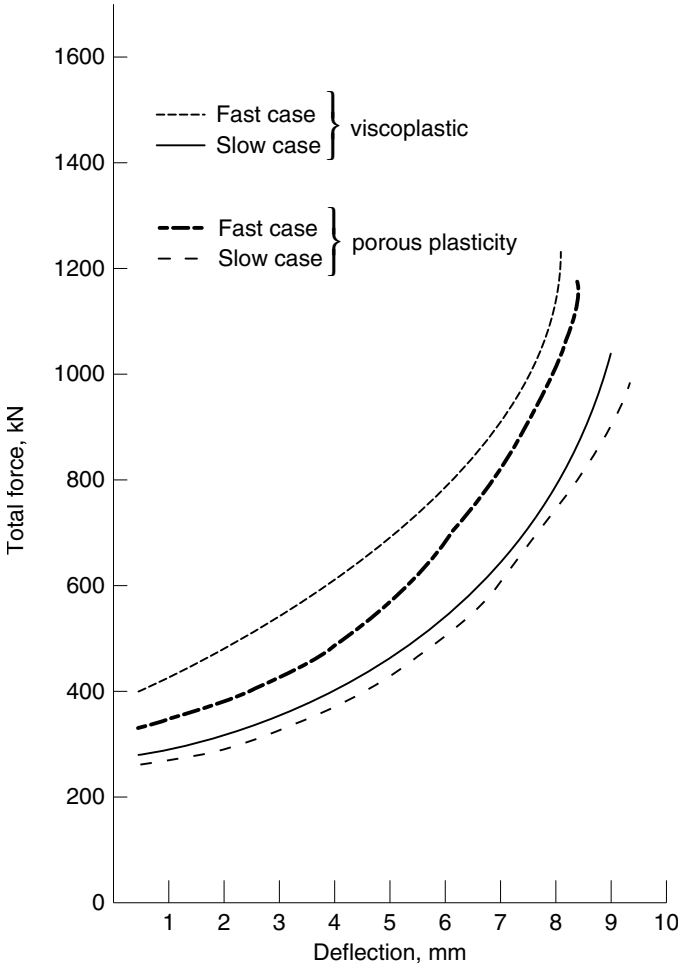


Figure 1.3.16–17 Rate dependence of the force-deflection response, Abaqus/Standard.

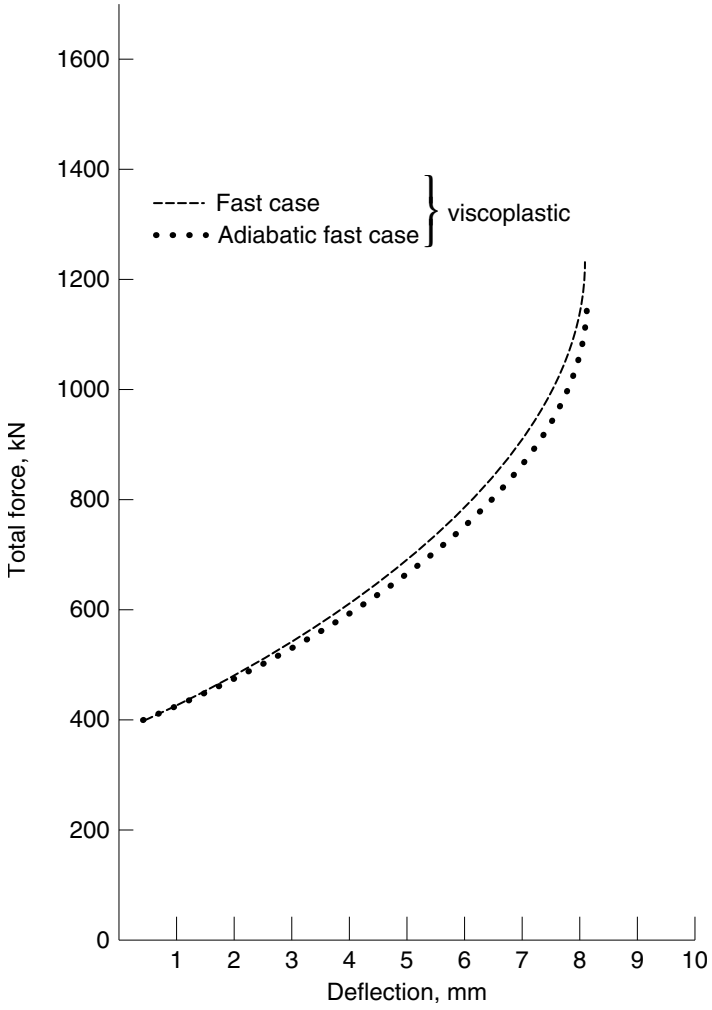


Figure 1.3.16–18 Force-deflection response: adiabatic versus fully coupled analysis, Abaqus/Standard.

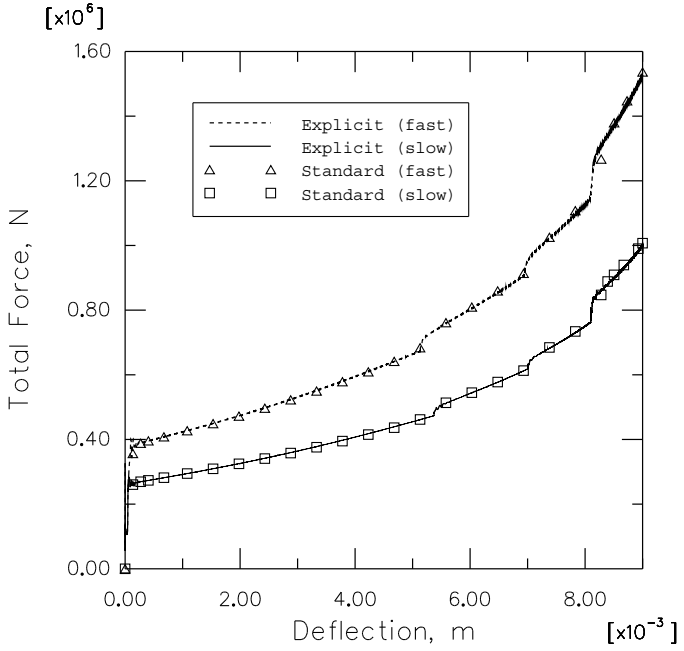


Figure 1.3.16–19 Force-deflection response: Abaqus/Explicit versus Abaqus/Standard.

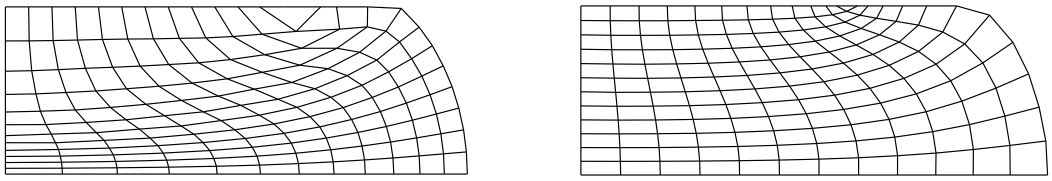


Figure 1.3.16–20 Deformed configuration at 60% upsetting: slow case, Abaqus/Explicit (without adaptive meshing, left; with adaptive meshing, right).

1.3.17 UNSTABLE STATIC PROBLEM: THERMAL FORMING OF A METAL SHEET

Product: Abaqus/Standard

This example demonstrates the use of automatic contact stabilization to avoid unstable static problems. Geometrically nonlinear static problems can become unstable for a variety of reasons. Instability may occur in contact problems, either because of chattering or because contact intended to prevent rigid body motions is not established initially. Localized instabilities can also occur; they can be either geometrical, such as local buckling, or material, such as material softening.

This problem models the thermal forming of a metal sheet; the shape of the die may make it difficult to place the undeformed sheet exactly in initial contact, in which case the initial rigid body motion prevention algorithm is useful. Metal forming problems are characterized by relatively simply shaped parts being deformed by relatively complex-shaped dies. The initial placement of the workpiece on a die or the initial placement of a second die may not be a trivial geometrical exercise for an engineer modeling the forming process. Abaqus accepts initial penetrations in contact pairs and instantaneously tries to resolve them; as long as the geometry allows for this to happen without excessive deformation, the misplacement of the workpiece usually does not cause problems. On the other hand, if the workpiece is initially placed away from the dies, serious convergence problems may arise. Unless there are enough boundary conditions applied or a stabilization method is used, singular finite element systems of equations result because one or more of the bodies has free rigid body motions. This typically arises when the deformation is applied through loads instead of boundary conditions. Contact stabilization can be helpful for avoiding convergence problems while contact is established without significantly influencing the results of interest (see "Automatic stabilization of rigid body motions in contact problems" in "Adjusting contact controls in Abaqus/Standard," Section 36.3.6 of the Abaqus Analysis User's Guide).

This example looks at the thermal forming of an aluminum sheet. The deformation is produced by applying pressure and gravity loads to push the sheet against a sculptured die. The deformation is initially elastic. Through heating, the yield stress of the material is lowered until permanent plastic deformations are produced. Subsequently, the assembly is cooled and the pressure loads are removed, leaving a formed part with some springback. Although the sheet is initially flat, the geometrical nature of the die makes it difficult to determine the exact location of the sheet when it is placed on the die. Therefore, an initial gap between the two bodies is modeled, as shown in Figure 1.3.17–1.

Geometry and model

The model consists of a trapezoidal sheet 10.0 m (394.0 in) long, tapering from 2.0 m (78.75 in) to 3.0 m (118.0 in) wide, and 10.0 mm (0.4 in) thick. The die is a ruled surface controlled by two circles of radii 13.0 m (517.0 in) and 6.0 m (242.0 in) and dimensions slightly larger than the sheet. The sheet is initially placed over 0.2 m (7.9 in) apart from the die. The sheet has a longitudinal symmetry boundary condition, and one node prevents the remaining nodes from experiencing in-plane rigid body motion. The die is fixed throughout the analysis. The sheet mesh consists of 640 S4R shell elements, while the die is represented by 640 R3D4 rigid elements. The material is an aluminum alloy with a flow stress of 1.0×10^8 Pa (14.5 ksi) at room temperature. A flow stress of 1.0×10^3 Pa (0.15 psi) at 400°C is also

provided, essentially declaring that at the higher temperature the material will flow plastically at any stress. A Coulomb friction coefficient of 0.1 is used to model the interaction between the sheet and die.

Results and discussion

The analysis consists of three steps. In the first step a gravity load and a pressure load of 1.0×10^5 Pa (14.5 psi) are applied, both pushing the sheet against the die. This step is aided by contact stabilization to prevent unrestrained motion of the sheet prior to establishing contact with the die. In this case, contact stabilization normal to the nearby contact surfaces provides adequate stabilization. Avoiding tangential contact stabilization is recommended, if possible, because tangential contact stabilization is more likely to influence solution variables. The clearance distance range over which the contact stabilization is effective has been specified in this example such that contact stabilization is active for the initial separation distance between the sheet and die. Abaqus ramps down the contact stabilization such that no contact stabilization remains at the end of the first step. This guarantees that the viscous forces decrease to zero, thus avoiding any discontinuity in the forces at the start of the next step. The shape and relatively low curvatures of the die are such that the deformation at the end of the step is elastic (Figure 1.3.17–2). In the second step a two-hour heating (from room temperature to 360°C) and cooling (back to 50°C) cycle is applied to the loaded assembly. As a result of the decrease in flow stress permanent (plastic) deformation develops, as shown in Figure 1.3.17–3. Finally, in the third step the pressure load is removed and the springback of the deformed sheet is calculated, as depicted in Figure 1.3.17–4.

Acknowledgments

SIMULIA would like to thank British Aerospace Airbus, Ltd. for providing the basic data from which this example was derived.

Input files

unstablestatic_forming.inp	Thermal forming model.
unstablestatic_forming_surf.inp	Thermal forming model with surface-to-surface contact.

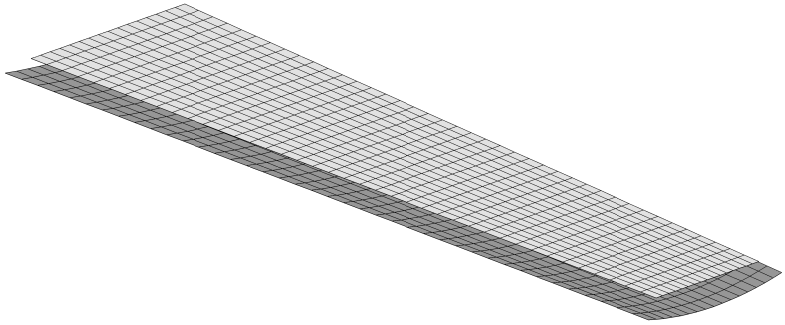


Figure 1.3.17–1 Initial placement of the sheet apart from the die.

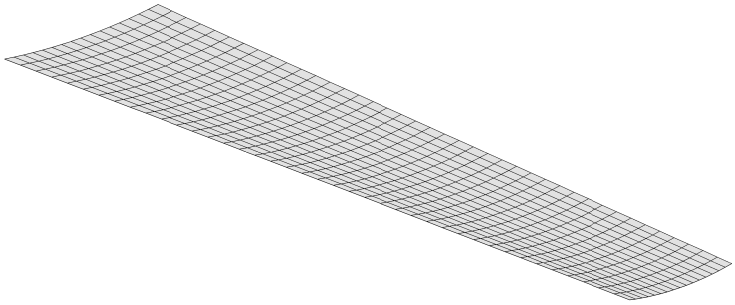


Figure 1.3.17–2 Elastic deformation after gravity and pressure loading.

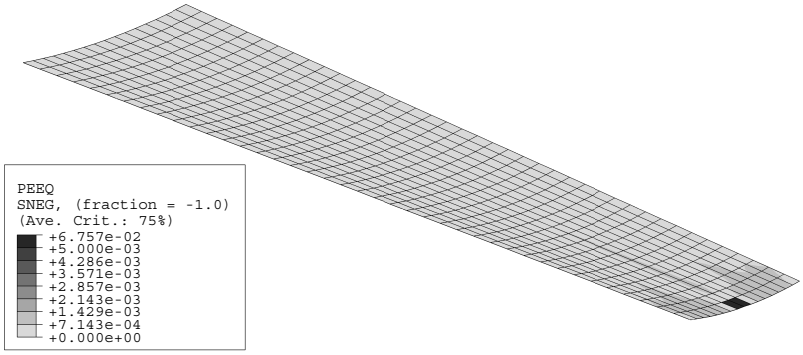


Figure 1.3.17–3 Permanent deformation produced by heating.

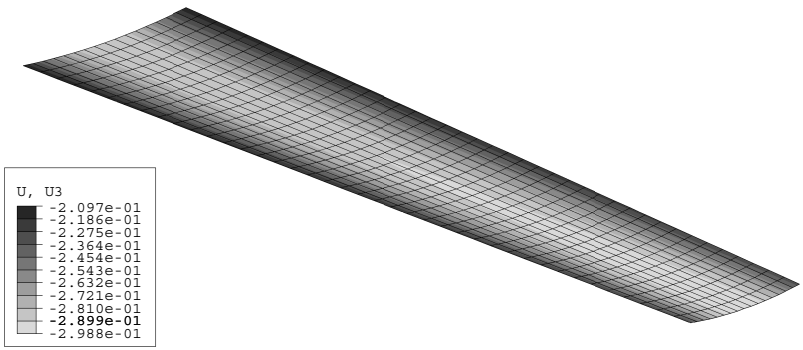


Figure 1.3.17–4 Springback.

1.3.18 INERTIA WELDING SIMULATION USING Abaqus/Standard AND Abaqus/CAE

Products: Abaqus/Standard Abaqus/CAE

Objectives

This example demonstrates the following Abaqus features:

- thermal-mechanical coupling for inertia welding simulation,
- semi-automatic remeshing using Python scripting and output database scripting methods for extracting deformed configurations,
- defining a complex friction law in a user subroutine,
- flywheel loading through user subroutine definitions, and
- combining and presenting results from a sequence of output database (`.odb`) files.

Application description

This example examines the inertia friction welding process of the pipes shown in Figure 1.3.18–1. The specific arrangement considered is the resulting as-welded configuration shown in Figure 1.3.18–2. In this weld process kinetic energy is converted rapidly to thermal energy at a frictional interface. The resulting rapid rise in interface temperature is exploited to produce high-quality welds. In this example the weld process is simulated, and the initial temperature rise and material plastic flow are observed. An important factor in the process design is control of the initial speed of the flywheel so that, when the flywheel stops, the temperature rises to just below the melting point, which in turn results in significant flow of material in the region of the weld joint. Understanding the friction, material properties, and heat transfer environment are important design aspects in an effective inertia welding process; therefore, simulation is a helpful tool in the process design.

Geometry

The weld process in this example is shown in Figure 1.3.18–1, where two pipes are positioned for girth-weld joining. The two pipes are identical, each with a length of 21.0 mm, an inside radius of 42.0 mm, and an outside radius of 48.0 mm. The pipes are adjacent, touching each other initially at the intended weld interface.

Materials

The pipes are made of Astroloy, a high-strength alloy used in gas turbine components. Figure 1.3.18–3 shows flow stress curves as a function of temperature and plastic strain rate. At temperatures relevant to the welding process, the material is highly sensitive to plastic strain rate and temperature. Specific heat is a function of temperature, as shown in Figure 1.3.18–4.

Other material properties are defined as follows:

Young's modulus:	180,000 MPa
Poisson's ratio:	0.3
Density:	7.8×10^{-9} Mg/mm^3
Conductivity:	14.7 W/m/°C at 20°C
	28 W/m/°C at 1200°C

Initial conditions

The pipes are initially set at 20°C, representing room temperature.

Boundary conditions and loading

A pressure of 360 MPa is applied to the top surface of the upper pipe. The initial rotational velocity of the flywheel is set at 48.17 rad/s, or 7.7 revolutions per second. The mass moment of inertia of the flywheel is 102,000 Mg mm^2.

Interactions

The principal interaction occurs at the weld interface between the pipes; however, a secondary concern is the possibility of contact of weld flash with the side of the pipes. The weld-interface friction behavior is assumed to follow that described by Moal and Massoni (1995), where the ratio of shear stress to the prescribed pressure is observed to be a complex function of interface slip rate. The heat generation from the frictional sliding, combined with plastic deformation, contributes to the temperature rise in the pipes.

Abaqus modeling approaches and simulation techniques

Abaqus/CAE and Abaqus/Standard are used together to affect the weld simulation in a way that permits extreme deformation of the pipes in the weld region. This process is automated through the use of Python scripts. Three cases are studied in this example.

Summary of analysis cases

Case 1 Initial flywheel velocity = 48.17 rad/s. This case produces a successful weld.

Case 2 Initial flywheel velocity = 20.0 rad/s. This case illustrates an unsuccessful weld scenario; the flywheel has insufficient energy to begin the weld process.

Case 3 Initial flywheel velocity = 70.0 rad/s. This case illustrates an unsuccessful weld scenario; the flywheel has excessive energy, resulting in a temperature rise into the liquidus regime of the pipe material.

The following sections discuss analysis considerations that are applicable to all the cases. Python scripts that generate the model databases and Abaqus/Standard input files are provided for Case 1, with instructions in the scripts for executing the Case 2 and Case 3 simulations.

Analysis types

The analysis is nonlinear, quasi-static with thermal-mechanical coupling. A fully coupled temperature-displacement procedure is used.

Analysis techniques

The key feature required for successful simulation of this process is remeshing. In this example, because of the large deformation near the weld region, multiple analyses are employed to limit element distortion. These analyses are executed in sequence, with remeshing performed between executions, and are automated through the use of Python scripts.

At each remesh point the current model configuration represents a significant change in the pipes' shape and in the current analysis mesh. Abaqus/CAE is used to extract the outer surface of the pipes, reseed the surface, and remesh the pipe regions. This process employs the Abaqus Scripting Interface **`PartFromOdb`** command, which is used to extract orphan mesh parts representing the deformed pipes. These parts are then passed to the **`Part2DGeomFrom2DMesh`** command. This command creates a geometric Part object from the orphan mesh imported earlier. Once the profile of the deformed part has been created, options in the Mesh module are used to remesh the part. The new mesh results in a new Abaqus/Standard analysis, and the map solution procedure maps state variables from the previous analysis (see "Mesh-to-mesh solution mapping," Section 12.4.1 of the Abaqus Analysis User's Guide).

Mesh design

The pipes are modeled as axisymmetric. The element formulation used is the fully coupled temperature-displacement axisymmetric elements with twist degrees of freedom (element types CGAX4HT and CGAX3HT), where the twist degree of freedom enables modeling of rotation and shear deformation in the out-of-plane direction. The hybrid formulation is required to handle the incompressible nature of the material during the plastic flow. The mesh is divided into two regions for each pipe. In the region near the weld interface, smaller elements are created (see Figure 1.3.18–5). During the remeshing process, the region near the weld surface is recalculated so that the new flash region is also meshed with smaller elements (see Figure 1.3.18–6).

Material model

The material model defined for this example approximates the high-temperature behavior of Astroloy, where it is reported by Soucail et al. (1992) using a Norton-Hoff constitutive law to describe the temperature and strain-rate viscoplastic behavior. A similar model is defined in Abaqus as a rate-dependent perfectly plastic material model. For the loading in this model, these material parameters result in the onset of local plastic flow only after the interface temperature has exceeded roughly 1200°C, near the material solidus temperature of 1250°C. Above this temperature the Mises flow stress is highly sensitive to variations in temperature and strain rate. A special adjustment in the flow stress at high strain rates is necessary to avoid divergence during the iteration procedure of the nonlinear solution. In the material model definition an extreme case of stress data is defined when the strain rate is 1.0×10^6 s^{-1}. Stress data when the strain rate equals zero are also defined to be the same as the stress

data at strain rate 1.0×10^{-5} s^{-1}. As a result of large deformation, thermal expansion is not considered in the material model.

It is assumed that 90% of the inelastic deformation energy contributes to the internal heat generation, which is the Abaqus default for inelastic heat fraction.

Initial conditions

An initial temperature of 20°C is specified for the entire model. The flywheel initial velocity is specified as a solution-dependent variable initial condition for the user element (see inertiaweld_sub.f).

Boundary conditions

The bottom pipe is fixed completely. The top pipe is free of boundary conditions.

Loading

A pressure of 360 MPa is applied as a distributed load on the top surface of the upper pipe.

The heat flux resulting from frictional heat generation is considered to be many times greater than any heat lost due to convection. Thus, all boundaries are assumed to be adiabatic.

Interactions

Five interactions are defined: four contact interactions and one actuator-sensor interaction.

The contact interactions include a pair of interactions that define the weld interface between the pipes, which is identified in Figure 1.3.18–1. This pair of interactions is symmetrical: one interaction defines the top pipe as the master surface with the bottom pipe as the slave surface, and the second interaction reverses the master-slave pairing. This "balanced master-slave" arrangement is important for the analysis to obtain more uniform contact pressure in the weld interface and to avoid hourglass effects, and it is combined with a softened contact interaction property to promote a sharing of the local contact pressure among nodes on both sides of the interface.

To simulate the Moal and Massoni (1995) friction definition, the weld interface friction model is defined in user subroutine **`FRIC`** (see inertiaweld_sub.f). This model is nonlocal in the sense that the interface pressure for all contact nodes is the applied pressure of 360 MPa, and the sliding velocity is computed based on the rotational angular velocity of the flywheel. This treatment of the friction force helps stabilize the solution. Frictional heat generation is calculated based on the frictional traction and the sliding velocity. Nondefault contact controls definitions are used to improve convergence. These definitions include delaying the friction computation upon contact and automatic tolerance control to avoid contact chattering.

The two remaining contact interactions address the possibility of self-contact near the weld area in the pipes. Self-contact in the flash area can cause problems during remeshing: the **`Part2DGeomFrom2DMesh`** command that was used to generate the new, current configuration geometry performs curve-fit operations that can result in self intersections of the boundary, which lead to invalid part topology and a meshing failure. To avoid this problem, a softened contact model that introduces a normal pressure with a small separation distance (−0.01 mm) is used. It is important to keep this separation distance as small as practical to avoid causing any nonphysical contact behavior.

The actuator-sensor interaction, which acts through user subroutine **UEL**, enables the simulation of a flywheel attached to the top pipe. The user element also has a sensor role in the analysis, measuring the weld upset, or axial shortening, of the weld assembly. When a critical user-defined upset distance is exceeded, the user element subroutine calls **XIT** to terminate the analysis and signal a remesh point. This parameter, the allowed "upset distance," correlates well with the extent of mesh deformation in the weld region.

Analysis steps

Each of the analyses run in the simulation sequence uses a single fully coupled temperature-displacement step. Since the duration of the step is controlled by the user element, as described above, it is not known a priori for each analysis. Therefore, the step duration is set in each analysis to be the total remaining time for the simulation, a time that is reached only in the final analysis in the simulation sequence.

Solution controls

Full Newton iteration is used for the nonlinear solution. Automatic time stepping is based on the maximum temperature change in each increment. Results from the previous increment are used as the initial guess in the current increment, with the parabolic extrapolation technique adopted to transfer the previous state to the current state.

Heat fluxes due to frictional heat generation can cause contact chattering. To overcome this difficulty, contact controls with the automatic tolerances parameter are used.

Output requests

Field output of temperature, stress, and equivalent plastic strain is requested. Various energy measures are requested for history output, allowing you to record the flywheel energy change during the simulation.

Convergence

In general, convergence is difficult in the initial increments, especially after remeshing. More iterations are also observed near the end of the analysis-run sequence when deformation increases and the mesh distortion becomes severe.

Run procedure

This example is run using the Abaqus/CAE Python scripts provided. Use the following command to run the example:

```
abaqus cae -script inertiaweld.py
```

The simulation results in a sequence of separate output database (**.odb**) files, each corresponding to one of the analyses.

For postprocessing convenience a number of scripts are provided for evaluating simulation results across the sequence of files. Use the following command to create the **inertia_weld_animation.avi** animation file:

```
abaqus cae -script inertiaweld_movie.py
```

Run the following Python scripts to create *X*–*Y* plots:

- inertiaweld_xy.py, for a history plot of energy results.
- inertiaweld_upset.py, for a history plot of assembly upset.
- inertiaweld_maxtemp.py, for a history plot of the maximum temperature.

Discussion of results and comparison of cases

This example focuses mainly on the successful weld case, Case 1. Case 2 and Case 3 are provided to show the precarious nature of the inertia weld process, where it is essential to select the flywheel size correctly, measured by the mass moment of inertia, and the initial velocity of the flywheel. Case 2 and Case 3 illustrate why simulation is a valuable tool in the weld process design as each case clearly shows that the selected process parameters do not result in a successful weld.

Welding success: Case 1

The example simulation creates multiple output database (`.odb`) files, requiring 22 remeshings to reach the simulation time of 5.0 seconds. The results in the first analysis before the first remeshing show that the temperature rises very fast near the weld interface. At about 1.6 seconds the temperature reaches 1172°C and the material starts to flow, squeezing out to form flash (see Figure 1.3.18–7). After 2.31 seconds and 5 remeshings the flash extends enough to fold back and contact the pipe (see Figure 1.3.18–8). At 3.51 seconds the flywheel velocity slows down to 3.51 rad/s, the temperature starts to drop, and the material flow slows down. At this point a considerable amount of flash build-up can be observed (see Figure 1.3.18–9). After 5.0 seconds the flywheel stops, the temperature drops below 1000°C, and the pipes are welded (see Figure 1.3.18–10).

The flash prediction in Figure 1.3.18–11 is in good agreement with those shown by Moal and Massoni (1995). Figure 1.3.18–12 plots the kinetic energy history (created with inertiaweld_xy.py) and shows how the flywheel slows down and stops. The "RemeshPoints" marks indicate the time at which the remeshing procedure occurred. The upset distance during the welding process is shown in Figure 1.3.18–13 (created with inertiaweld_upset.py). The curve of the maximum temperature for Case 1 is plotted in Figure 1.3.18–14 (created with inertiaweld_maxtemp.py).

Insufficient flywheel energy: Case 2

If a lower initial flywheel velocity is selected, the temperature may not reach a level high enough for the material to flow. For this case the initial velocity is reduced to 20 rad/s. Figure 1.3.18–15 shows the configuration at 5.0 seconds, where not much deformation is observed and the temperature near the interface is about 250°C. The history plot of the maximum temperature for Case 2 in Figure 1.3.18–14 shows the pipe temperature reaches only 700°C about 1.1 seconds before it cools down. In this case the material is not hot enough to initiate the material flow, and welding will not be successful.

Excessive flywheel energy: Case 3

If a higher initial velocity of the flywheel is selected, the material becomes so hot that it starts to melt. For this case the initial velocity of the flywheel is set at 70 rad/s. Figure 1.3.18–16 shows that the temperature rapidly reaches 1360°C (at about 0.9 seconds), which is well beyond the melting temperature at 1250°C,

before convergence failures stop the analysis. Figure 1.3.18–14 shows the history plot of the maximum temperature in the pipes for Case 3. In this case the excessive energy results in melting and a failed weld.

Python scripts

inertiaweld.py	Main Python script to control model building, remeshing, and the analyses.
inertiaweld_solver_param.py	Script to initialize base options for an Abaqus/Standard analysis.
inertiaweld_utils.py	Script to define utility functions, such as applying seeds and replacing the part after deformation.
inertiaweld_original_build.py	Script to create the initial model.
inertiaweld_job_param.py	Script to provide control parameters and input data, such as mesh size and simulation time.
inertiaweld_remesh_build.py	Script to create a remeshed model and provide input files for restart.
inertiaweld_movie.py	Script to create an animation of this welding simulation.
inertiaweld_xy.py	Script to create an *X*–*Y* plot of flywheel kinetic energy history.
inertiaweld_upset.py	Script to create an *X*–*Y* plot of the upset history.
inertiaweld_maxtemp.py	Script to create an *X*–*Y* plot of the maximum temperature history during the welding process.

Input files

inertiaweld_sub.f	FORTRAN file containing user subroutines **UEL** and **FRIC**.

References

Abaqus Analysis User's Guide

- "Mesh-to-mesh solution mapping," Section 12.4.1 of the Abaqus Analysis User's Guide

Abaqus Keywords Reference Guide

- *MAP SOLUTION

Abaqus User Subroutines Reference Guide

- "FRIC," Section 1.1.8 of the Abaqus User Subroutines Reference Guide
- "UEL," Section 1.1.28 of the Abaqus User Subroutines Reference Guide
- "Terminating an analysis," Section 2.1.15 of the Abaqus User Subroutines Reference Guide

Abaqus Scripting Reference Guide

- "PartFromOdb," Section 37.1.11 of the Abaqus Scripting Reference Guide
- "Part2DGeomFrom2DMesh," Section 37.1.14 of the Abaqus Scripting Reference Guide

Other

- Moal, A., and E. Massoni, "Finite Element Simulation of the Inertia Welding of Two Similar Parts," Engineering Computations, vol. 12, pp. 497–512, 1995.
- Soucail, M., A. Moal, L. Naze, E. Massoni, C. Levaillant, and Y. Bienvenu, "Microstructural Study and Numerical Simulation of Inertia Friction Welding of Astroloy," 7th International Symposium on Superalloys, Seven Springs, USA, 1992.

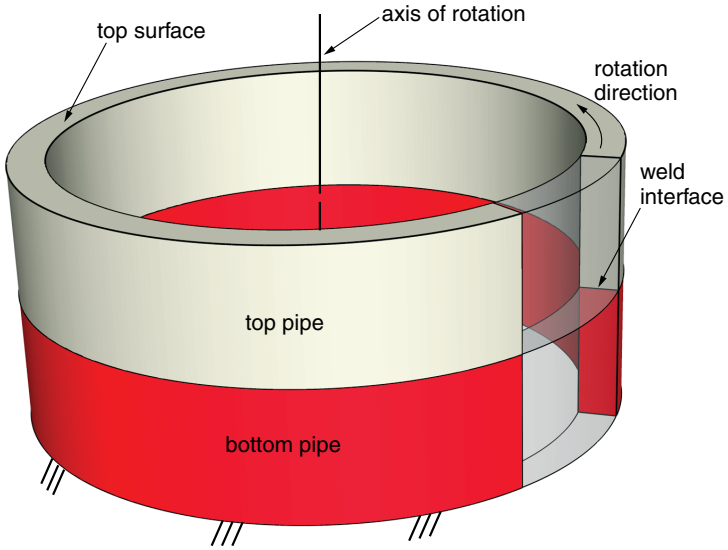


Figure 1.3.18–1 Inertia weld configuration for two pipes.

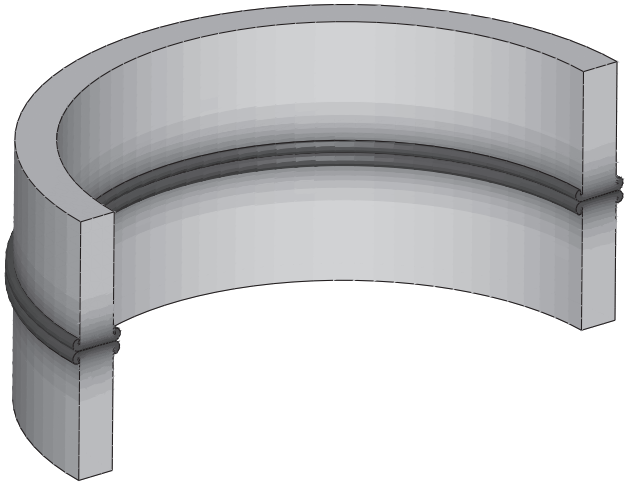


Figure 1.3.18–2 Half-section view of the inertia welding final as-welded configuration showing the temperature fields at 5 seconds after welding begins.

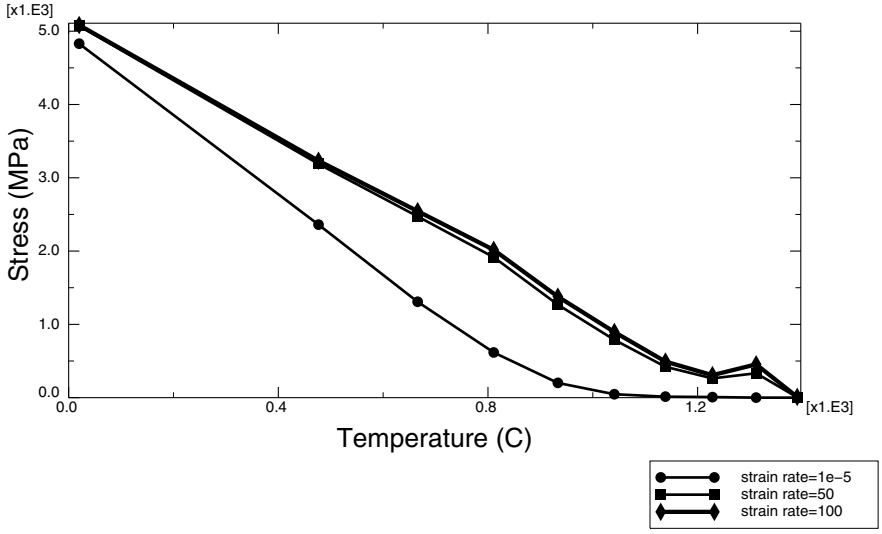


Figure 1.3.18–3 Flow stress curves.

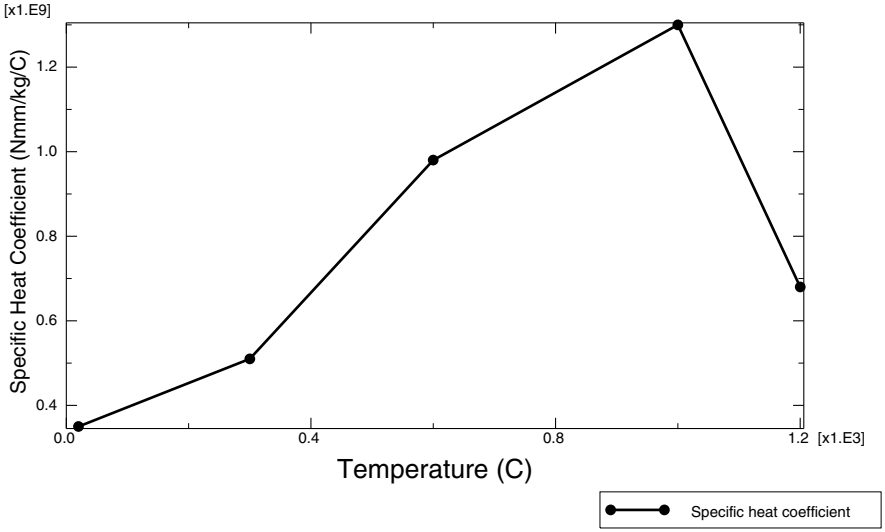


Figure 1.3.18–4 Specific heat curve (N mm/kg/°C).

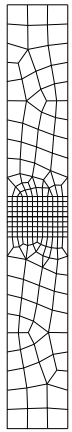


Figure 1.3.18–5 Initial mesh.

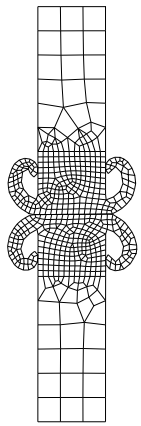


Figure 1.3.18–6 Typical mesh design on a deformed configuration pair of parts.

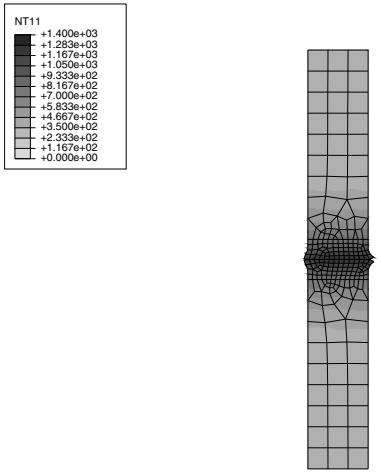


Figure 1.3.18–7 Temperature at 1.78 sec.

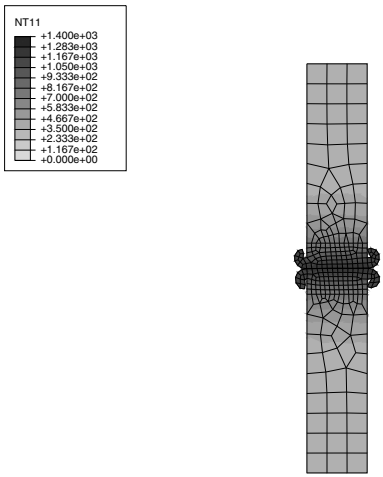


Figure 1.3.18–8 Temperature at 2.31 sec.

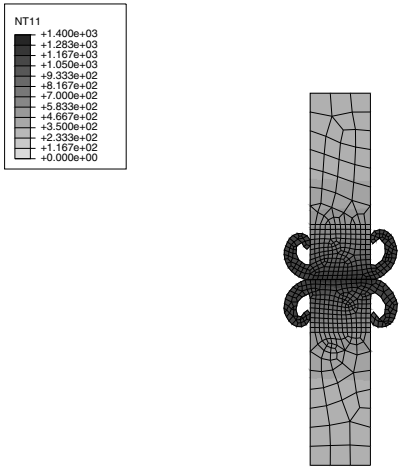


Figure 1.3.18–9 Temperature at 3.51 sec.

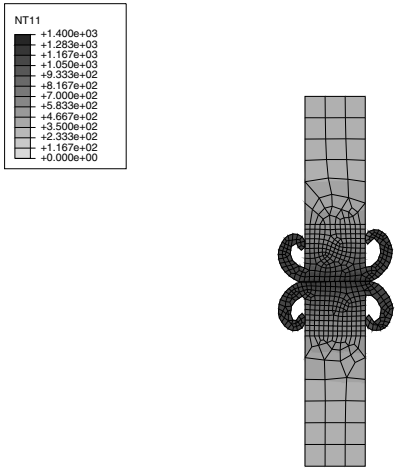


Figure 1.3.18–10 Temperature at 5.0 sec.

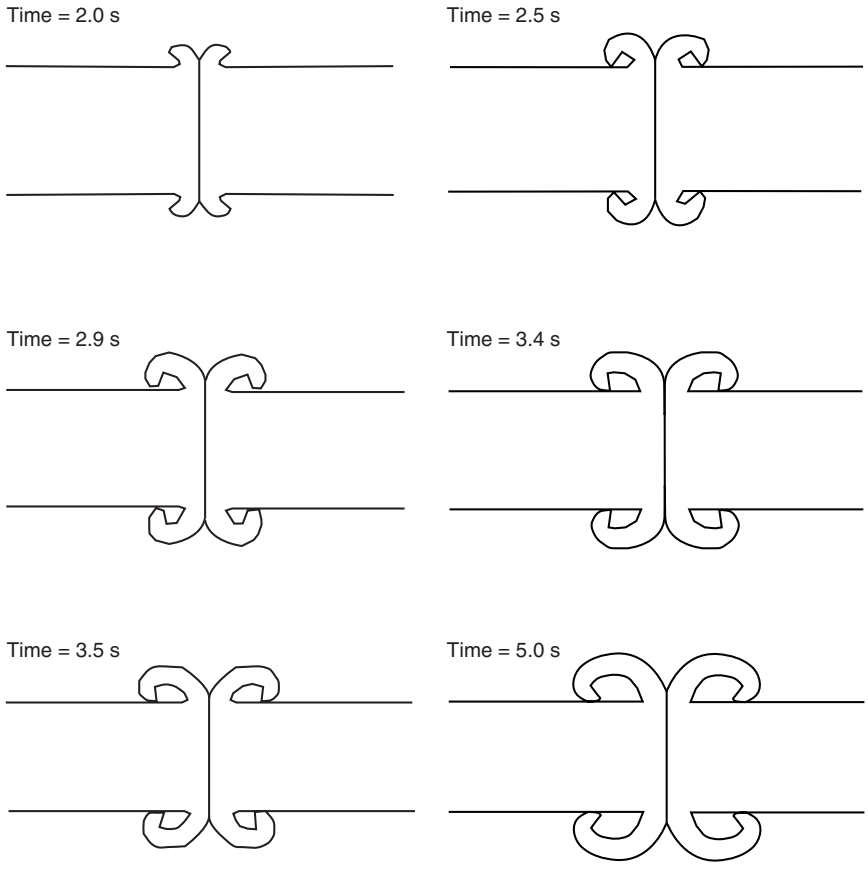


Figure 1.3.18–11 Flash prediction in current analysis.

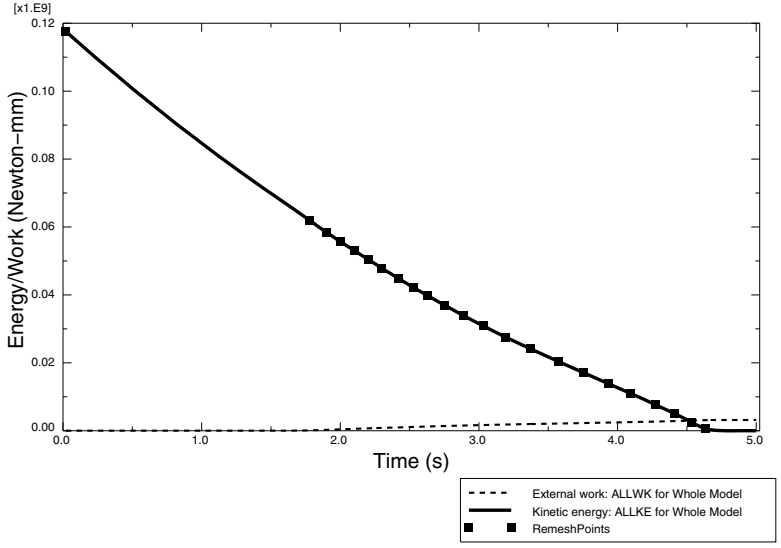


Figure 1.3.18–12 Kinetic energy history showing the flywheel slowdown.

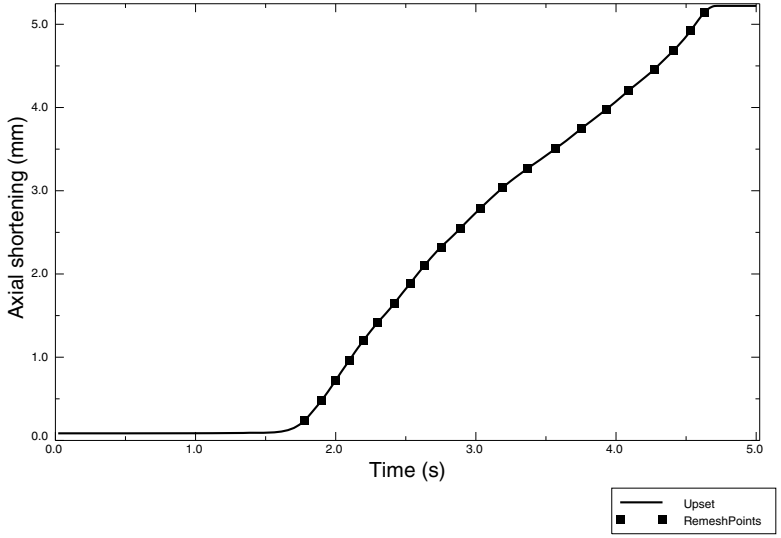


Figure 1.3.18–13 Assembly upset (shortening) history.

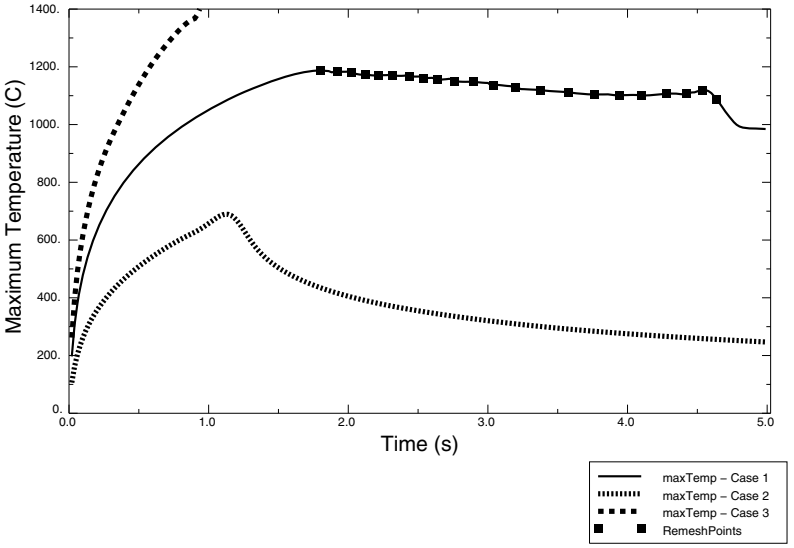


Figure 1.3.18–14 Maximum temperature history comparisons of the three cases. Case 1: Successful weld; Case 2: Failed weld with insufficient flywheel energy; Case 3: Failed weld with excessive flywheel energy.

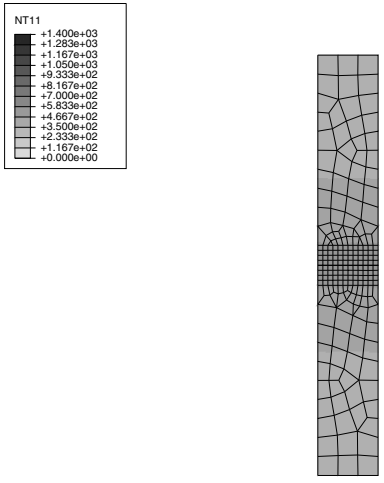


Figure 1.3.18–15 Temperature distribution at 5.0 sec for Case 2 (insufficient flywheel energy for welding).

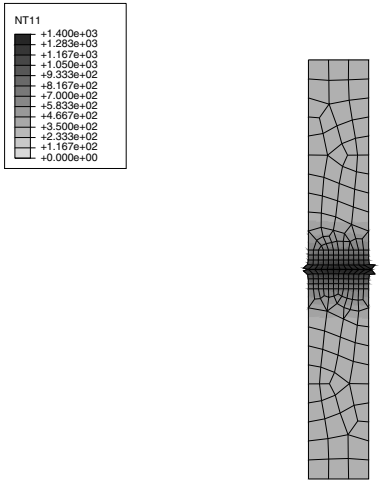


Figure 1.3.18–16 Temperature distribution at 0.9 sec for Case 3 (excessive initial flywheel energy results in a rapid temperature rise into the melting regime).

1.3.19 MOLDFLOW TRANSLATION EXAMPLES

Product: Abaqus/Standard

Objectives

This example demonstrates the following features of the **abaqus moldflow** translator:

- transforming finite element model information from a Moldflow analysis into a partial Abaqus input file,
- adding appropriate data for the analysis,
- performing natural frequency analyses, and
- calculating deformation due to initial stresses.

Application description

The results of a Moldflow simulation, which models the plastics injection mold-filling process, include calculations of material properties and residual stresses in the plastic part. Two types of parts, a filled bracket and an unfilled bracket, are studied in this example to illustrate how you can transform finite element model information from a Moldflow simulation into a partial Abaqus input file, adding boundary conditions and step data, and requesting output.

Abaqus modeling approaches and simulation techniques

This example illustrates the use of the **abaqus moldflow** translator.

Summary of analysis cases

Case 1 Natural frequency analysis of a fiber-filled bracket.

Case 2 Natural frequency analysis of an unfilled bracket.

Case 3 Deformation due to initial stresses in a three-dimensional filled bracket.

Run procedure

The three cases share the same general approach. The following procedure summarizes the typical usage of the **abaqus moldflow** translator:

1. Export the data from Moldflow simulation as follows:
   - For a midplane Moldflow simulation, export the finite element mesh data to a file named *job-name*`.pat` and the results data (material properties and residual stresses) to a file named *job-name*`.osp`.

- For a three-dimensional solid Moldflow simulation using Moldflow Version MPI 6, run the Visual Basic script **`mpi2abq.vbs`** to export the finite element mesh data to a file named *job-name*`_mesh.inp` and the results data to **`.xml`** files.

2. Run the **abaqus moldflow** translator to create a partial Abaqus input (**`.inp`**) file from the Moldflow interface files.
3. Edit the Abaqus input file to add appropriate data for the analysis (for example, add boundary conditions and step data).
4. Submit the Abaqus input file for analysis.

Extracting the files

All files associated with these analyses are included with the Abaqus release; you can use the Abaqus **fetch** utility to extract example problem files from the compressed archive files.

To extract all the relevant files for a particular example problem, enter the following commands:

Case 1 **`abaqus fetch job=moldflow_ex1*`**

Case 2 **`abaqus fetch job=moldflow_ex2*`**

Case 3 **`abaqus fetch job=bracket3d_mpi6*`**

For information on using wildcard expressions with the Abaqus **fetch** utility, see "Fetching sample input files," Section 3.2.16 of the Abaqus Analysis User's Guide.

Case 1: Natural frequency analysis of a fiber-filled bracket

The bracket in Case 1 consists of 926 nodes and 1719 S3R elements. The model contains seven different element sets. Each element set has a different thickness and is modeled as a laminated composite with 20 layers.

Ten unrestrained vibration modes are computed. The first six frequencies are approximately zero. The frequencies for the first four flexible modes are listed in Table 1.3.19–1.

The Abaqus finite element model is shown in Figure 1.3.19–1.

Case 2: Natural frequency analysis of an unfilled bracket

Case 2 uses the same Abaqus finite element model as Case 1, but the material properties are transversely isotropic. The shell section definition is homogeneous instead of composite. Twenty-one equally spaced Simpson integration points are used through the shell thickness.

The frequencies for the first four flexible vibration modes of the unfilled bracket are listed in Table 1.3.19–2. The unfilled material in this example is softer than the filled material in Case 1; consequently, the frequencies are lower.

Case 3: Deformation due to initial stresses in a three-dimensional filled bracket

Case 3 uses a solid Abaqus finite element model that is similar to the model used in Case 1.

To execute the **abaqus moldflow** translator, enter the following command:

```
abaqus moldflow job=bracket3d_mpi6 3D initial_stress=on
```

A contour plot of initial stresses is shown in Figure 1.3.19–2.

Input files

Case 1

moldflow_ex1.inp	Input file to analyze a fiber-filled bracket.

Case 2

moldflow_ex2.inp	Input file to analyze an unfilled bracket.

Reference

Abaqus Analysis User's Guide

- "Translating Moldflow data to Abaqus input files," Section 3.2.37 of the Abaqus Analysis User's Guide

Table 1.3.19–1 Frequencies for the first four flexible modes for the fiber-filled bracket.

Mode	Frequency, Hz
7	334
8	430
9	740
10	752

Table 1.3.19–2 Frequencies for the first four flexible modes for the unfilled bracket.

Mode	Frequency, Hz
7	146
8	217
9	363
10	371

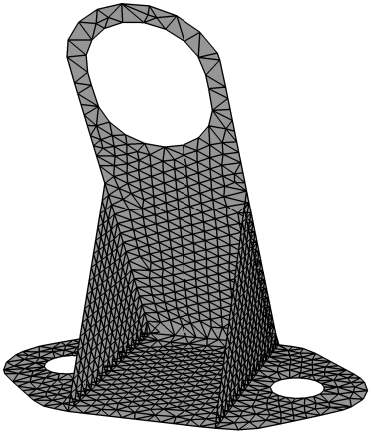


Figure 1.3.19–1 Finite element mesh of the fiber-filled bracket.

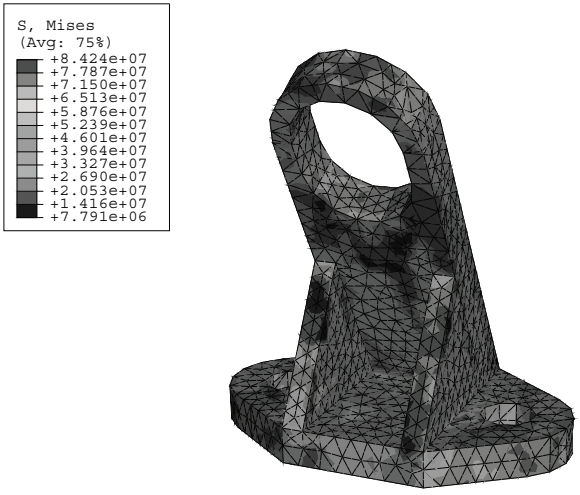


Figure 1.3.19–2 Contour plot of the initial stresses for the filled bracket using Moldflow Version MPI 6.

1.4 Fracture and damage

- "A plate with a part-through crack: elastic line spring modeling," Section 1.4.1
- "Contour integrals for a conical crack in a linear elastic infinite half space," Section 1.4.2
- "Elastic-plastic line spring modeling of a finite length cylinder with a part-through axial flaw," Section 1.4.3
- "Crack growth in a three-point bend specimen," Section 1.4.4
- "Analysis of skin-stiffener debonding under tension," Section 1.4.5
- "Failure of blunt notched fiber metal laminates," Section 1.4.6
- "Debonding behavior of a double cantilever beam," Section 1.4.7
- "Debonding behavior of a single leg bending specimen," Section 1.4.8
- "Postbuckling and growth of delaminations in composite panels," Section 1.4.9

1.4.1 A PLATE WITH A PART-THROUGH CRACK: ELASTIC LINE SPRING MODELING

Product: Abaqus/Standard

The line spring elements in Abaqus allow inexpensive evaluation of the effects of surface flaws in shell structures, with sufficient accuracy for use in design studies. The basic concept of these elements is that they introduce the local solution, dominated by the singularity at the crack tip, into a shell model of the uncracked geometry. The relative displacements and rotations across the cracked section, calculated in the line spring elements, are then used to determine the magnitude of the local strain field and hence the J-integral and stress intensity factor values, as functions of position along the crack front. This example illustrates the use of these elements and provides some verification of the results they provide by comparison with a published solution and also by making use of the shell-to-solid submodeling technique.

Problem description

A large plate with a symmetric, centrally located, semi-elliptic, part-through crack is subjected to edge tension and bending. The objective is to estimate the Mode I stress intensity factor, K_I, as a function of position along the crack front. Symmetry allows one quarter of the plate to be modeled, as shown in Figure 1.4.1–1. The 8-node shell element, S8R, and the corresponding 3-node (symmetry plane) line spring element LS3S are used in the model.

A mesh using LS6 elements is also included. Only half-symmetry is used in this case. When LS6 elements are used, the shell elements on either side of an LS6 element must be numbered such that the normals to these shell elements point in approximately the same direction.

Geometry and model

For each load case (tension and bending) two plate thicknesses are studied: a "thick" case, for which the plate thickness is 76.2 mm (3.0 in); and a "thin" case, for which the plate thickness is 19.05 mm (0.75 in). For both thicknesses the semi-elliptic crack has a maximum depth (a_0 in Figure 1.4.1–2) of 15.24 mm (0.6 in) and a half-length, c, of 76.2 mm (3.0 in). The plate is assumed to be square, with dimensions 609.6 × 609.6 mm (24 × 24 in).

The material is assumed to be linear elastic, with Young's modulus 207 GPa (30×10^6 lb/in^2) and Poisson's ratio 0.3.

A quarter of the plate is modeled, with symmetry along the edges of the quarter-model at $x = 0$ and $y = 0$. On the edge containing the flaw ($y = 0$), the symmetry boundary conditions are imposed only on the unflawed segment of the edge, since they are built into the symmetry plane of the line spring element being used (LS3S).

The loading consists of a uniform edge tension (per unit length) of 52.44 kN/m (300 lb/in) or a uniform edge moment (per unit length) of 1335 N-m/m (300 lb-in/in).

Results and discussion

The stress intensity factors for the thick and thin plates are compared with the detailed solutions of Raju and Newman (1979) and Newman and Raju (1979) in Figure 1.4.1–3 (tension load) and Figure 1.4.1–4 (bending load). These plots show that the present results agree reasonably well with those of Raju and Newman over the middle portion of the flaw ($\phi > 30°$), with better correlation being provided for the thick case, possibly because the crack is shallower in that geometry. The accuracy is probably adequate for basic assessment of the criticality of the flaw for design purposes. For values of ϕ less than about 30° (that is, at the ends of the flaw), the stress intensity values predicted by the line spring model lose accuracy. This accuracy loss arises from a combination of the relative coarseness of the mesh, (especially in this end region where the crack depth varies rapidly), as well as from theoretical considerations regarding the appropriateness of line spring modeling at the ends of the crack. These points are discussed in detail by Parks (1981) and Parks et al. (1981).

Shell-to-solid submodeling around the crack tip

An input file for the case a_0/t= 0.2, which uses the shell-to-solid submodeling capability, is included. This C3D20R element mesh allows the user to study the local crack area using the energy domain integral formulation for the *J*-integral. The submodel uses a focused mesh with four rows of elements around the crack tip. A $1/\sqrt{r}$ singularity is utilized at the crack tip, the correct singularity for a linear elastic solution. Symmetry boundary conditions are imposed on two edges of the submodel mesh, while results from the global shell analysis are interpolated to two edges by using the submodeling technique. The global shell mesh gives satisfactory *J*-integral results; hence, we assume that the displacements at the submodel boundary are sufficiently accurate to drive the deformation in the submodel. No attempt has been made to study the effect of making the submodel region larger or smaller. The submodel is shown superimposed on the global shell model in Figure 1.4.1–5.

The variations of the *J*-integral values along the crack in the submodeled analysis are compared to the line spring element analysis in Figure 1.4.1–3 (tension load) and Figure 1.4.1–4 (bending load). Excellent correlation is seen between the three solutions. A more refined mesh in the shell-to-solid submodel near the plate surface would be required to obtain *J*-integral values that more closely match the reference solution.

Input files

crackplate_ls3s.inp	LS3S elements.
crackplate_surfaceflaw.f	A small program used to create a data file containing the surface flaw depths.
crackplate_ls6_nosym.inp	LS6 elements without symmetry about $y = 0$.
crackplate_postoutput.inp	*POST OUTPUT analysis.
crackplate_submodel.inp	Shell-to-solid submodel.

References

- Newman, J. C., Jr., and I. S. Raju, "Analysis of Surface Cracks in Finite Plates Under Tension or Bending Loads," NASA Technical Paper 1578, National Aeronautics and Space Administration, December 1979.
- Parks, D. M., "The Inelastic Line Spring: Estimates of Elastic-Plastic Fracture Mechanics Parameters for Surface-Cracked Plates and Shells," Journal of Pressure Vessel Technology, vol. 13, pp. 246–254, 1981.
- Parks, D. M., R. R. Lockett, and J. R. Brockenbrough, "Stress Intensity Factors for Surface-Cracked Plates and Cylindrical Shells Using Line Spring Finite Elements," *Advances in Aerospace Structures and Materials*, Edited by S. S. Wang and W. J. Renton, ASME, AD–01, pp. 279–286, 1981.
- Raju, I. S., and J. C. Newman, Jr., "Stress Intensity Factors for a Wide Range of Semi-Elliptic Surface Cracks in Finite Thickness Plates," Journal of Engineering Fracture Mechanics, vol. 11, pp. 817–829, 1979.

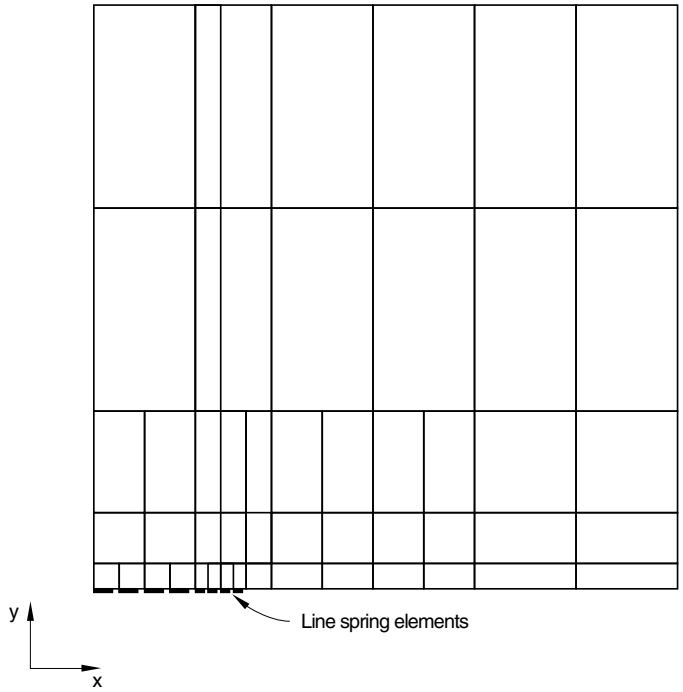
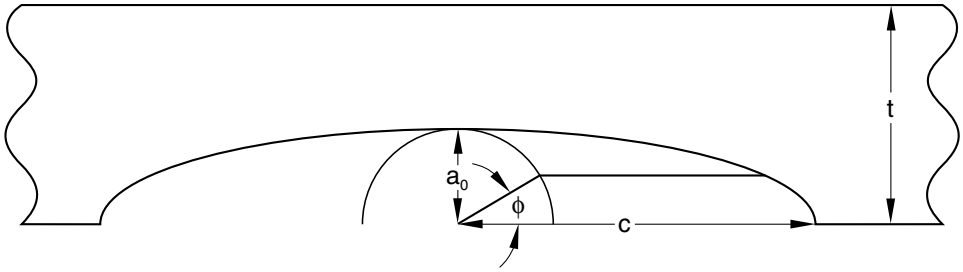


Figure 1.4.1–1 Quarter model of large plate with center surface crack.



a_0 = maximum flaw depth

2c = surface length of crack

t = shell thickness

ϕ = angle on an inscribed circle for locating a point on the crack

Figure 1.4.1–2 Schematic surface crack geometry for a semi-elliptical crack.

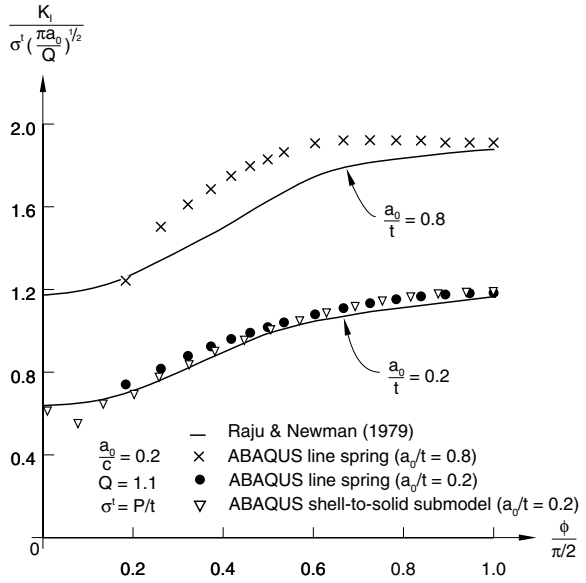


Figure 1.4.1–3 Stress intensity factor dependence on crack front position: tension loading.

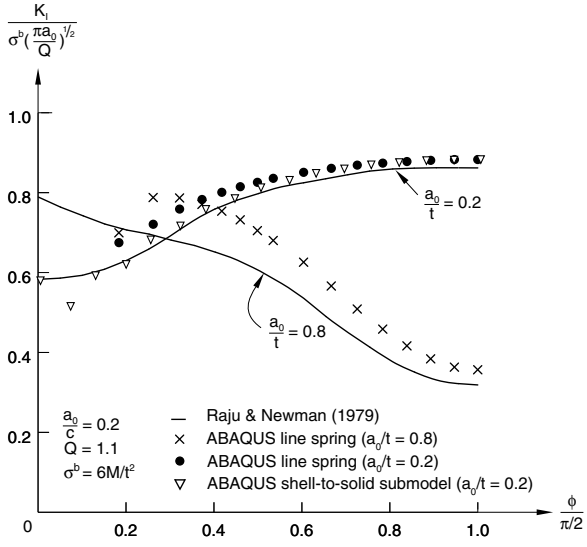


Figure 1.4.1–4 Stress intensity factor dependence on crack front position: moment loading.

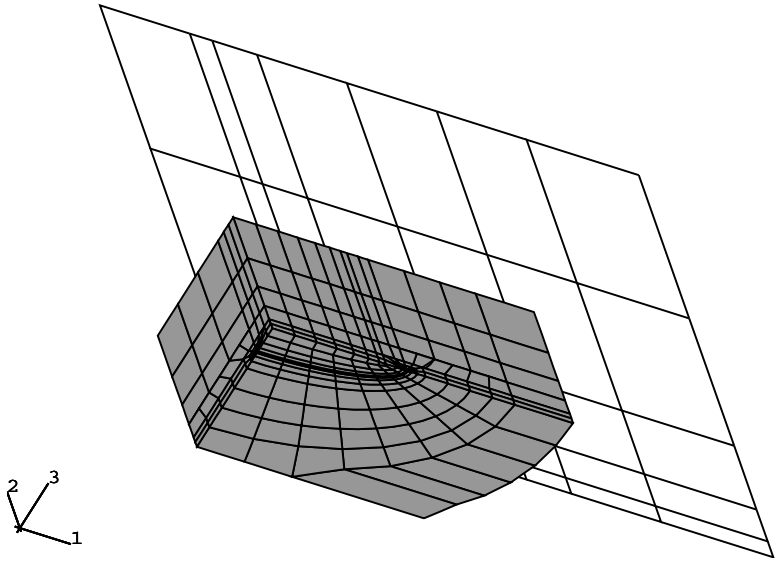


Figure 1.4.1–5 Solid submodel superimposed on shell global model.

1.4.2 CONTOUR INTEGRALS FOR A CONICAL CRACK IN A LINEAR ELASTIC INFINITE HALF SPACE

Products: Abaqus/Standard Abaqus/CAE

Objectives

This example demonstrates the following Abaqus features and techniques for linear elastic fracture mechanics:

- evaluating contour integrals for axisymmetric and three-dimensional fracture mechanics based on linear static stress analysis;
- partitioning steps required to generate a mesh suitable for evaluating contour integrals in two- and three-dimensional analyses;
- evaluating three-dimensional contour integrals when the crack extension direction varies along the crack front;
- node-based submodeling in fracture problems (comparing results for a single refined analysis with the submodeling approach);
- surface-based submodeling based on global model stresses, with guidelines for obtaining adequate accuracy; and
- applying continuum infinite elements simulating an infinite domain.

Application description

This example examines the fracture behavior of a conical crack, which may result from a small hard object impacting a large brittle body. It shows how to evaluate the propensity of the crack to propagate under static loading but does not cover the event that formed the crack.

The J-integral is a widely applied fracture mechanics parameter that relates to energy release associated with crack growth and is a measure of the deformation intensity at a crack tip. In practice, the calculated J-integral can be compared with a critical value for the material under consideration to predict fracture. The T-stress represents stress parallel to the crack face. Together, the T-stress and the J-integral provide a two-parameter fracture model describing Mode I elastic-plastic crack-tip stresses and deformation in plane strain or three dimensions over a wide range of crack configurations and loadings. The stress intensity factors, K_i, relate to the energy release rate and measure the propensity for crack propagation.

This example uses axisymmetric and three-dimensional models to demonstrate the Abaqus fracture mechanics capability, where the crack extension direction varies along a curved crack front. Submodeling and the use of infinite elements to simulate far-field boundaries are also demonstrated.

Geometry

The problem domain contains a conical crack in an infinite solid half-space, as shown in Figure 1.4.2–1. The crack extension direction changes as the crack is swept around a circle. The units for this example are nonphysical; therefore, dimensions, loads, and material properties are described in terms of length and force units. The crack circumscribes a circle with a radius of 10 length units on the free surface. The crack intersects the free surface at 45° and extends 15 length units into the solid domain.

Materials

The material is a linear elastic solid.

Boundary conditions and loading

The semi-infinite domain is constrained from rigid body motion. The applied load is a static pressure with a magnitude of 10 force/length^2 applied on the circular free surface of the block circumscribed by the crack. The loading is illustrated in Figure 1.4.2–1.

Abaqus modeling approaches and simulation techniques

This example includes six cases demonstrating different modeling approaches using Abaqus/Standard. The crack is modeled as a seam since the crack surfaces in the unloaded state lie next to one another with no gap.

The geometry is axisymmetric and can be modeled as such. However, the three-dimensional cases demonstrate the Abaqus fracture mechanics capability, where the crack extension direction varies along a curved crack front. The infinite half-space is treated using multiple techniques. In Case 1 through Case 4, the domain is extended well beyond the region of interest. Far-field boundary conditions applied a significant distance from the region of interest have negligible influence on the response near the crack. Cases 5 and Case 6 demonstrate the use of continuum infinite elements. Axisymmetric and three-dimensional cases are provided with and without submodeling. "Fracture mechanics," Section 11.4 of the Abaqus Analysis User's Guide, provides detailed information on fracture mechanics procedures.

Summary of analysis cases

Case 1 Full axisymmetric model using Abaqus/CAE.

Case 2 Full three-dimensional model using Abaqus/CAE.

Case 3 Axisymmetric approach with submodeling using Abaqus/CAE.

Case 4 Three-dimensional approach with submodeling using Abaqus/CAE.

Case 5 Axisymmetric approach with submodeling and infinite elements using input files.

Case 6 Three-dimensional approach with submodeling and infinite elements using input files.

The following sections discuss analysis considerations that are applicable to several or all the cases. More detailed descriptions are provided later including discussions of results and listings of files provided. The

models for Case 1 through Case 4 were generated using Abaqus/CAE. In addition to the Python scripts that generate the model databases, Abaqus/Standard input files are also provided for those cases.

Mesh design

The mesh includes a seam along the crack with duplicate nodes, which allow the crack to open when loaded. The geometry is partitioned to map rings of elements around the crack tip for the contour integral calculations. The models use either quadrilateral or brick elements with a collapsed side to create triangular elements for two-dimensional cases or wedge-shaped elements for three-dimensional cases, which introduce a singularity at the crack tip. To be used for the evaluation of contour integrals, the mesh around the crack tip must be modeled as described in "Using contour integrals to model fracture mechanics," Section 31.2 of the Abaqus/CAE User's Guide. In the axisymmetric cases a circular partition is created to mesh around the crack tip. In the three-dimensional cases the corresponding partition is a curved tubular volume enclosing the crack tip.

A refined mesh at the crack tip is required to obtain contour-independent results; i.e., there is no significant variation in the contour integral values calculated for successive rings of elements around the crack tip. In the circular partitioned region surrounding the crack tip where the contour integrals are calculated, the mesh should be biased moderately toward the crack tip. The accuracy of the contour integrals is not very sensitive to the biasing. Engineering judgment is required to establish adequate mesh refinement to produce contour-independent results while avoiding the possibility of creating elements at the crack tip that are so small in relation to other elements that they introduce numerical conditioning issues and associated round-off errors.

When the deformation and the material are linear as in this example, the diameter of the circular partition used to map the crack-tip mesh for contour integral calculations is not critical. (If the material is elastic-plastic, the size of the circular partition should generally contain the plastic zone and allow a number of the contours for the contour integrals to enclose the plastic zone while still remaining in the elastic region.) The remaining partitions are created so that the element shapes satisfy the element quality criteria in the regions away from the crack tip.

To understand the types of singularities created by collapsing the side of an element in two or three dimensions, see "Constructing a fracture mechanics mesh for small-strain analysis" in "Contour integral evaluation," Section 11.4.2 of the Abaqus Analysis User's Guide. In this application we want to have a square root singularity in strain at the crack since the material is linear elastic and we will perform a small-strain analysis.

The stress intensity factors and the T-stresses are calculated using the interaction integral method, in which auxiliary plane strain crack-tip fields are employed. The crack front radius of curvature is significant for this problem. Therefore, to calculate the contour integrals accurately for the three-dimensional cases, a very refined mesh is used to approach the plane strain condition locally around the crack front. This refined mesh makes the contour integral domain sufficiently small to minimize the influence of curvature on the results.

Additional details of the meshing procedures for the axisymmetric and three-dimensional cases are discussed below within the descriptions of the individual cases.

Material model

The linear static structural analysis requires specification of Young's modulus, which is 30,000,000 units of force/length2, and Poisson's ratio, which is 0.3. One solid, homogenous section is used to assign material properties to the elements.

Loading

A uniform pressure load of 10 units of force/length2 is applied along the free top surface of the crack. In the axisymmetric models the load region, where the pressure is applied is represented by a line segment. For the three-dimensional cases the load region where the pressure is applied an area.

Analysis steps

Each analysis is performed using a single linear static step.

Output requests

Output requests are used to specify calculation of contour integrals, stress intensity factors, and T-stress. See "Requesting contour integral output," Section 31.2.11 of the Abaqus/CAE User's Guide, for more information regarding fracture mechanics output. The global models used in the submodeling cases include output requests necessary to write displacement and stress results to the output database (`.odb`) file; in the case where node-based submodeling is used, displacement results are used to establish boundary conditions on the corresponding submodels. In the case where surface-based submodeling is used, stress results are used to establish boundary tractions on the corresponding submodel.

Submodeling

Realistic fracture analyses tend to require significant computer resources. To obtain accurate results when analyzing the stress field around a crack tip, a refined mesh must be used to capture the strong gradients near the tip. The required mesh refinement can make fracture mechanics models large since a crack is normally a very small feature compared with the model dimensions. An alternative technique that reduces computational resources is to use submodeling to obtain accurate results by running two smaller models sequentially instead of performing a single global analysis with a refined mesh around the crack. The first step is to solve a less refined global model to obtain a solution that is accurate away from the crack tip but is not sufficiently refined to capture strong gradients near the region of interest. A refined submodel of the crack-tip region is then used to obtain a more accurate solution and, hence, more accurate contour integrals. The boundaries of the submodel must be far enough from the region of interest that the less refined global model is able to provide accurate results at the submodel boundaries, particularly important when surface-based submodeling is used. This condition is verified during postprocessing by confirming that the stress contours at the boundaries of the submodel are similar to the stress contours at the same location in the global model.

Although the submodeling approach is not required for this example because the refined models for the entire domain analyzed in Case 1 and Case 2 are small enough to run on commonly available computing platforms, this application provides an opportunity to demonstrate submodeling techniques

for both axisymmetric and three-dimensional fracture mechanics cases, as well as showing the differences between node-based submodeling based on displacements and surface-based submodeling based on stresses. Submodeling procedures are described in detail in "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, and Chapter 38, "Submodeling," of the Abaqus/CAE User's Guide.

Modeling an infinite domain

Case 1 through Case 4 simulate the infinite extent of the domain with a continuum mesh that is large compared to the crack dimensions with appropriate far-field boundary conditions. In those cases the domain extends 20 times the crack length. Case 5 and Case 6 demonstrate the use of continuum infinite elements and represent the region of interest with reduced-integration continuum elements to a distance approximately 10 times the crack dimensions surrounded by a layer of continuum infinite elements. Far-field boundary conditions are not required in these cases.

Case 1 Full axisymmetric model with Abaqus/CAE

The axisymmetric domain is a solid with a radius equal to the height of 300 length units (see Figure 1.4.2–2). The top edge of the model represents the free surface containing the crack. The semi-infinite domain is simulated by extending the continuum model to a distance 20 times the length of the crack and applying appropriate far-field boundary conditions. This model uses continuum axisymmetric quadratic reduced-integration (CAX8R) elements.

Mesh design

When calculating contour integrals in two-dimensional problems, quadrilateral elements must be used around the crack tip where the contour integral calculations will be performed with triangular elements adjacent to the crack tip. These triangular elements are actually collapsed quadrilaterals, which introduce a singularity. The axisymmetric model must be partitioned as shown in Figure 1.4.2–2 to define the crack, introduce a singularity by collapsing elements at the crack tip, and create rings of quadrilateral elements for contour integral calculations. A straight line partition is created where the seam crack is defined along with a circular partition, which enables mapping rings of elements around the crack tip. When structured meshing is used for this partition, triangular elements are created adjacent to the crack tip with quadrilaterals surrounding them (see "Using contour integrals to model fracture mechanics," Section 31.2 of the Abaqus/CAE User's Guide). Abaqus/CAE automatically converts triangular elements at the crack tip to quadrilaterals with one side collapsed to introduce a singularity.

"Creating a seam," Section 31.1.2 of the Abaqus/CAE User's Guide, describes how to pick partition segments to define the seam (crack). Procedures to create a square root singularity in strain at the crack tip are described in "Controlling the singularity at the crack tip," Section 31.2.8 of the Abaqus/CAE User's Guide. After defining the seam, pick the crack tip to specify the region defining the first contour integral and define the q vector to specify the crack extension direction as described in "Creating a contour integral crack," Section 31.2.9 of the Abaqus/CAE User's Guide.

Boundary conditions

The right edge of the model shown in Figure 1.4.2–3 is unconstrained to represent the far-field boundary. The bottom edge of the model is constrained to zero displacement (U2) to eliminate rigid body motion while simulating the far-field boundary. These edges are far enough away from the area of interest around the crack to represent an infinite domain with negligible influence on the area of interest.

Run procedure

The Case 1 model is generated using Abaqus/CAE to create and to mesh native geometry. Python scripts are provided to automate building the model and running the solution. The scripts can be run interactively or from the command line.

To create the model interactively, start Abaqus/CAE and select **Run Script** from the **Start Session** dialog box that appears. Select the first file for the case, **`AxisymmConeCrack_model.py`**. When the script completes, you can use Abaqus/CAE commands to display and to query the model. When you are ready to analyze the model, select **Run Script** from the **File** menu and choose the next script, **`AxisymmConeCrack_job.py`**. The Python scripts provided allow you to modify the model interactively with Abaqus/CAE to explore additional variations on the cases provided here.

Alternatively, the Python scripts can be run from the command line with the Abaqus/CAE **`noGUI`** option in the order listed:

```
abaqus cae noGUI=filename.py
```

where **`abaqus`** is the system command to run the program and *filename* is the name of the script to be run.

As an alternative to the Python scripts, an Abaqus/Standard input file **`AxisymmConeCrack.inp`** is also provided to run this case. You can submit the analysis using the input file with the following command:

```
abaqus input=AxisymmConeCrack.inp
```

Case 2 Full three-dimensional model with Abaqus/CAE

The three-dimensional domain is a cube with an edge length of 300 units, as shown in Figure 1.4.2–4. The mesh represents a quarter-symmetric segment of the problem domain. The top of the model represents the free surface containing the crack. The semi-infinite domain is represented by extending the continuum mesh to a distance 20 times the length of the crack with appropriate symmetry and far-field boundary conditions.

Mesh design

When calculating three-dimensional contour integrals, rings of brick elements must be used around the crack tip where the contour integral calculations will be performed with wedge elements adjacent to the crack tip (these wedges are actually collapsed bricks). Concentric tubular partitions are created to map the mesh around the crack tip. The three-dimensional domain and partitioning of the geometry are illustrated in Figure 1.4.2–5 and Figure 1.4.2–6. The seam crack is shown in Figure 1.4.2–7. When

structured meshing is used for the inner tubular partition, wedge elements are created adjacent to the crack tip with rings of bricks surrounding them (see "Using contour integrals to model fracture mechanics," Section 31.2 of the Abaqus/CAE User's Guide, for details). A swept mesh used in the inner ring creates wedge elements at the crack tip. The outer ring is meshed with hexahedral elements using the structured meshing technique.

For details on how to define the crack propagation direction where the direction of the vectors varies along the crack front (referred to as the q vector), see "Defining the crack extension direction," Section 31.2.4 of the Abaqus/CAE User's Guide. Figure 1.4.2–8 illustrates the q vectors.

Boundary conditions

Symmetric displacement boundary conditions are applied normal to the symmetry planes. The far-field faces on the sides of the model, which are not symmetry planes, are unconstrained. The bottom face of the model is constrained from displacement in the direction of the pressure load (U2=0) to prevent rigid body motion while simulating a far-field boundary condition.

Run procedure

The Python scripts provided to generate the Abaqus/CAE model and to analyze the model are run following the same procedures as those described for Case 1.

As an alternative to the Python scripts, an Abaqus/Standard input file **SymmConeCrackOrphan.inp** is also provided to run this case. You can submit the analysis using the input file with the following command:

```
abaqus input= SymmConeCrackOrphan.inp
```

The files defining nodes and elements for this case, **SymmConeCrackOrphan_node.inp** and **SymmConeCrackOrphan_elem.inp**, must be available when the input file is submitted.

Figure 1.4.2–9 shows a deformed shape plot for the three-dimensional model of the crack from the full three-dimensional analysis of Case 2. The displacement is exaggerated using a scaling factor to visualize the crack opening.

Case 3 Axisymmetric approach with submodeling using Abaqus/CAE

Case 3 uses the submodeling approach and, hence, requires two sequential analyses, which are referred to as the global model and the submodel. First, a less refined global model is solved to obtain the displacement solution with sufficient accuracy away from the crack tip. A refined submodel of the area of interest driven by the displacement solution from the global model is then used to obtain an accurate solution in the crack-tip region. Each of these models is much smaller than the fully refined axisymmetric global model used in Case 1.

Mesh design

The axisymmetric global model has a relatively less refined mesh in the crack region. The global model used for Case 3 has two rings of elements where the mesh focuses on the crack tip, compared to 13 rings of elements around the crack tip in the full model used in Case 1.

The axisymmetric global model and the submodel meshes for Case 3 are shown in Figure 1.4.2–10. The axisymmetric submodel has a refined mesh around the crack tip with 12 rings of elements surrounding the crack tip. It is assumed that the global model's coarse mesh is sufficiently accurate to drive the submodel: the submodel can obtain accurate contour integral results if the global model's displacement field is accurate at the boundaries of the submodel, which lie far from the crack tip. You can verify this at the postprocessing stage by comparing stress contours at the boundaries of the submodel to the corresponding contours of the global model.

Boundary conditions

The boundary conditions applied to the axisymmetric global model are the same as those used in Case 1. The displacement solution from the global model is applied to the submodel boundaries when the submodeling technique is used.

Run procedure

The models used for Case 3 are generated using Abaqus/CAE to create and to mesh native geometry. The same procedures used to run the Python scripts for Cases 1 and 2 are used to create and to analyze the global model. The script that builds the submodel refers to the global model output database (**.odb**) file, which must be available when the submodel is analyzed. After analyzing the global model, run the scripts to build and to analyze the submodel using the same procedure used for the global model.

Abaqus/Standard input files are also provided to run this case. First, run the job to create and to analyze the global model; then run the submodel job. The results from the global model must be available to run the submodel. A typical execution procedure is as follows:

```
abaqus job=globalModelJobName  input=globalModelInputfile.inp
abaqus job=submodelJobname, input=subModelInputfile.inp,
globalmodel=globalModelJobNameOutputDatabase
```

Case 4 Three-dimensional approach with submodeling using Abaqus/CAE

Case 4 uses a three-dimensional submodeling approach and requires two sequential analyses, a global model and a submodel. First, a global model is solved with sufficient refinement to provide an accurate displacement and stress solution away from the crack tip.

Two versions of refined submodels of the area of interest are then used to obtain an accurate solution in the crack-tip region. In one submodel analysis the area of interest is driven by the displacement solution from the global model. In the other submodel analysis the area of interest is driven by the stress solution from the global model.

Each of the models used in this case is much smaller than the fully refined three-dimensional global model used in Case 2.

Mesh design

The three-dimensional global model, with a less refined mesh in the crack region, is first analyzed and then used to drive the submodel. For the three-dimensional global model only 18 elements are used

along the crack line, whereas 38 elements are used along the crack line in the submodel. Figure 1.4.2–11 shows the meshes for the three-dimensional global model and the submodel.

Boundary conditions

The boundary conditions applied to the global model in Case 4 are the same as those used in the full three-dimensional Case 2. The submodeling approach uses either the displacement or stress solution from the global model to drive the submodel boundaries.

Run procedure

The models used for Case 4 are generated using Abaqus/CAE to create and to mesh native geometry. The same procedures used to run Case 3 can be used with Case 4. The script that builds the submodel refers to the global model output database (**.odb**) file, which must be available when the submodel is analyzed.

As an alternative to the Python scripts, Abaqus/Standard input files are also provided to run this case. These are submitted using the same procedure described for the input files under Case 3.

Case 5 Axisymmetric submodeling approach with infinite elements using Abaqus/Standard input files

Case 5 uses the submodeling approach with an axisymmetric mesh utilizing continuum infinite elements to simulate the far-field boundary condition. The submodeling technique requires two sequential analyses, a less refined global model and a refined submodel at the crack tip.

The global model in Case 5 comprises an axisymmetric representation of the hemispherical domain with continuum elements to a radius of 170 length units. Eight-node biquadratic axisymmetric quadrilateral, reduced-integration elements (CAX8R) are used to model the solid domain in the region adjacent to the crack. The domain is further extended using a layer of continuum infinite elements to a radius of 340 length units. Five-node quadratic axisymmetric one-way infinite elements (CINAX5R) are used to simulate the far-field region of the solid. The submodel used in Case 5 does not encompass the complete crack face but extends to a distance far enough from the crack tip that strong variations in the stress field are captured within the submodel. This result can be verified by comparing stress contours of the submodel with the corresponding stress contours in the global model.

Mesh design

The axisymmetric global model, with a relatively less refined mesh in the crack region, is first analyzed and then used to drive the submodel. The axisymmetric global model and the submodel meshes for Case 5 are shown in Figure 1.4.2–12.

Boundary conditions

The continuum infinite elements eliminate the need for far-field constraints, which were required in Case 1 through Case 4.

Run procedure

The models used for Case 5 are generated using Abaqus/Standard input files. First, run the job to create and to analyze the global model; then run the submodel job. The results from the global model must be available to run the submodel. A typical execution procedure is as follows:

```
abaqus job=globalModelJobName  input=globalModelInputfile.inp
abaqus job=submodelJobname, input=subModelInputfile.inp,
globalmodel=globalModalJobNameOutputDatabase
```

Case 6 Three-dimensional approach with infinite elements using submodeling with Abaqus/Standard input files

Case 6 uses the submodeling approach with a three-dimensional mesh utilizing continuum infinite elements to simulate the far-field boundary condition. The domain modeled for Case 6 encompasses an eighth of a sphere, representing one-quarter of the semi-infinite problem domain. Continuum elements are used to a radius of 170 length units. The domain is extended using a layer of infinite continuum elements to a radius of 340 length units.

The submodeling technique requires two separate analyses, a less refined global model and a refined submodel at the crack tip. Case 6 uses Abaqus/Standard input files to generate the models rather than Abaqus/CAE Python scripts. Twenty-node quadratic, reduced-integration solid elements (C3D20R) are used to model the solid domain in the region adjacent to the crack. The domain is further extended using a layer of 12-node quadratic one-way infinite brick elements (CIN3D12R) to a radius of 340 length units.

The submodel used in Case 6 does not encompass the complete crack face but extends to a distance far enough from the crack tip that strong variations in the stress field are captured within the submodel.

Mesh design

The three-dimensional global model, with a relatively less refined mesh in the crack region, is first analyzed and then used to drive the submodel. The three-dimensional global model and the submodel meshes for Case 6 are shown in Figure 1.4.2–13.

Boundary conditions

The continuum infinite elements eliminate the need for far-field constraints, which were required in Case 1 through Case 4.

Run procedure

The models of Case 6 are generated using Abaqus/Standard input files. The same procedure used in Case 5 is also used in Case 6. The files containing the node and element definitions for the three-dimensional global model must be available when the input file is run to create the global model . The output database (**.odb**) file from the global model must be available to run the submodel.

Discussion of results and comparison of cases

Contour integral results obtained from the data (`.dat`) file for each case are summarized in Table 1.4.2–1 through Table 1.4.2–4. These results are also available from the output database (`.odb`) file by displaying history output in the Visualization module of Abaqus/CAE. While there is no analytical solution available for comparison, an additional axisymmetric analysis with extreme mesh refinement is used as the basis for a reference solution. Each table includes the reference solution value in the table title.

Abaqus calculates the J-integral using two methods. Values of the J-integral are based on the stress intensity factors, JK, and by evaluating the contour integral directly, JA. The stress intensity factors K_I and K_{II}, and the T-stresses are given in Table 1.4.2–2, Table 1.4.2–3, and Table 1.4.2–4, respectively. When the stress intensity factors are requested, Abaqus automatically outputs the J-integrals based on the stress intensity factors, JK. Values of K_{III} are not tabulated because these values should equal zero based on the loading and are negligibly small relative to K_I and K_{II}.

The tables list values for contour 1 through contour 5. Each contour corresponds to a successive ring of elements progressing outward radially from the crack tip. For the axisymmetric cases one set of results is available for each contour. For the three-dimensional cases Abaqus/Standard provides contour integral values at each crack-tip node. The values listed in Table 1.4.2–1 through Table 1.4.2–4 for the three-dimensional cases correspond to the location halfway along the circumference of the crack, which lies midway between the symmetry faces of the three-dimensional models. A detailed examination of the results for the three-dimensional cases confirms that the contour integral values are essentially constant at each node along the circumference of the crack tip. The exception is the value calculated for K_{III}; which fluctuates but remains small relative to K_I and K_{II} over almost the full length of the crack; however K_{III} increases at the open end faces of the crack corresponding to the symmetry planes. A loss of accuracy occurring at the node corresponding to the open end of a three-dimensional crack is a known limitation that can be expected when applying this method.

Results from the first contour are generally not used when evaluating fracture problems because the first contour is influenced by the singularity associated with the crack tip. The average quantities reported in the tabular results exclude the first contour. Comparisons refer only to contour 2 through contour 5. The axisymmetric and the three-dimensional modeling approaches are in close agreement, with and without submodeling. For each case, values of the tabulated quantities for J calculated by evaluating the contour integrals directly, (JA), K_I, K_{II}, and T-stresses deviate by less than 2% of the average of the corresponding values for contour 2 through contour 5. The J-integral for each case, calculated from the stress intensity factors (JK) deviate by less than 3.5% of the average of corresponding values. The larger deviation for JK versus JA is expected because the method of calculating contour integrals from the stress intensity factors (JK) is more sensitive to numerical precision than calculating the contour integrals directly (JA). K_{III} is analytically equal to zero due to the geometry and loading symmetry in this example; the numerical results for K_{III} are negligibly small relative to K_I and K_{II}.

Submodeling results

The global models used to calculate the deformation and stress fields that drive the submodels use crack-tip meshes that are too coarse to give accurate results for the contour integrals; therefore, the results for the global models are not tabulated. Results are tabulated for the submodels that refer to this global

analysis. Generally these results verify that the submodeling approach provides adequate accuracy in fracture problems where it may not be practical to use a sufficiently refined mesh in the crack-tip region of a global model. The node-based submodeling approach provides greater accuracy than the surface-based approach.

Node-based submodeling results

The *J*-integral values for the node-based submodel analyses match those for the full model analyses (analyses with adequate mesh refinement around the crack tip) to within less than 1%.

Surface-based submodeling results

The three-dimensional submodeling case also considers surface-based submodeling, where the submodel is driven by the global model stress field. Two different pairs of global models and surface-based submodels are considered: one that matches the mesh design used in the node-based analysis, and one where adjustments are made to improve accuracy. The *J*-integral values for the first analysis pair, with the same meshes as in node-based submodeling, match those for the full model only to within 6%. These inaccurate results arise from a modeling arrangement that violates guidelines established in "Surface-based submodeling," Section 10.2.3 of the Abaqus Analysis User's Guide, namely that

- the submodel surface should intersect the global model in regions of relatively low stress gradients, and
- the submodel surface should intersect the global model in regions of uniform element size.

Adjusted global and submodel analyses that adhere to these guidelines are run. In this case the submodel driven surface is farther from the crack region and the high stress gradient, and the global model mesh is refined so that elements are more uniform in the region of the submodel surface. Figure 1.4.2–14 shows a comparison of the submodel/global model pairs. The modeling arrangement on the left places the lower submodel boundary too near to the crack and high stress gradients and cuts through high aspect ratio elements. The arrangement on the right provides lower aspect ratio elements in the global model and positions the lower submodel boundary further from the crack. The adjusted analysis with the further boundary now matches the *J*-integral values for the full model to within 2%. This accuracy difference illustrates the importance of adhering to the guidelines for surface-based submodel design. In practice, in the absence of a reference global solution, you should use the following guidelines to ensure your surface-based solution is adequate:

- As with any submodel analysis, compare solution results between the global model and submodel on the submodel boundary. In this case a stress comparison is appropriate. Figure 1.4.2–15 compares the 2-component of stress for the two surface-based submodel analyses and their corresponding global model. Results are plotted on a path lying in the lower submodel boundary and extending from the center radially outward. The near-boundary submodel has a significantly greater stress discrepancy with the global model.
- In cases where inertia relief is employed to address rigid body modes in surface-based submodeling, if the inertia relief force output variable (IRF) is small compared to the prevailing force level in the model, the surface-based stress distribution is equilibrated. In this model the prevailing force is the

10 units of pressure acting on the surface circumscribed on the crack (a radius of 6), or 786 units of force for the three-dimensional quarter symmetry model.

In this analysis the inertia relief force in the 2-direction is similar in both cases (33 for the near-boundary model and 32 for the far-boundary model) and relatively small; hence, in this case, the inertia relief force would not suggest poorer results with the near-boundary submodel, and its small value is not a sufficient measure of the adequacy of the submodel design.

Files

You can use the Python scripts for Abaqus/CAE and input files for Abaqus/Standard to create and to run the cases.

Case 1 Full axisymmetric analysis

AxisymmConeCrack_model.py	Script to create the model, including instructions for creating the mesh used for the reference solution.
AxisymmConeCrack_job.py	Script to analyze the model.
AxisymmConeCrack.inp	Input file to create and to analyze the model.

Case 2 Full three-dimensional model

SymmConeCrack_model.py	Script to create the model.
SymmConeCrack_job.py	Script to analyze the model.
SymmConeCrackOrphan.inp	Input file to create and to analyze the model.
SymmConeCrackOrphan_node.inp	Nodes for SymmConeCrackOrphan.inp.
SymmConeCrackOrphan_elem.inp	Elements for SymmConeCrackOrphan.inp.

Case 3 Axisymmetric submodel analysis

AxisymmConeCrackGl_model.py	Script to create the model.
AxisymmConeCrackGl_job.py	Script to analyze the model and to create the output database file that drives the submodel.
AxisymmConeCrackSub_model.py	Script to create the submodel.
AxisymmConeCrackSub_job.py	Script to analyze the submodel using the results from the global model output database file to drive it.
AxisymmConeCrackGl.inp	Input file to create and to analyze the global model.
AxisymmConeCrackSub.inp	Input file to create and to analyze the submodel.

Case 4 Three-dimensional submodel analysis

SymmConeCrackGl_model.py	Script to create the global model.
SymmConeCrackGl_job.py	Script to analyze the global model and to create the output database file that drives the submodel. Refer to parameter definitions in the script to create the adjusted global model referred to in “Submodeling results.”
SymmConeCrackSub_model.py	Script to create the node-based submodel.

SymmConeCrackSub_job.py	Script to analyze the node-based submodel using the results from the global model output database file to drive it.
SymmConeCrackSubSb_near_model.py	Script to create the surface-based submodel.
SymmConeCrackSubSb_near_job.py	Script to analyze the surface-based submodel using the stress results from the global model output database file to drive it.
SymmConeCrackSubSb_far_model.py	Script to create the surface-based submodel with a far-boundary submodel.
SymmConeCrackSubSb_far_job.py	Script to analyze the surface-based submodel with the far-boundary submodel using the stress results from the global model output database file to drive it.
SymmConeCrackGlOrphan.inp	Input file to create and to analyze the global model.
SymmConeCrackGlOrphan_node.inp	Nodes for SymmConeCrackGlOrphan.inp.
SymmConeCrackGlOrphan_elem.inp	Elements for SymmConeCrackGlOrphan.inp.
SymmConeCrackGlOrphanAdj.inp	Input file to create and to analyze the global model that is adjusted for improved surface-based submodel accuracy.
SymmConeCrackGlOrphanAdj_node.inp	Nodes for SymmConeCrackGlOrphanAdj.inp.
SymmConeCrackGlOrphanAdj_elem.inp	Elements for SymmConeCrackGlOrphanAdj.inp.
SymmConeCrackSubOr.inp	Input file to create and to analyze the node-based submodel.
SymmConeCrackSubOr_node.inp	Nodes for SymmConeCrackSubOr.inp.
SymmConeCrackSubOr_elem.inp	Elements for SymmConeCrackSubOr.inp.
SymmConeCrackSubOrSb_near.inp	Input file to create and to analyze the submodel using the surface-based submodel technique to drive the submodel stresses.
SymmConeCrackSubOrSb_near_node.inp	Nodes for SymmConeCrackSubOrSb_near.inp.
SymmConeCrackSubOrSb_near_elem.inp	Elements for SymmConeCrackSubOrSb_near.inp.
SymmConeCrackSubOrSb_far.inp	Input file to create and to analyze the submodel with the far-boundary submodel using the surface-based submodel technique to drive the submodel stresses.
SymmConeCrackSubOrSb_far_node.inp	Nodes for SymmConeCrackSubOrSb_far.inp.
SymmConeCrackSubOrSb_far_elem.inp	Elements for SymmConeCrackSubOrSb_far.inp.

Case 5 Axisymmetric submodel analysis using infinite continuum elements

conicalcrack_axiglobal.inp	Input file to analyze the axisymmetric global model and to create the output database file that drives the submodel.
conicalcrack_axisubmodel_rms.inp	Input file to analyze the axisymmetric submodel using the results from the global model output database file to drive it.

Case 6 Three-dimensional submodeling analysis using infinite continuum elements

conicalcrack_3dglobal.inp	Input file to analyze the three-dimensional global model and to create the output database file that drives the submodel.
conicalcrack_3dsubmodel_rms.inp	Input file to analyze the three-dimensional submodel using the results from the global model output database file to drive it.

Reference

Other

- 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.

Table 1.4.2–1 J-integral estimates ($\times 10^{-7}$) for conical crack using Abaqus. JK denotes the J values estimated from stress intensity factors; JA denotes the J values estimated directly by Abaqus. The reference solution J-integral value is 1.33.

Solution	Contour						Average Value, Contours 2–5
	J estimate method	1	2	3	4	5	
Case 1: Full axisymmetric	**JK**	1.326	1.308	1.288	1.262	1.228	1.272
	JA	1.334	1.333	1.334	1.334	1.334	1.334
Case 2: Full three-dimensional	**JK**	1.303	1.325	1.312	1.295	1.274	1.302
	JA	1.308	1.334	1.336	1.337	1.337	1.336
Case 3: Submodel axisymmetric	**JK**	1.327	1.319	1.311	1.300	1.287	1.304
	JA	1.330	1.329	1.330	1.330	1.330	1.330
Case 4: Node-based submodel three-dimensional	**JK**	1.314	1.316	1.303	1.285	1.264	1.292
	JA	1.318	1.326	1.328	1.328	1.328	1.328
Case 4: Surface-based submodel three-dimensional	**JK**	1.396	1.398	1.385	1.367	1.345	1.374
	JA	1.400	1.408	1.409	1.408	1.407	1.408
Case 4: Surface-based submodel with far boundary, three-dimensional	**JK**	1.345	1.347	1.335	1.317	1.296	1.324
	JA	1.349	1.357	1.359	1.358	1.358	1.358
Case 5: Submodel axisymmetric with infinite elements	**JK**	1.413	1.359	1.363	1.363	1.361	1.362
	JA	1.407	1.360	1.365	1.365	1.365	1.364

Solution	Contour						Average Value, Contours 2–5
	J estimate method	1	2	3	4	5	
Case 6: Submodel three-dimensional with infinite elements	**JK**	1.329	1.363	1.367	1.368	1.368	1.367
	JA	1.336	1.361	1.366	1.366	1.366	1.365

Table 1.4.2–2 Stress intensity factor K_I estimates for conical crack using Abaqus. Contour 1 is omitted from the average value calculations. The reference solution K_I value is 0.491.

Solution	Contour					Average Value, Contours 2–5
	1	2	3	4	5	
Case 1: Full axisymmetric	0.495	0.497	0.499	0.500	0.499	0.499
Case 2: Full three-dimensional	0.492	0.501	0.503	0.502	0.500	0.502
Case 3: Submodel axisymmetric	0.491	0.493	0.494	0.495	0.496	0.494
Case 4: Node-based submodel three-dimensional	0.491	0.496	0.498	0.497	0.494	0.497
Case 4: Surface-based submodel three-dimensional	0.426	0.431	0.433	0.431	0.427	0.430

Solution	Contour					Average Value, Contours 2–5
	1	2	3	4	5	
Case 4: Surface-based submodel with far boundary, three-dimensional	0.436	0.441	0.443	0.442	0.439	0.441
Case 5: Submodel axisymmetric with infinite elements	0.537	0.527	0.528	0.528	0.529	0.528
Case 6: Submodel three dimensional with infinite elements	0.522	0.528	0.529	0.530	0.530	0.528

Table 1.4.2–3 Stress intensity factor K_{II} estimates for conical crack using Abaqus. Contour 1 is omitted from the average value calculations. The reference solution K_{II} value is −2.03.

Solution	Contour					Average Value, Contours 2–5
	1	2	3	4	5	
Case 1: Full axisymmetric	–2.032	–2.016	–2.000	–1.978	–1.949	–1.986
Case 2: Full three-dimensional	–2.013	–2.029	–2.018	–2.004	–1.987	–2.010
Case 3: Submodel axisymmetric	–2.033	–2.026	–2.019	–2.010	–1.999	–2.014

Solution	Contour					Average Value, Contours 2–5
	1	2	3	4	5	
Case 4: Node-based submodel three-dimensional	–2.023	–2.023	–2.012	–1.997	–1.980	–2.003
Case 4: Surface-based submodel three-dimensional	–2.102	–2.103	–2.092	–2.078	–2.061	–2.084
Case 4: Surface-based submodel with far boundary, three-dimensional	–2.060	–2.060	–2.050	–2.036	–2.019	–2.041
Case 5: Submodel axisymmetric with infinite elements	–2.090	–2.050	–2.053	–2.052	–2.051	–2.051
Case 6: Submodel three dimensional with infinite elements	2.027	2.053	2.057	2.057	2.057	2.056

Table 1.4.2–4 T-stress estimates for conical crack using Abaqus. Contour 1 is omitted from the average value calculations. The reference solution T-stress value is 0.979.

Solution	Contour					Average Value, Contours 2–5
	1	2	3	4	5	
Case 1: Full axisymmetric	–0.982	–0.979	–0.976	–0.972	–0.967	–0.973
Case 2: Full three-dimensional	–0.942	–0.972	–0.966	–0.960	–0.954	–0.963
Case 3: Submodel Axisymmetric	–0.980	–0.978	–0.977	–0.975	–0.973	–0.976
Case 4: Node-based submodel three-dimensional	–0.947	–0.966	–0.959	–0.953	–0.947	–0.956
Case 4: Surface-based submodel three-dimensional	–0.981	–0.996	–0.989	–0.983	–0.976	–0.986
Case 4: Surface-based submodel with far boundary, three-dimensional	–0.958	–0.973	–0.966	–0.960	–0.954	–0.963
Case 5: Submodel axisymmetric with infinite elements	–1.182	–0.983	–0.985	–0.984	–0.984	–0.984
Case 6: Submodel three-dimensional with infinite elements	–0.599	–0.982	–0.984	–0.983	–0.982	–0.982

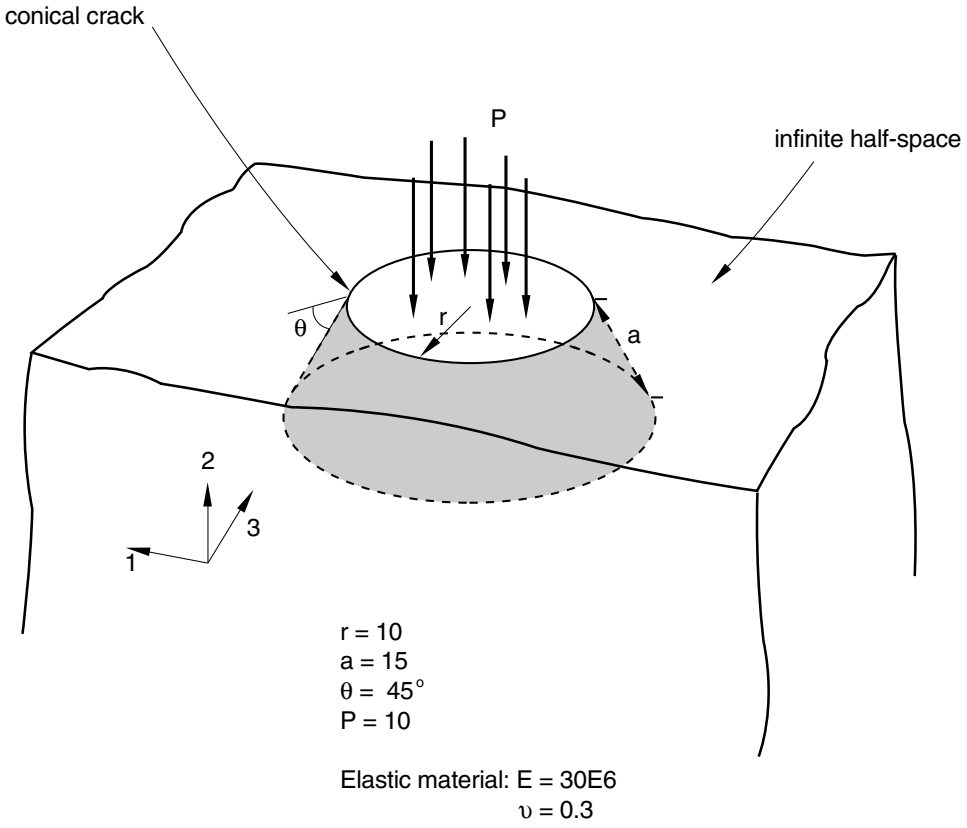


Figure 1.4.2–1 Conical crack in a half-space.

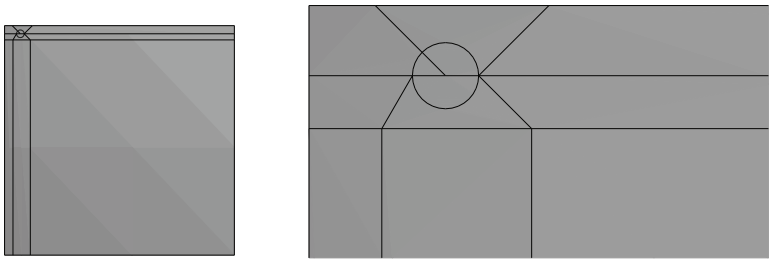


Figure 1.4.2–2 Case 1: Partitioning axisymmetric geometry.

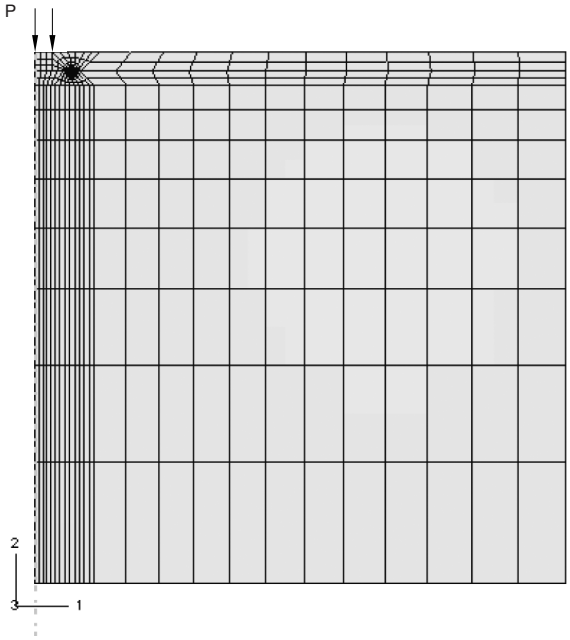


Figure 1.4.2–3 Case 1: Full axisymmetric mesh.

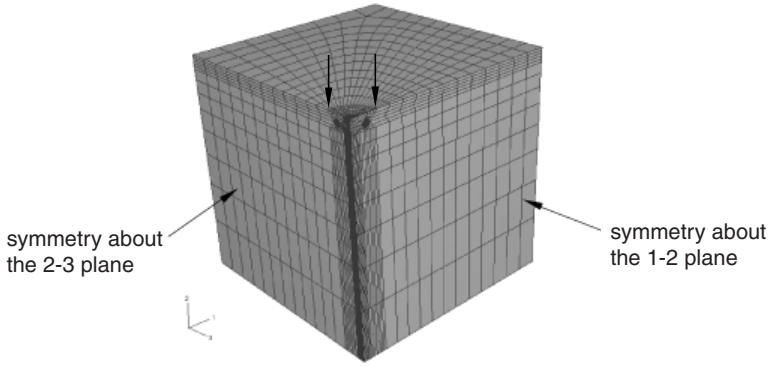


Figure 1.4.2–4 Case 2: Full three-dimensional mesh.

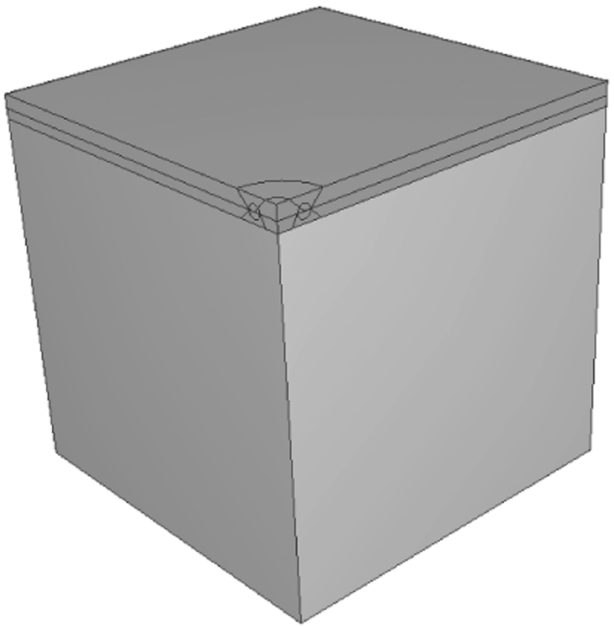


Figure 1.4.2–5 Case 2: Partitioned full three-dimensional model.

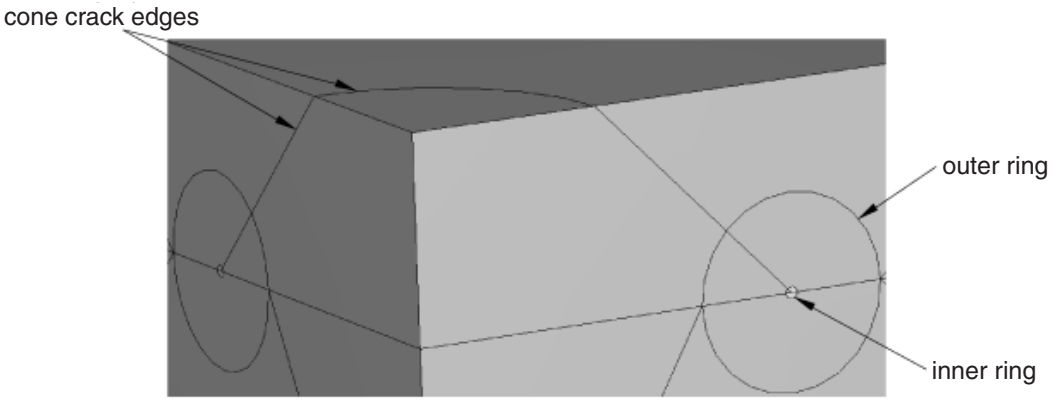


Figure 1.4.2–6 Case 2: Partitions around the crack line. The smaller inner ring is swept meshed using wedge elements. The outer ring is meshed using hexahedral elements and the structured meshing technique. The cone partitions are also visible.

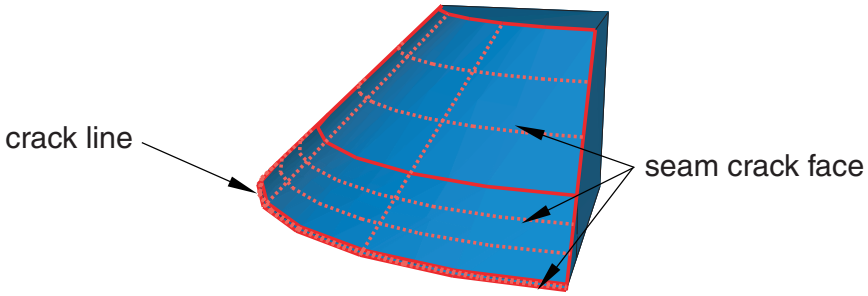


Figure 1.4.2–7 Case 2: Seam crack faces for the cone.

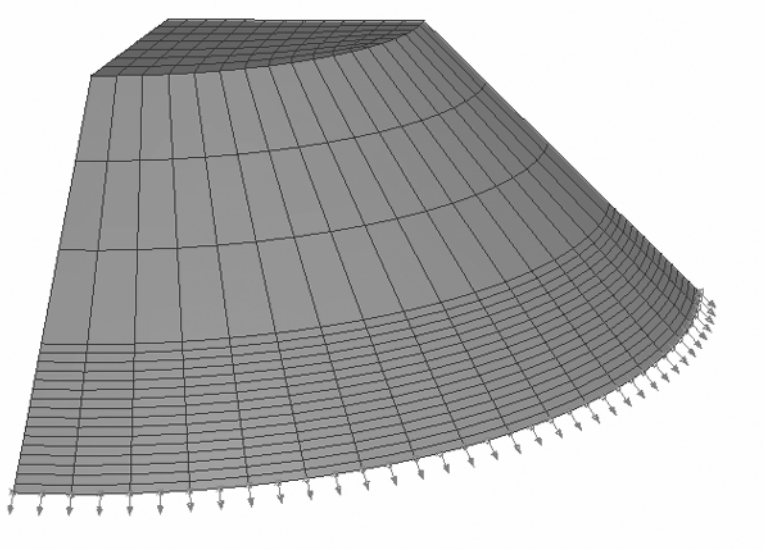


Figure 1.4.2–8 Case 2: q vectors defined along the entire crack line on an orphan mesh.

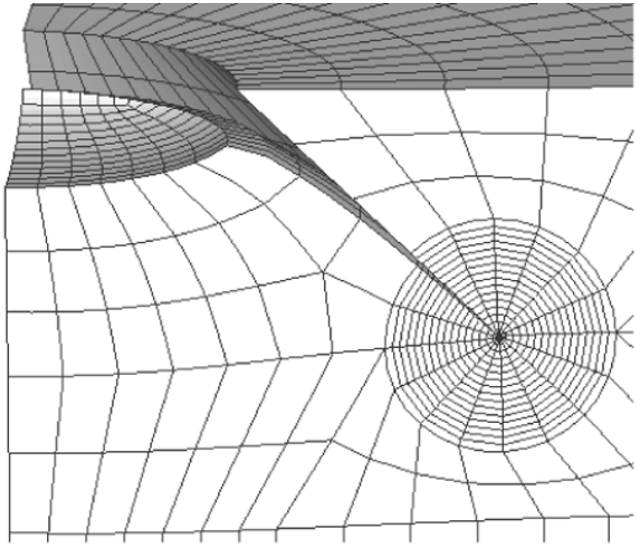


Figure 1.4.2–9 Case 2: Results three-dimensional displaced shape.

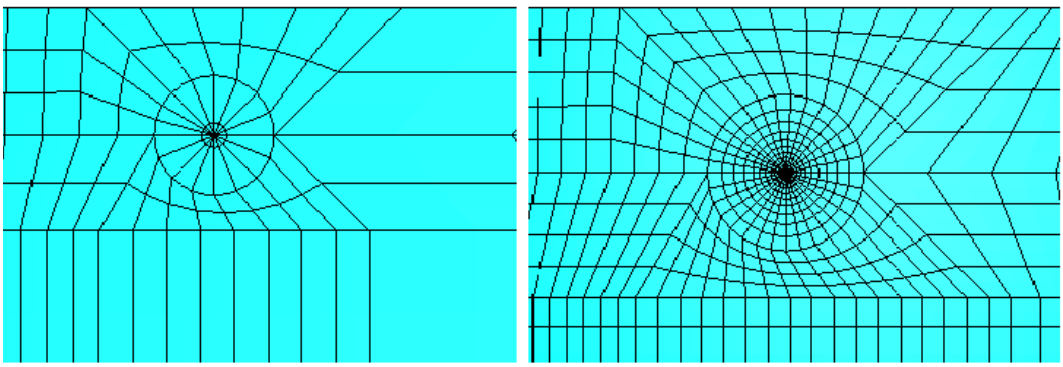


Figure 1.4.2–10 Case 3: Axisymmetric global and submodel meshes around the crack line.

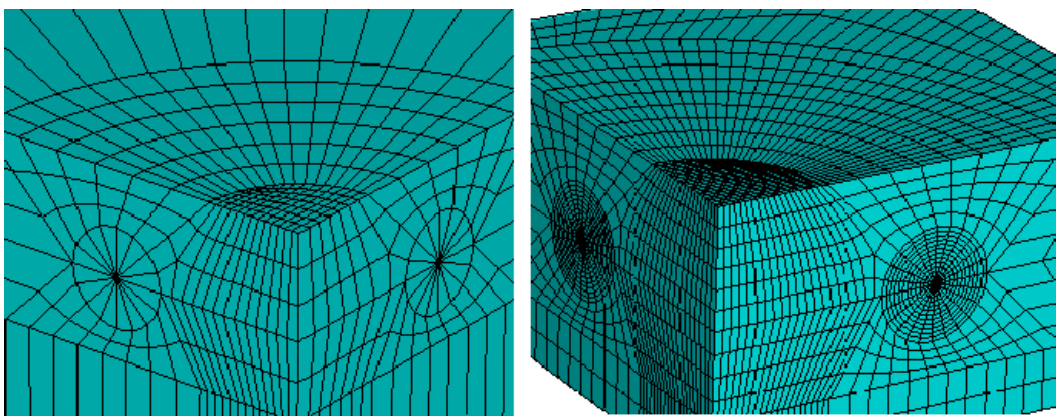


Figure 1.4.2–11 Case 4: Full three-dimensional global model and submodel meshes around the crack line.

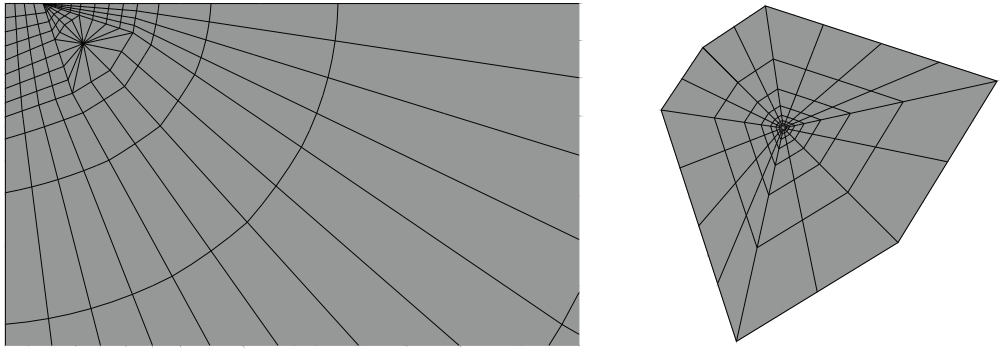


Figure 1.4.2–12 Case 5: Axisymmetric global model using infinite elements and submodel meshes.

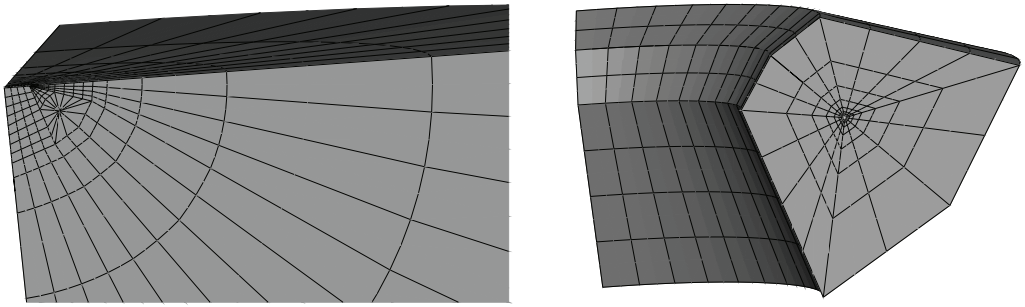


Figure 1.4.2–13 Case 6: Three-dimensional global model with infinite elements and submodel meshes.

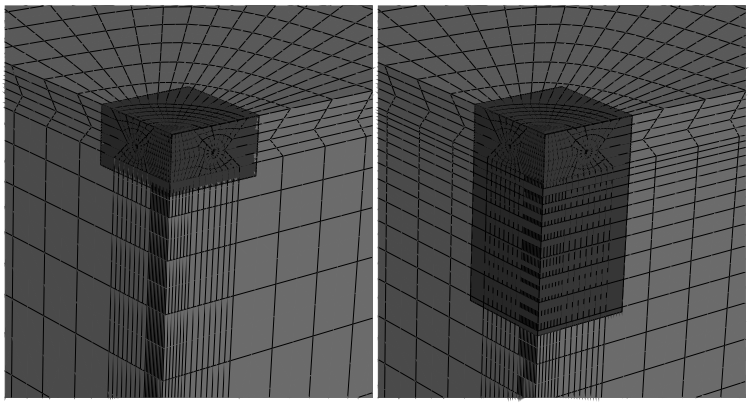


Figure 1.4.2–14 Case 4: Comparison of inadequate (left) and adequate (right) global and submodel designs for a surface-based submodel stress solution.

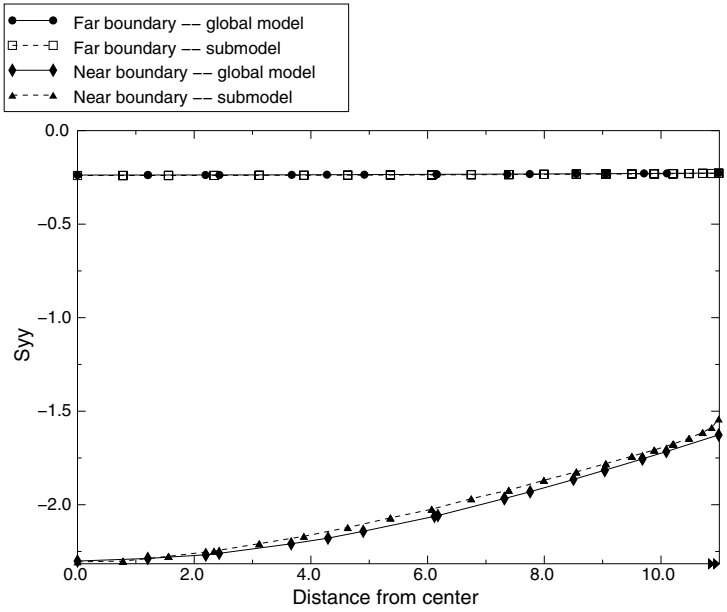


Figure 1.4.2–15 Case 4: Confirmation of stress agreement between the global model and submodel.

1.4.3 ELASTIC-PLASTIC LINE SPRING MODELING OF A FINITE LENGTH CYLINDER WITH A PART-THROUGH AXIAL FLAW

Product: Abaqus/Standard

The elastic-plastic line spring elements in Abaqus are intended to provide inexpensive solutions for problems involving part-through surface cracks in shell structures loaded predominantly in Mode I by combined membrane and bending action in cases where it is important to include the effects of inelastic deformation. This example illustrates the use of these elements. The case considered is a long cylinder with an axial flaw in its inside surface, subjected to internal pressure. It is taken from the paper by Parks and White (1982).

When the line spring element model reaches theoretical limitations, the shell-to-solid submodeling technique is utilized to provide accurate J-integral results. The energy domain integral is used to evaluate the J-integral for this case.

Geometry and model

The cylinder has an inside radius of 254 mm (10 in), wall thickness of 25.4 mm (1 in), and is assumed to be very long. The mesh is shown in Figure 1.4.3–1. It is refined around the crack by using multi-point constraints (MPCs). There are 70 shell elements of type S8R in the symmetric quarter-model and eight symmetric line spring elements (type LS3S) along the crack. The mesh is taken from Parks and White, who suggest that this mesh is adequately convergent with respect to the fracture parameters (J-integral values) that are the primary objective of the analysis. No independent mesh studies have been done. The use of MPCs to refine a mesh of reduced integration shell elements (such as S8R) is generally satisfactory in relatively thick shells as in this case. However, it is not recommended for thin shells because it introduces constraints that "lock" the response in the finer mesh regions. In a thin shell case the finer mesh would have to be carried out well away from the region of high strain gradients.

Three different flaws are studied. All have the semi-elliptic geometry shown in Figure 1.4.3–2, with, in all cases, $c = 3a_0$. The three flaws have a_0/t ratios of 0.25 (a shallow crack), 0.5, and 0.8 (a deep crack). In all cases the axial length of the cylinder is taken as 14 times the crack half-length, c: this is assumed to be sufficient to approximate the infinite length.

An input data file for the case $a_0/t = .5$ without making the symmetry assumption about $z = 0$ is also included. This mesh uses the LS6 line spring elements and serves to check the elastic-plastic capability of the LS6 elements. The results are the same as for the corresponding mesh using LS3S elements and symmetry about $z = 0$. The formulation of the LS6 elements assumes that the plasticity is predominately due to Mode I deformation around the flaw and neglects the effect of the Mode II and Mode III deformation around the flaw. In the global mesh the displacement in the z-direction is constrained to be zero at the node at the end of the flaw where the flaw depth goes to zero. To duplicate this constraint in the mesh using LS6 elements, the two nodes at the end of the flaw (flaw depth = 0) are constrained to have the same displacements.

Material

The cylinder is assumed to be made of an elastic-plastic metal, with a Young's modulus of 206.8 GPa (30×10^6 lb/in^2), a Poisson's ratio of 0.3, an initial yield stress of 482.5 MPa (70000 lb/in^2, and constant work hardening to an ultimate stress of 689.4 MPa (10^5 lb/in^2) at 10% plastic strain, with perfectly plastic behavior at higher strains.

Loading

The loading consists of uniform internal pressure applied to all of the shell elements, with edge loads applied to the far end of the cylinder to provide the axial stress corresponding to a closed-end condition. Even though the flaw is on the inside surface of the cylinder, the pressure is not applied on the exposed crack face. Since pressure loads on the flaw surface of line spring elements are implemented using linear superposition in Abaqus, there is no theoretical basis for applying these loads when nonlinearities are present. We assume that this is not a large effect in this problem. For consistency with the line spring element models, pressure loading of the crack face is not applied to the shell-to-solid submodel.

Results and discussion

The line spring elements provide ***J***-integral values directly. Figure 1.4.3–3 shows the J-integral values at the center of the crack as functions of applied pressure for the three flaws. In the input data the maximum time increment size has been limited so that adequately smooth graphs can be obtained. Figure 1.4.3–4 shows the variations of the J-integral values along the crack for the half-thickness crack ($a_0/t = 0.5$), at several different pressure levels (a normalized pressure, $\hat{p} = p\bar{R}/\sigma_y t$, is used, where $\bar{R}$ is the mean radius of the cylinder). These results all agree closely with those reported by Parks and White (1982), where the authors state that these results are also confirmed by other work. In the region $\phi < 30°$ the results are inaccurate for two reasons. First, the depth of the flaw is changing very rapidly in this region, which makes the line spring approximation quite inaccurate. Second, J^{el} is of the same order of magnitude as J^{pl}, but the line spring plasticity model is only valid when $J^{el} << J^{pl}$. The results toward the center of the crack ($\phi > 30°$) are more accurate than those at the ends of the crack since the flaw depth changes less rapidly with position in this region and J^{pl} is much larger than J^{el}. For this reason only ***J*** values for $\phi > 30°$ are shown in Figure 1.4.3–4.

Shell-to-solid submodeling around the crack tip

An input file for the case $a_0/t = 0.25$, which uses the shell-to-solid submodeling capability, is included. This C3D20R element mesh allows the user to study the local crack area using the energy domain integral formulation for the J-integral. The submodel uses a focused mesh with four rows of elements around the crack tip. A $1/\boldsymbol{r}$ singularity is utilized at the crack tip, the correct singularity for a fully developed perfectly plastic solution. Symmetry boundary conditions are imposed on two edges of the submodel mesh, while results from the global shell analysis are interpolated to two surfaces via the submodeling technique. The global shell mesh gives satisfactory ***J***-integral results; hence, we assume that the displacements at the submodel boundary are sufficiently accurate to drive the deformation in the submodel. No attempt has

been made to study the effect of making the submodel region larger or smaller. The submodel is shown superimposed on the global shell model in Figure 1.4.3–5.

In addition, an input file for the case a_0/t =0.25, which consists of a full three-dimensional C3D20R solid element model, is included for use as a reference solution. This model has the same general characteristics as the submodel mesh. See inelasticlinespring_c3d20r_ful.inp for further details about this mesh. One important difference exists in performing this analysis with shell elements as opposed to continuum elements. The pressure loading is applied to the midsurface of the shell elements as opposed to the continuum elements, where the pressure is accurately applied along the inside surface of the cylinder. For this analysis this discrepancy results in about 10% higher ***J***-integral values for the line spring shell element analysis as compared to the full three-dimensional solid element model.

Results from the submodeled analyses are compared to the LS3S line spring element analysis and full solid element mesh for variations of the ***J***-integral values along the crack at the a normalized pressure loading of $p\bar{R}/(\sigma_y t) = 0.898$, where $\bar{R}$ is the mean radius of the cylinder. As seen in Figure 1.4.3–6, the line spring elements underestimate the J-integral values for ϕ <50° for reasons described previously. Note that at ϕ =0° the ***J***-integral should be zero due to the lack of crack-tip constraint at the cylinder surface. A more refined mesh would be required to model this phenomenon properly. It is quite obvious that the use of shell-to-solid submodeling is required to augment a line spring element model analysis to obtain accurate J-integral values near the surface of the cylinder.

Input files

inelasticlinespring_05.inp	a_0/t = 0.5.
inelasticlinespring_05_nosym.inp	a_0/t = 0.5 without the symmetry assumption across z =0, using line spring element type LS6.
inelasticlinespring_progcrack.f	A program used to create a data file giving the flaw depths as a function of position along the crack.
inelasticlinespring_025.inp	Shallow crack case, a_0/t =0.25.
inelasticlinespring_08.inp	Deep crack case, a_0/t =0.8.
inelasticlinespring_c3d20r_sub.inp	C3D20R (a_0/t =0.25) submodel.
inelasticlinespring_c3d20r_ful.inp	C3D20R (a_0/t =0.25) full model.

Reference

- Parks, D. M., and C. S. White, "Elastic-Plastic Line-Spring Finite Elements for Surface-Cracked Plates and Shells," Transactions of the ASME, Journal of Pressure Vessel Technology, vol. 104, pp. 287–292, November 1982.

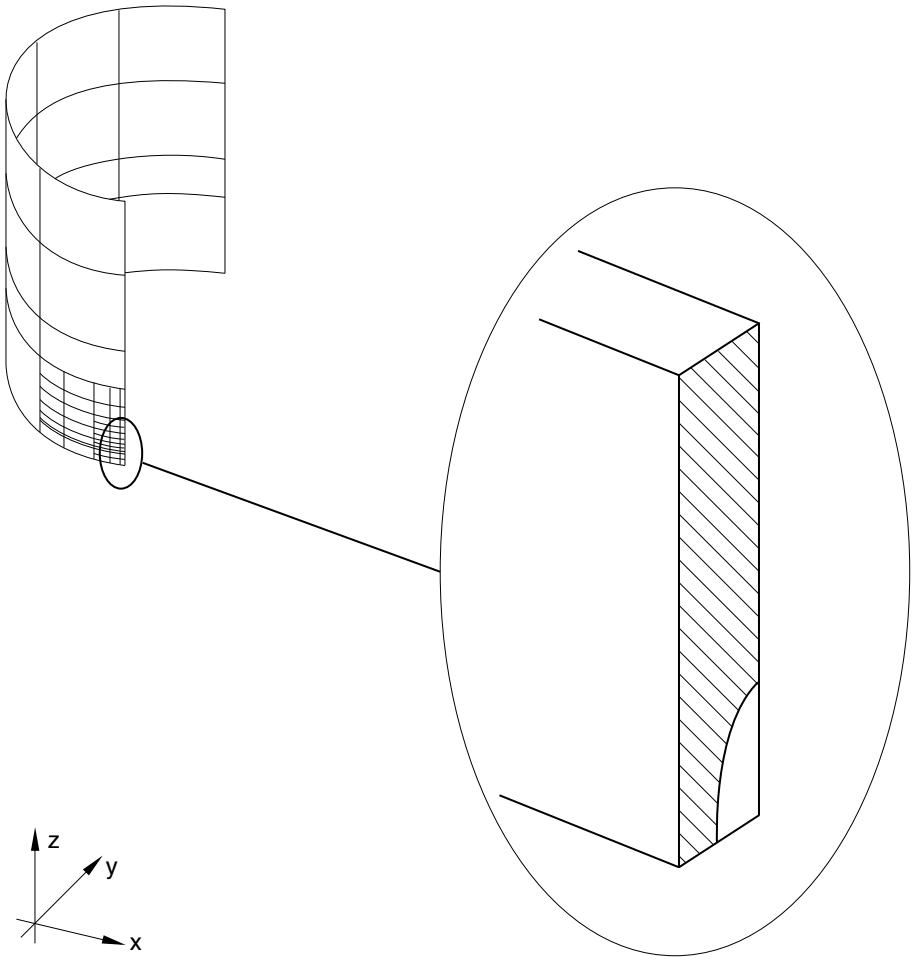
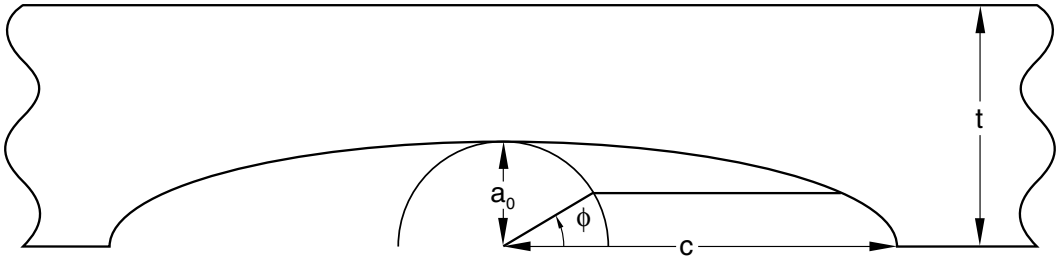


Figure 1.4.3–1 Finite element model for an axial flaw in a pressurized cylinder.



a_0 = maximum flaw depth

2c = surface length of crack

t = shell thickness

ϕ = angle on an inscribed circle for locating a point on the crack

Figure 1.4.3–2 Schematic of a semi-elliptical surface crack.

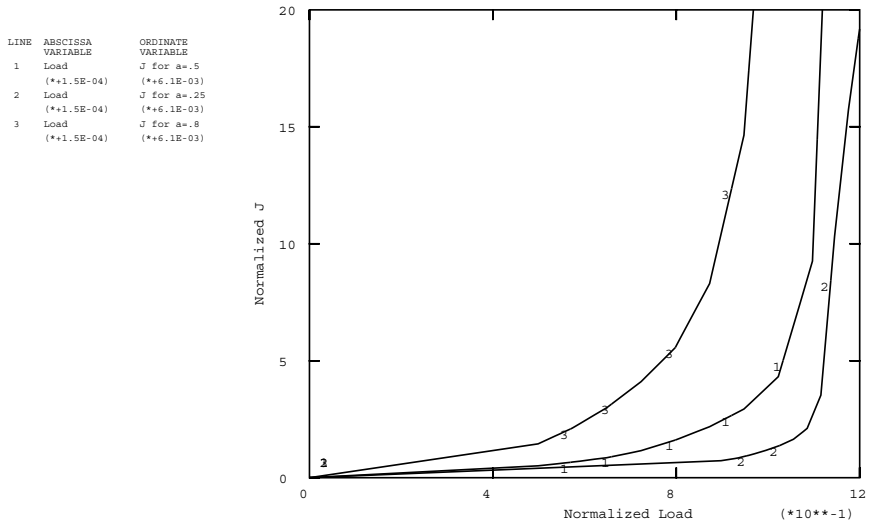


Figure 1.4.3–3 Normalized *J*-integral values $EJ/(\sigma_y^2 t)$ versus normalized applied pressure $p\bar{R}/(\sigma_y t)$, where $\bar{R}$ is the mean radius of the cylinder.

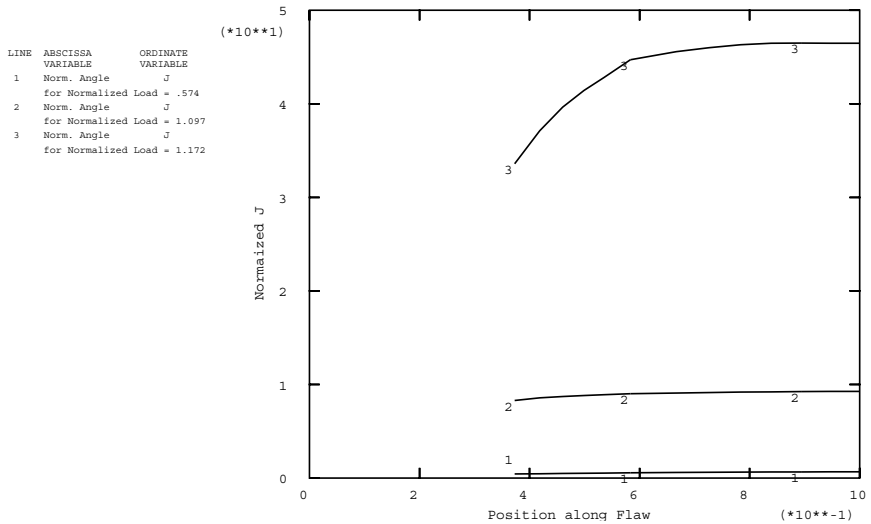


Figure 1.4.3–4 Normalized *J*-integral values $EJ/(\sigma_y^2 t)$ versus position along the flaw surface given by $2\phi/\pi$, for a_0/t =0.5, and normalized applied pressures $p\bar{R}/(\sigma_y t)$= .574, 1.097, and 1.172. $\bar{R}$ is the mean radius of the cylinder.

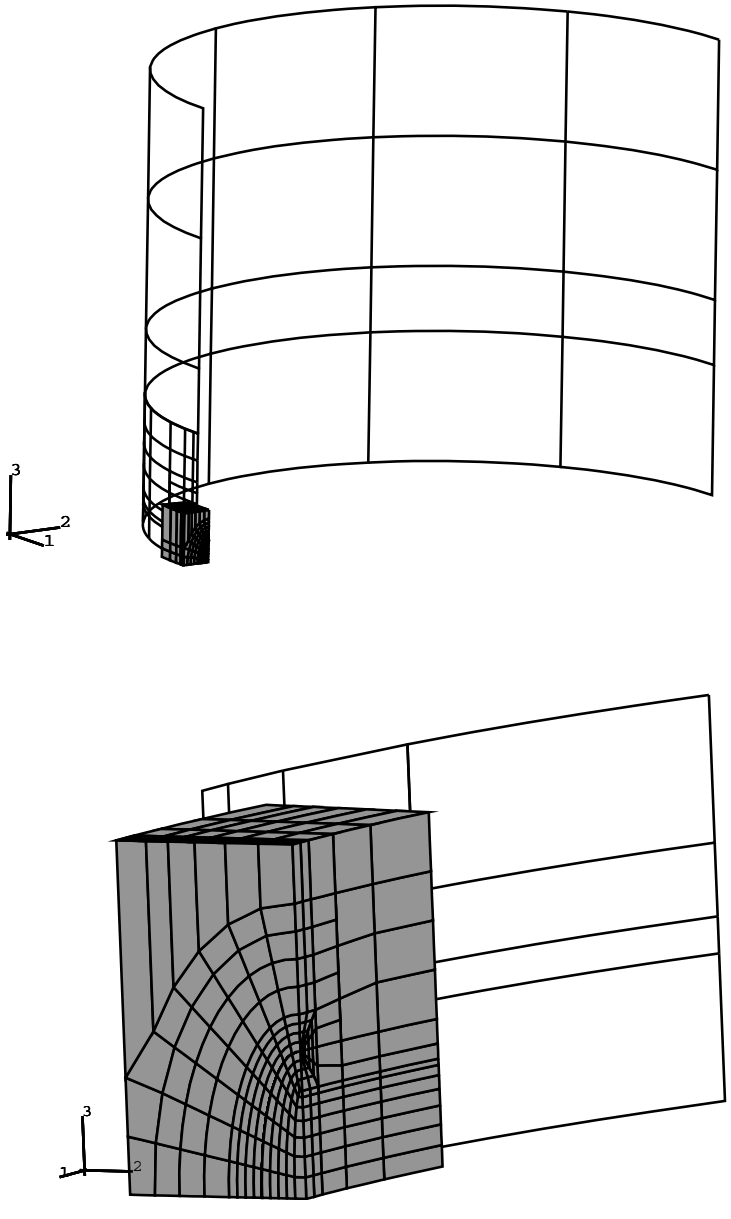


Figure 1.4.3–5 Solid submodel superimposed on shell global model.

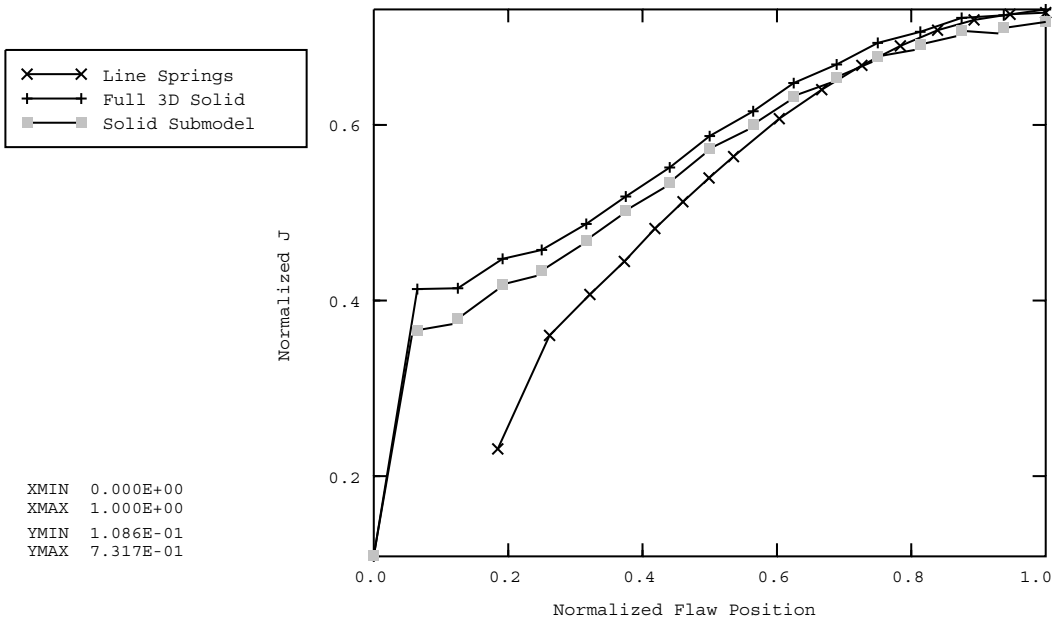


Figure 1.4.3–6 Normalized *J*-integral values $EJ/(\sigma_y^2 t)$ versus position along the flaw surface given by $2\phi/\pi$ for a_0/t =0.25 and at the normalized pressure. $p\bar{R}/(\sigma_y t)$ =0.898. $\bar{R}$ is the mean radius of the cylinder.

1.4.4 CRACK GROWTH IN A THREE-POINT BEND SPECIMEN

Product: Abaqus/Standard

This example illustrates the modeling of crack length versus time to simulate crack propagation and the use of crack opening displacement as a crack propagation criterion. For stable crack growth in ductile materials, experimental evidence indicates that the value of the crack opening displacement (COD) at a specified distance behind the crack tip associated with ongoing crack extension is usually a constant. Abaqus provides the critical crack opening displacement, at a specified distance behind the crack tip, as a crack propagation criterion. The other crack propagation model used in this example—prescribed crack length versus time—is usually used to verify the results obtained from experiments. Abaqus also provides the critical stress criterion for crack propagation in brittle materials.

In this example an edge crack in a three-point bend specimen is allowed to grow based on the crack opening displacement criterion. Crack propagation is first modeled by giving the crack length as a function of time. The data for the crack length are taken from Kunecke, Klingbeil, and Schicker (1993). The data for the crack propagation analysis using the COD criterion are taken from the first analysis. This example demonstrates how the COD criterion can be used in stable crack growth analysis.

Problem description

An edge crack in a three-point bend specimen in plane strain, subjected to Mode I loading, is considered (see Figure 1.4.4–1). The crack length to specimen width ratio is 0.2. The length of the specimen is 55 mm, and its width is 10 mm. The specimen is subjected to bending loads such that initially a well-contained plastic zone develops for the stationary crack. Subsequently, the crack is allowed to grow.

Geometry and model

Due to symmetry only one-half of the specimen is analyzed. The crack tip is modeled as initially blunted so that finite deformation effects near the crack tip can be taken into account (geometric nonlinearities considered in the step). The mesh is composed of 1737 CPE4 elements (Figure 1.4.4–2). A reasonably fine mesh, necessary to obtain a smooth load versus crack length relation, is used to model the area in which the plastic zone grows and crack propagation occurs. The loading point and the support points for the specimen are simulated by analytical rigid surfaces, as shown in Figure 1.4.4–2.

Material

The material is assumed to be elastic-plastic, with a Young's modulus of $E = 200$ GPa and Poisson's ratio of 0.3. The plastic work hardening data are given in Table 1.4.4–1.

Loading and solution control

The analysis is carried out in two stages. The first stage consists of pushing the rigid surface 1.0 mm into the specimen. No crack growth is specified during this stage.

In the second stage the crack is allowed to propagate while the rigid surface is moved an additional 1.951 mm.

Once a crack-tip node debonds, the traction at the tip is initially carried as a reaction force at that node. This force is ramped down to zero according to the amplitude curve specified in the crack propagation analysis. The manner in which the forces at the debonded nodes are ramped down greatly influences the convergence of the solution. The convergence of the solution is also affected by reversals in plastic flow due to crack propagation. In such circumstances, very small time increments are required to continue the analysis. In the present analysis solution controls are defined on the displacement field and on the warping degree of freedom equilibrium equations to relax the tolerances so that more rapid convergence is achieved. Because of the localized nature of the nonlinearity in this problem, the resulting loss of accuracy is not significant. The definition of solution controls is generally not recommended.

Crack length versus time

In the case when the crack length is given as a function of time, the second step in the analysis consists of letting the crack grow according to a prescribed crack length versus time relationship, using the data taken from Kunecke, Klingbeil, and Schicker.

COD criterion

The loading of the specimen and the specification of the COD criterion for crack growth demonstrates the flexibility of the critical crack opening displacement criterion. Frequently, the crack opening displacement is measured at the mouth of the crack tip: this is called the crack mouth opening displacement (CMOD). The crack opening displacement can also be measured at the position where the initial crack tip was located. Alternatively, the crack-tip opening angle (CTOA), defined as the angle between the two surfaces at the tip of the crack, is measured. The crack-tip opening angle can be easily reinterpreted as the crack opening at a distance behind the crack tip. In this example the COD specification required to use both the CMOD and the CTOA criteria is demonstrated.

For the purposes of demonstration the crack opening displacement at the mouth of the crack is used as the initial debond criterion. The first three nodes along the crack propagation surface are allowed to debond when the crack opening displacement at the mouth of the crack reaches a critical value. To achieve this, the following loading sequence is adopted: in Step 1, the specimen is loaded to a particular value (crack propagation analysis is not used), and in Step 2 the first crack-tip node is allowed to debond (crack propagation analysis is used). Steps 3 and 4 and Steps 5 and 6 follow the same sequence as Steps 1 and 2 so that the two successive nodes can debond. Since, the crack opening displacement is measured at the mouth of the crack, the value of the distance behind the crack tip along the slave surface is different in Steps 2, 4, and 6.

The loading sequence adopted above outlines a way in which the CMOD measurements can be simulated without encountering the situation in which the COD is measured beyond the bound of the specimen, which would lead to an error message. In this example, the loads at which the crack-tip nodes debonded were known *a priori*. In general, such information may not be available, and the restart capabilities in Abaqus can be used to determine the load at which the fracture criterion is satisfied.

The remaining bonded nodes along the crack propagation surface are allowed to debond based on averaged values of the crack-tip opening angles for different accumulated crack lengths. The data

prescribed under the crack propagation criteria in Step 7 are the crack opening displacement values that were computed from the crack-tip opening angles observed in the analysis that uses the prescribed crack length versus time criterion. These crack-tip opening angles are converted to critical crack opening displacements at a fixed distance of 0.04 mm behind the crack tip. Hence, the crack opening displacement is measured very close to the current crack tip.

Results and discussion

Figure 1.4.4–3 shows a plot of the accumulated incremental crack length versus time. The user-specified data, as well as the results obtained from the finite element analysis based on the two criteria, are plotted. Good agreement is observed between the user input values and the results from the analysis. The curve based on the COD criterion does not correspond with the user-specified data toward the end of the analysis because an average crack opening displacement was assumed.

Figure 1.4.4–4 shows the reaction force at the node where the displacements are applied as a function of the accumulated incremental crack length, obtained from the analysis in which the crack length was specified as a function of time. The curve obtained when the COD criterion is used is almost identical and is not shown in this figure.

Figure 1.4.4–5 depicts the variation of the reaction force as a function of the displacement at the rigid body reference node.

The contours of equivalent plastic strain in the near crack-tip region for two different crack advance positions are shown in Figure 1.4.4–6 and Figure 1.4.4–7. Contours of the Mises equivalent stress at the final stage of the analysis are shown in Figure 1.4.4–8.

Input files

crackgrowth_lengthvtime.inp	Analysis with the crack length versus time criterion.
crackgrowth_cod.inp	Analysis with the COD criterion.
crackgrowth_model.inp	Model data for the two analysis files.

Reference

- G. Kunecke, D. Klingbeil, and J. Schicker, “Rißfortschrittssimulation mit der ABAQUS-option DEBOND am Beispiel einer statisch belasteten Kerbschlagbiegeprobe,” presented at the ABAQUS German Fracture Mechanics group meeting in Stuttgart, November 1993.

Table 1.4.4–1 Stress-strain data for isotropic plastic behavior.

True Stress (MPa)	True Strain
461.000	0.0
472.810	0.0187
521.390	0.0280
628.960	0.0590
736.306	0.1245
837.413	0.2970
905.831	0.5756
1208.000	1.9942

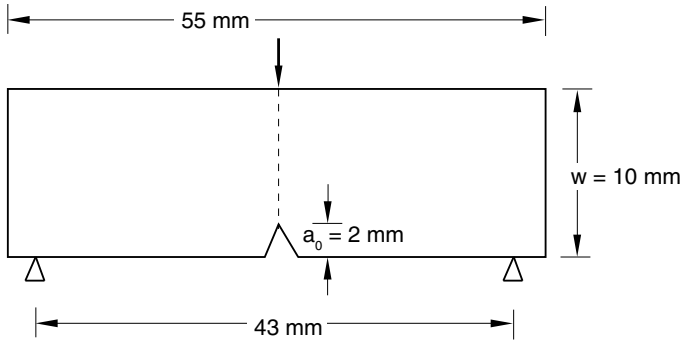


Figure 1.4.4–1 Schematic of the three-point bend specimen.

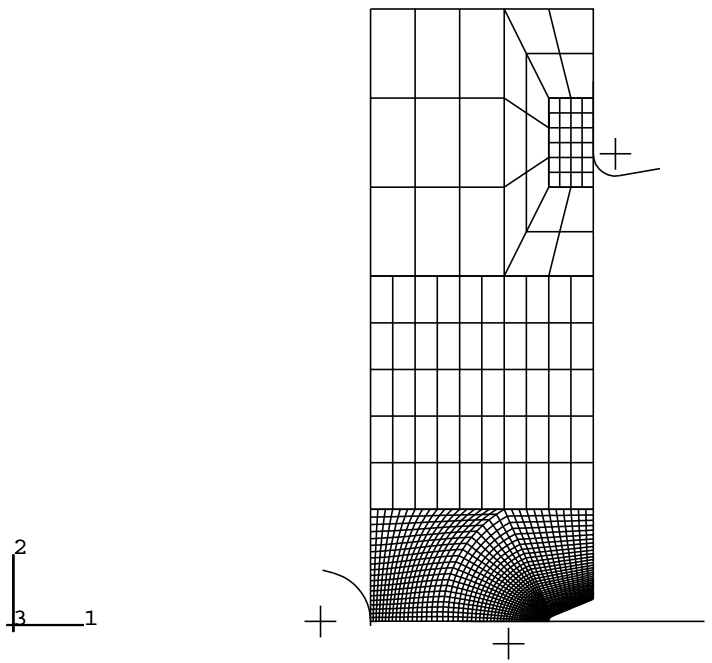


Figure 1.4.4–2 Finite element mesh for the three-point bend specimen.

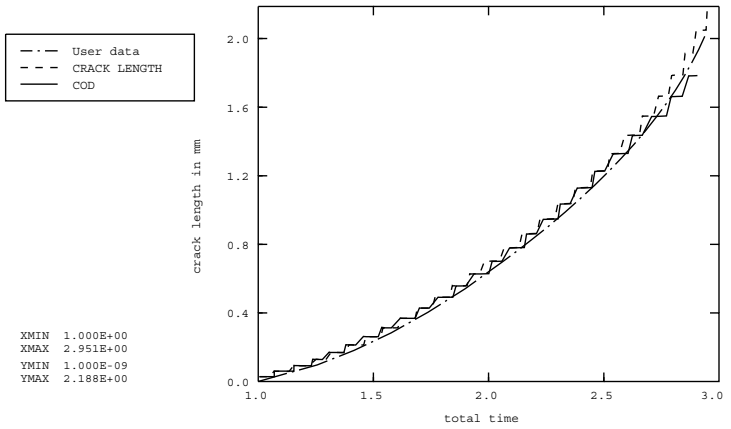


Figure 1.4.4–3 Accumulated incremental crack length versus time.

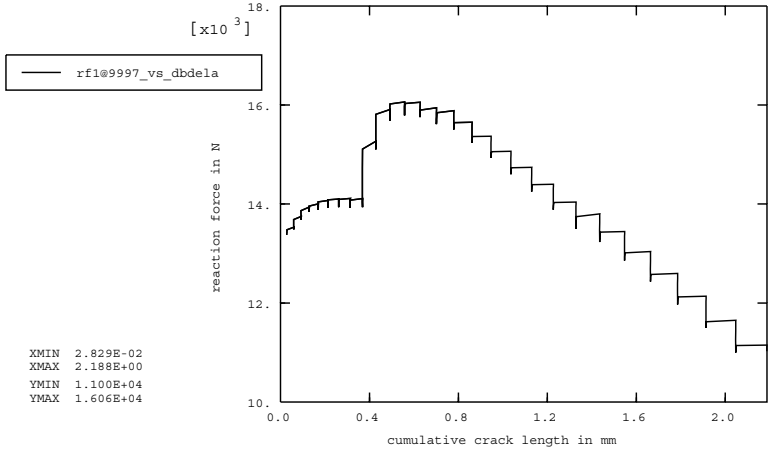


Figure 1.4.4–4 Variation of the reaction force as a function of the cumulative crack length.

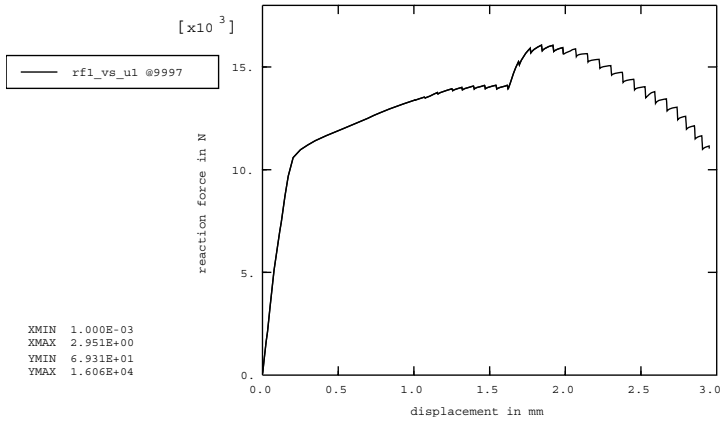


Figure 1.4.4–5 Variation of the reaction force as a function of displacement.

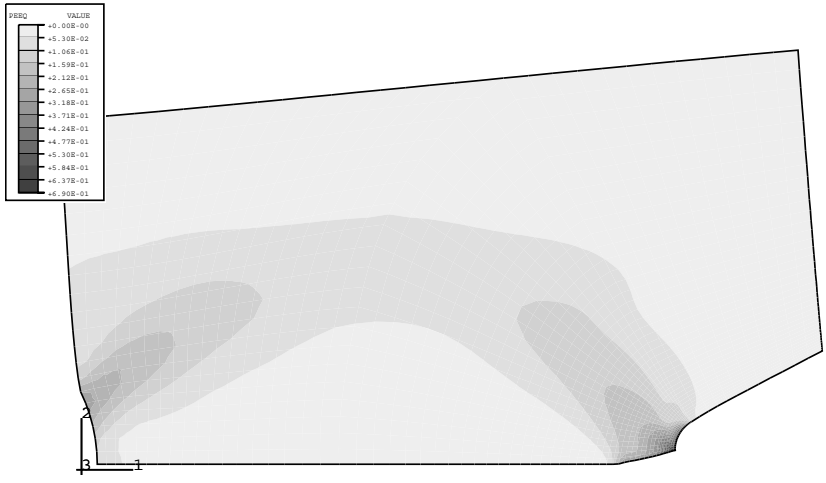


Figure 1.4.4–6 Plastic zone for an accumulated crack length of 1.03 mm.

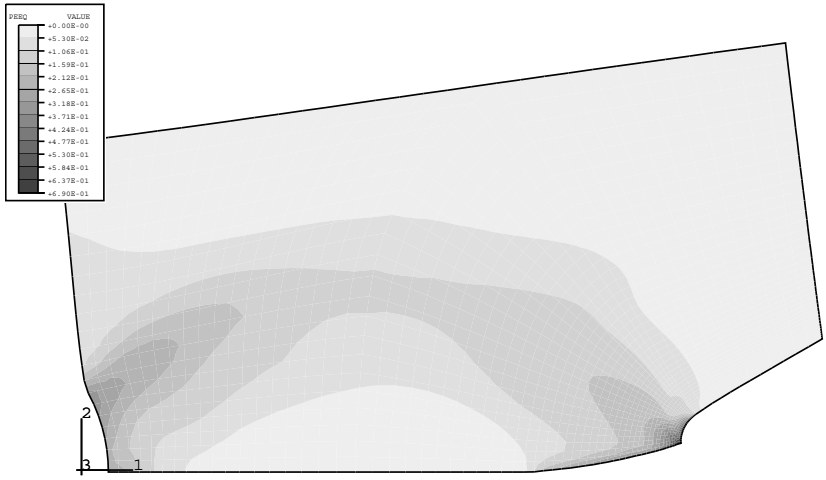


Figure 1.4.4–7 Plastic zone for an accumulated crack length of 2.18 mm.

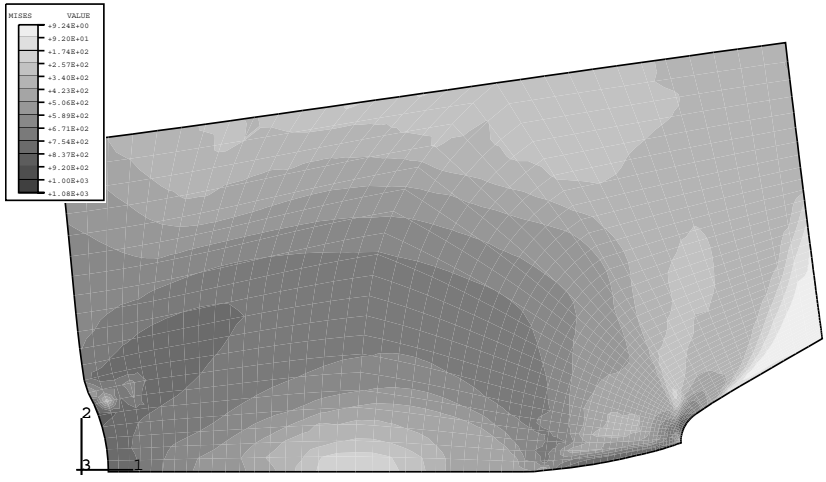


Figure 1.4.4–8 Contours of Mises stress for an accumulated crack length of 2.18 mm.

1.4.5 ANALYSIS OF SKIN-STIFFENER DEBONDING UNDER TENSION

Product: Abaqus/Standard

This example illustrates the application of cohesive elements in Abaqus to predict the initiation and progression of debonding at the skin-stiffener interface in stiffened panels, which is a common failure mode for this type of structure. The particular problem considered here is described in Davila (2003); it consists of a stringer flange bonded onto a skin, originally developed by Krueger (2000). The results presented are compared against the experimental results presented in Davila (2003). The problem is analyzed in Abaqus/Standard using a damaged, linear elastic constitutive model for the skin/stiffener interface.

Geometry and model

The problem geometry and loading are depicted in Figure 1.4.5–1: a 203-mm-long and 25.4-mm-wide specimen with a total skin thickness of 2.632 mm and maximum flange thickness of 1.88 mm, loaded in tension along the length direction. For the model in which the loading is simulated through prescribed displacements, the free gauge length is 127 mm. The skin thickness direction is comprised of 14 composite plies; while the flange is made up of 10 plies, each having a uniform thickness of 0.188 mm.

The finite element mesh for the three-dimensional model of the debonding problem is identical to that used in Davila (2003) except that the "decohesion" elements utilized in that reference to represent the skin/flange interface are replaced with Abaqus cohesive elements. Both the skin and the flange are modeled by two layers each of C3D8I elements. The interface between them is represented by COH3D8 elements, with the cohesive element mesh sharing nodes with the matching C3D8I meshes of the flange and the skin on either side. The model has a total of 828 solid elements and 174 cohesive elements. As stated in Davila, the two tapered ends of the flange are discretized differently to eliminate model symmetry and to prevent simultaneous delamination from occurring at both ends. The analysis includes a thermal loading step prior to the mechanical loading to simulate the residual stresses in the specimen due to a difference of 157°C between the curing temperature and room temperature. The temperature difference causes residual stresses at the skin/flange interface due to the fact that the thermal expansion coefficients of both the skin and flange material are orthotropic (even though they are specified to be the same) and the ply layups in the skin and flange are different.

Material

The material data, as given in Davila (2003), are reproduced below.

Composite material properties:

Engineering constants	
E_1	144.7 GPa
E_2	9.65 GPa
E_3	9.65 GPa

Engineering constants	
ν_{12}	0.30
ν_{13}	0.30
ν_{23}	0.45
G_{12}	5.2 GPa
G_{13}	5.2 GPa
G_{23}	3.4 GPa

The elastic properties of the interface material are defined using uncoupled traction-separation behavior (see "Defining elasticity in terms of tractions and separations for cohesive elements" in "Linear elastic behavior," Section 22.2.1 of the Abaqus Analysis User's Guide), with stiffness values of E= 1.0 × 10^6 MPa, G_1= 1.0 × 10^6 MPa, and G_2= 1.0 × 10^6 MPa. The quadratic traction-interaction failure criterion is chosen for damage initiation in the cohesive elements; and a mixed-mode, energy-based damage evolution law based on the Benzeggagh-Kenane criterion is used for damage propagation. The relevant material data are as follows: N_0=61 MPa, T_0= 68 MPa, S_0= 68 MPa, G_{1C}= 0.075 N/mm, G_{2C}= 0.547 N/mm, G_{3C}= 0.547 N/mm, and η= 1.45.

Results and discussion

The deformed geometry is given in Figure 1.4.5–2, which clearly shows the flange separation from the skin. In Figure 1.4.5–3 the load-extension predictions are compared with the experimental data presented by Davila (2003). The initiation of delamination is marked by the sharp slope change of the curves. The results presented here are obtained using three different viscous regularizations; i.e., $\mu = 1.0e^{-5}, 1.0e^{-4}$, and $1.0e^{-3}$. Higher viscosity provides better convergence but also affects the results more than lower viscosity. The results using $\mu = 1.0e^{-5}$ are close to the ones without viscosity, while the results using $\mu = 1.0e^{-3}$ agree best with the experimental results in terms of debonding initiation.

Input file

skinflangetension.inp	Input data for the three-dimensional skin/flange delamination model.

References

- Davila, C. G., and P. P. Camanho, "Analysis of the Effects of Residual Strains and Defects on Skin/Stiffener Debonding using Decohesion Elements," SDM Conference, Norfolk, VA, April 7–10, 2003.
- Krueger, R., M. K. Cvitkovich, T. K. O'Brien, and P. J. Minguet, "Testing and Analysis of Composite Skin/Stringer Debonding under Multi-Axial Loading," Journal of Composite Materials, vol. 34, no. 15, pp. 1263–1300, 2000.

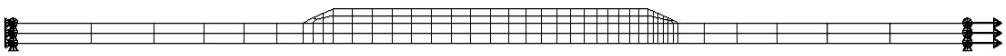


Figure 1.4.5–1 Model geometry for the skin/flange debond problem.

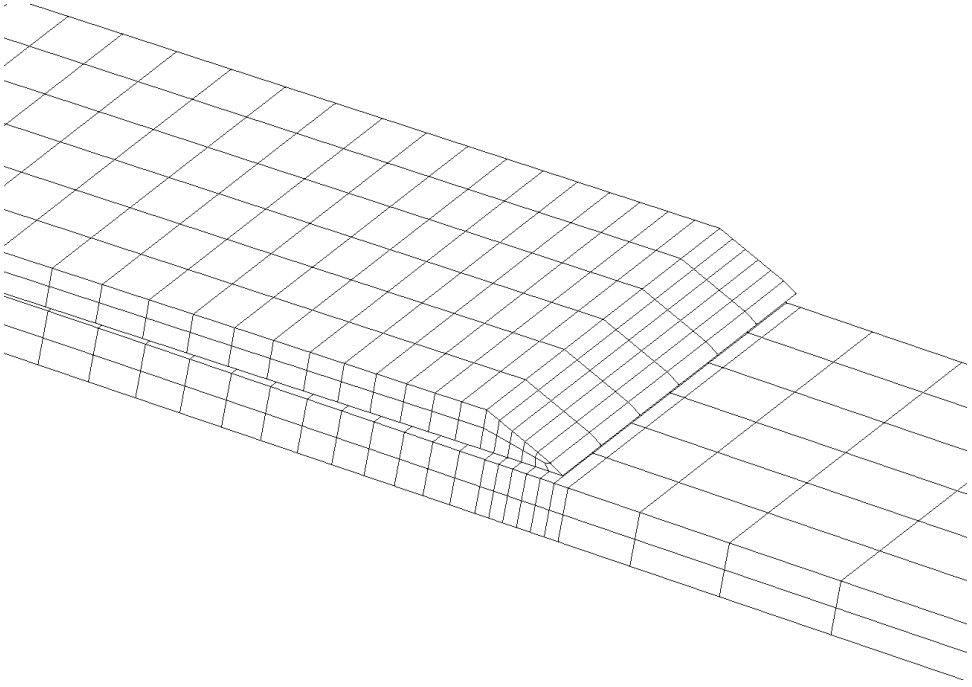


Figure 1.4.5–2 Deformed geometry after skin/flange debond.

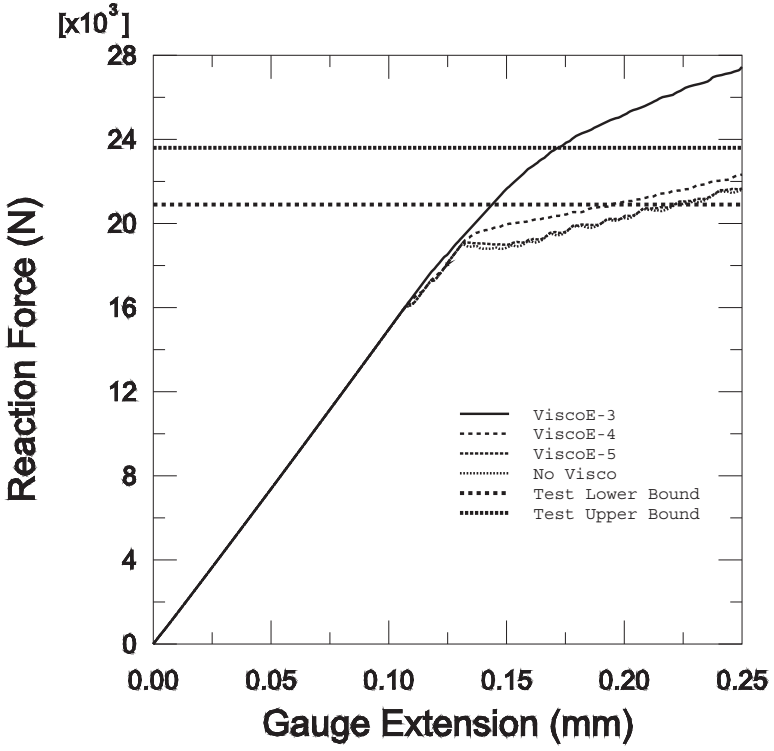


Figure 1.4.5–3 Predicted and experimental debond loads.

1.4.6 FAILURE OF BLUNT NOTCHED FIBER METAL LAMINATES

Products: Abaqus/Standard Abaqus/Explicit

Fiber metal laminates (FMLs) are composed of laminated thin aluminum layers bonded with intermediate glass fiber-reinforced epoxy layers. FMLs are of great interest in the aerospace industry due to their superior properties, such as high fracture toughness and low-density when compared to solid aluminum sheets.

This example simulates failure and damage in a FML containing a blunt notch subjected to quasi-static loading conditions. Cohesive elements are used to model the interlaminar delamination, and the Abaqus damage model for fiber-reinforced materials is used to predict behavior of the fiber-reinforced epoxy layer. In addition, the behavior of the fiber-reinforced epoxy layer is also described using the model proposed by Linde et al. (2004), which is implemented in user subroutine **UMAT**. Both Abaqus/Standard and Abaqus/Explicit are used for simulation when the Abaqus built-in damage model is used for fiber-reinforced epoxy layers. This type of problem is important in the aerospace industry since blunt notches (e.g., fastener holes) commonly occur in airplane structures; the strength of the structure containing a blunt notch is a crucial design parameter. The models presented in this example demonstrate how to predict the blunt notch strength, the failure patterns of the fiber and matrix within the fiber-reinforced epoxy layer, and the delamination between different layers of FMLs.

Problem description and material characteristics

Figure 1.4.6–1 shows the geometry of the laminate containing the blunt notch for this example. The laminate is subjected to uniaxial tension in the longitudinal direction. The laminate is made of three layers of aluminum and two layers of 0°/90° glass fiber-reinforced epoxy. Only 1/8 of the laminate needs to be modeled, with appropriate symmetric boundary conditions applied as shown in Figure 1.4.6–2. Figure 1.4.6–2 also shows the through-thickness lay-up of the 1/8 model.

The material behavior of aluminum is assumed to be isotropic elastic-plastic with isotropic hardening. The Young's modulus is 73800 MPa, and the Poisson's ratio is 0.33; the isotropic hardening data are listed in Table 1.4.6–1.

The material behavior of the glass fiber-reinforced epoxy layers is assumed to be orthotropic, with stiffer response along the fiber direction and softer behavior in the matrix. The elastic properties—longitudinal modulus, E_L; transverse modulus, E_T; shear moduli, G_{LT} and G_{TT}; and Poisson's ratios, ν_{TT} and ν_{LT}—are listed in Table 1.4.6–2. The subscript "L" refers to the longitudinal direction (or fiber direction), and the subscript "T" refers to the two transverse directions orthogonal to the fiber direction. The damage initiation and evolution behavior is also assumed to be orthotropic. Table 1.4.6–3 lists the ultimate values of the longitudinal failure stresses, $\sigma_L^{f,t}$ and $\sigma_L^{f,c}$; transverse failure stresses, $\sigma_T^{f,t}$ and $\sigma_T^{f,c}$; and in-plane shear failure stress, τ_{LT}^f. The superscripts "t" and "c" refer to tension and compression, respectively. The fracture energies of the fiber and matrix are assumed to be G_f=12.5 N/mm and G_m=1.0 N/mm, respectively.

Two material models that use the parameters listed above are considered, as follows:

1. The material is modeled based on the built-in model for damage in fiber-reinforced composites available in Abaqus (see "Damage and failure for fiber-reinforced composites: overview," Section 24.3.1 of the Abaqus Analysis User's Guide).
2. The material is modeled using an alternative damage model that is based on the model proposed by Linde et al. (2004). The alternative damage model is implemented in user subroutine **UMAT** and is referred to in this discussion as the **UMAT** model. Details of the **UMAT** model are provided below.

The adhesive used to bond neighboring layers is modeled using interface layers with a thickness of t=0.001 mm. To simulate the interlaminar delamination, these interface layers are modeled with cohesive elements. The initial elastic properties of each interface are assumed to be isotropic with Young's modulus E=2000 MPa and Poisson's ratio ν=0.33. The failure stresses of the interface layers are assumed to be $t_n^f=t_s^f=t_t^f$=50 MPa; the fracture energies are $G_n=G_s=G_t$=4.0 N/mm. The subscripts "n," "s," and "t" refer to the normal direction and the first and second shear directions (for further discussion of the constitutive modeling methods used for the adhesive layers, see "Defining the constitutive response of cohesive elements using a traction-separation description," Section 32.5.6 of the Abaqus Analysis User's Guide).

The plate is loaded with displacement boundary conditions applied at the right edge. To simplify the postprocessing, the displacement loading is applied at a reference point and an equation constraint is used to constrain the displacement along the loading direction between the right edge and the reference point. Except for those files designed exclusively to study the effect of the loading direction on the strength, the loading direction (along the global X-direction) aligns with the fiber direction of the 0° fiber-reinforced epoxy layer.

UMAT model for fiber-reinforced epoxy layers

For fiber-reinforced epoxy layers, the primary model considered is based on the built-in damage model for fiber-reinforced composites available in both Abaqus/Standard and Abaqus/Explicit. Alternatively, in Abaqus/Standard, the damage in the fiber-reinforced epoxy is also simulated using the model proposed by Linde et al. (2004), which is implemented in user subroutine **UMAT** and is discussed below.

In the **UMAT** model, the damage initiation criteria are expressed in terms of strains. Unlike the built-in model in Abaqus, which uses four internal (damage) variables, the **UMAT** model uses two damage variables to describe damage in the fiber and matrix without distinguishing between tension and compression. Although the performance of the two models is expected to be similar for monotonic loads, such as in this example problem, the results obtained might differ considerably for more complex loads in which, for example, tension is followed by compression. For the **UMAT** model, if the material is subjected to tensile stresses that are large enough to cause partial or full damage (the damage variable corresponding to this damage mode will be greater than zero), both tensile and compressive responses of the material will be affected. However, in the case of the built-in damage model, only the tensile response will be degraded while the material compressive response will not be affected. In many cases the latter behavior is more suitable for modeling fiber-reinforced composites. In this section the governing equations for damage initiation and evolution as proposed by Linde et al. (2004) are discussed, followed by a description of the user subroutine **UMAT** implementation.

Damage in the fiber is initiated when the following criterion is reached:

$$f_f = \sqrt{\frac{\epsilon_{11}^{f,t}}{\epsilon_{11}^{f,c}}(\epsilon_{11})^2 + \left(\epsilon_{11}^{f,t} - \frac{\left(\epsilon_{11}^{f,t}\right)^2}{\epsilon_{11}^{f,c}}\right)\epsilon_{11}} > \epsilon_{11}^{f,t},$$

where $\epsilon_{11}^{f,t} = \sigma_L^{f,t}/C_{11}$, $\epsilon_{11}^{f,c} = \sigma_L^{f,c}/C_{11}$, and C_{ij} are the components of the elasticity matrix in the undamaged state. Once the above criterion is satisfied, the fiber damage variable, d_f, evolves according to the equation

$$d_f = 1 - \frac{\epsilon_{11}^{f,t}}{f_f} e^{\left(-C_{11}\epsilon_{11}^{f,t}(f_f - \epsilon_{11}^{f,t})L^c/G_f\right)},$$

where L^c is the characteristic length associated with the material point. Similarly, damage initiation in the matrix is governed by the criterion

$$f_m = \sqrt{\frac{\epsilon_{22}^{f,t}}{\epsilon_{22}^{f,c}}(\epsilon_{22})^2 + \left(\epsilon_{22}^{f,t} - \frac{\left(\epsilon_{22}^{f,t}\right)^2}{\epsilon_{22}^{f,c}}\right)\epsilon_{22} + \left(\frac{\epsilon_{22}^{f,t}}{\epsilon_{12}^{f}}\right)^2 (\epsilon_{12})^2} > \epsilon_{22}^{f,t},$$

where $\epsilon_{22}^{f,t} = \sigma_T^{f,t}/C_{22}$, $\epsilon_{22}^{f,c} = \sigma_T^{f,c}/C_{22}$, and $\epsilon_{12}^{f} = \tau_{LT}^{f}/C_{44}$. The evolution law of the matrix damage variable, d_m, is

$$d_m = 1 - \frac{\epsilon_{22}^{f,t}}{f_m} e^{\left(-C_{22}\epsilon_{22}^{f,t}(f_m - \epsilon_{22}^{f,t})L^c/G_m\right)}.$$

During progressive damage the effective elasticity matrix is reduced by the two damage variables d_f and d_m, as follows:

$$\mathbf{C}_d = \begin{bmatrix} (1-d_f)C_{11} & (1-d_f)(1-d_m)C_{12} & (1-d_f)C_{13} & 0 & 0 & 0 \\ & (1-d_m)C_{22} & (1-d_m)C_{23} & 0 & 0 & 0 \\ & & C_{33} & 0 & 0 & 0 \\ & symmetric & & (1-d_f)(1-d_m)C_{44} & 0 & 0 \\ & & & & C_{55} & 0 \\ & & & & & C_{66} \end{bmatrix}.$$

The use of the fracture energy-based damage evolution law and the introduction of the characteristic length L^c in the damage evolution law help to minimize the mesh sensitivity of the numerical results, which is a common problem of constitutive models with strain softening response. However, since the characteristic length calculation is based only on the element geometry without taking into account the real cracking direction, some level of mesh sensitivity remains. Therefore, elements with an aspect ratio close to one are recommended (for a discussion of mesh sensitivity, see "Concrete damaged plasticity," Section 23.6.3 of the Abaqus Analysis User's Guide).

In user subroutine **UMAT** the stresses are updated according to the following equation:

$$\boldsymbol{\sigma} = \mathbf{C}_d : \boldsymbol{\epsilon}.$$

The Jacobian matrix can be obtained by differentiating the above equation:

$$\begin{aligned}\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\epsilon}} &= \mathbf{C}_d + \frac{\partial \mathbf{C}_d}{\partial \boldsymbol{\epsilon}} : \boldsymbol{\epsilon} \\ &= \mathbf{C}_d + (\frac{\partial \mathbf{C}_d}{\partial d_m} : \boldsymbol{\epsilon})(\frac{\partial d_m}{\partial f_m}\frac{\partial f_m}{\partial \boldsymbol{\epsilon}}) + (\frac{\partial \mathbf{C}_d}{\partial d_f} : \boldsymbol{\epsilon})(\frac{\partial d_f}{\partial f_f}\frac{\partial f_f}{\partial \boldsymbol{\epsilon}}).\end{aligned}$$

The above Jacobian matrix is not symmetric; therefore, the unsymmetric equation solution technique is recommended if the convergence rate is slow.

To improve convergence, a technique based on viscous regularization (a generalization of the Duvaut-Lions regularization) of the damage variables is implemented in the user subroutine. In this technique we do not use the damage variables calculated from the aforementioned damage evolution equations directly; instead, the damage variables are "regularized" via the following equations:

$$\dot{d}_m^v = \frac{1}{\eta}(d_m - d_m^v),$$

$$\dot{d}_f^v = \frac{1}{\eta}(d_f - d_f^v),$$

where d_m and d_f are the matrix and fiber damage variables calculated according to the damage evolution laws presented above, d_m^v and d_f^v are the "regularized" damage variables used in the real calculations of the damaged elasticity matrix and the Jacobian matrix, and η is the viscosity parameter controlling the rate at which the regularized damage variables d_m^v and d_f^v approach the true damage variables d_m and d_f.

To update the "regularized" damage variables at time $t_0 + \Delta t$, the above equations are discretized in time as follows:

$$d_m^v|_{t_0+\Delta t} = \frac{\Delta t}{\eta + \Delta t} d_m|_{t_0+\Delta t} + \frac{\eta}{\eta + \Delta t} d_m^v|_{t_0},$$

$$d_f^v|_{t_0+\Delta t} = \frac{\Delta t}{\eta + \Delta t} d_f|_{t_0+\Delta t} + \frac{\eta}{\eta + \Delta t} d_f^v|_{t_0}.$$

From the above expressions it can be seen that

$$\frac{\partial d_m^v}{\partial d_m} = \frac{\partial d_f^v}{\partial d_f} = \frac{\Delta t}{\eta + \Delta t}.$$

Therefore, the Jacobian matrix can be further formulated as follows:

$$\frac{\partial \Delta\boldsymbol{\sigma}}{\partial \Delta\boldsymbol{\epsilon}} = \mathbf{C}_d + \left[(\frac{\partial \mathbf{C}_d}{\partial d_m^v} : \boldsymbol{\epsilon})(\frac{\partial d_m}{\partial f_m}\frac{\partial f_m}{\partial \boldsymbol{\epsilon}}) + (\frac{\partial \mathbf{C}_d}{\partial d_f^v} : \boldsymbol{\epsilon})(\frac{\partial d_f}{\partial f_f}\frac{\partial f_f}{\partial \boldsymbol{\epsilon}})\right]\frac{\Delta t}{\eta + \Delta t}.$$

Care must be exercised to choose an appropriate value for η since a large value of viscosity might cause a noticeable delay in the degradation of the stiffness. To estimate the effect of viscous regularization, the approximate amount of energy associated with viscous regularization is integrated incrementally in user subroutine **UMAT** by updating the variable **SCD** as follows:

$$\Delta E_{SCD} = \frac{1}{2}\left[(\mathbf{C}_d : \boldsymbol{\epsilon})|_{t_0} + (\mathbf{C}_d : \boldsymbol{\epsilon})|_{t_0+\Delta t}\right] : d\boldsymbol{\epsilon} - \frac{1}{2}\left[(\mathbf{C}_d^0 : \boldsymbol{\epsilon})|_{t_0} + (\mathbf{C}_d^0 : \boldsymbol{\epsilon})|_{t_0+\Delta t}\right] : d\boldsymbol{\epsilon},$$

where $\mathbf{C}_d^0$ is the damaged elasticity matrix calculated using the damage variables, d_m and d_f; and $\mathbf{C}_d$ is the damaged elasticity matrix calculated using the regularized damage variables, d_m^v and d_f^v. To avoid unrealistic results due to viscous regularization, the above calculated energy (available as output variable ALLCD) should be small compared to the other real energies in the system, such as the strain energy ALLSE.

This user subroutine can be used with either three-dimensional solid elements or elements with plane stress formulations. In the user subroutine the fiber direction is assumed to be along the local 1 material direction. Therefore, when solid elements are used or when shell elements are used and the fiber direction does not align with the global ***X***-direction, a local material orientation should be specified. The damage variables—d_m, d_f, d_m^v, and d_f^v—are stored as solution-dependent variables, which can be viewed in the Visualization module of Abaqus/CAE.

Finite element model

The finite element model uses a separate mesh for each of the respective layers shown in Figure 1.4.6–2: two aluminium layers, two fiber-reinforced epoxy layers, and three adhesive layers. While not required, a similar finite element discretization in the plane of the laminate, such as that shown in Figure 1.4.6–3, can be used for all layers.

Modeling considerations for aluminium layers

Due to the interactions with the fiber-reinforced epoxy layers, the stress state within the aluminum layers (especially surrounding the notch tip) cannot be approximated using the plane stress assumption. To model this three-dimensional plasticity stress state accurately, solid elements must be used for the aluminum layers. In Abaqus/Standard incompatible mode elements (C3D8I) are used since local bending might exist in the post-failure region surrounding the notch. For the Abaqus/Explicit analysis, reduced-integration elements (C3D8R) are used for modeling the aluminum layers.

Modeling considerations for glass fiber-reinforced epoxy layers

The plane stress assumption can be used safely within the fiber-reinforced epoxy layers; therefore, either solid elements or shell elements can be adopted for these layers. However, it is important to have an accurate representation of the through-thickness geometry to model the interface between the adhesive and the fiber-reinforced epoxy realistically. This is achieved most conveniently with solid elements or continuum shell elements instead of conventional shell elements. The damage model for fiber-reinforced materials is available only for elements with a plane stress formulation. Therefore, continuum shell

elements are used with this model. Models are also included in which continuum elements (C3D8R or C3D8) are used along with user subroutine **UMAT** to model the fiber-reinforced epoxy layers.

Modeling considerations for adhesive layers

Cohesive elements (COH3D8) are used for the interface layers. The elastic response is defined in terms of a traction-separation law with uncoupled behavior between the normal and shear components. For convenience, a constitutive thickness of 1.0 mm is used so that we do not need to distinguish between the separation displacement and the nominal strain (NE). However, since the actual thickness is 0.001 mm, the diagonal terms in the elasticity matrix need to be scaled by the inverse of the actual thickness as follows:

$$K_{nn} = \frac{E}{t},\ K_{ss} = \frac{G}{t},\ K_{tt} = \frac{G}{t}.$$

The quadratic nominal strain criterion is used for the damage initiation:

$$\sqrt{(\frac{\epsilon_n}{\epsilon_n^f})^2 + (\frac{\epsilon_s}{\epsilon_s^f})^2 + (\frac{\epsilon_t}{\epsilon_t^f})^2} > 1.0,\ \epsilon_n^f = \frac{t_n^f}{E/t},\ \epsilon_s^f = \frac{t_s^f}{G/t},\ \epsilon_t^f = \frac{t_t^f}{G/t}.$$

The damage evolution is based on fracture energy with the quadratic power law for the mixed mode behavior and exponential softening behavior (see "Defining the constitutive response of cohesive elements using a traction-separation description," Section 32.5.6 of the Abaqus Analysis User's Guide).

Results and discussion

Results for each analysis are discussed in the following sections.

Abaqus/Standard results

Damage to the fiber-reinforced epoxy plays a key role in the response for the loading considered. Figure 1.4.6–4 shows the load-displacement curve for the 0° loading direction for both of the damage models considered for the fiber-reinforced epoxy. The response shows a "bilinear" shape before the sudden loss of loading capacity; i.e., an initial linear curve representing the initial elastic region, a smoothly deflecting nonlinear curve representing the local plasticity, and a second linear curve representing the net section yielding. The effect of the element type was studied using the **UMAT** model and C3D8R, C3D8, and SC8R elements; and the results are summarized in Figure 1.4.6–5 and Table 1.4.6–5. The numerical results obtained using different element types and different damage models are similar and show a good agreement with the experimental results of De Vries (2001).

The fiber and matrix damage patterns in the 0° fiber-reinforced epoxy layer at the failure load are shown in Figure 1.4.6–6 and Figure 1.4.6–7 for the built-in damage model for fiber-reinforced materials and in Figure 1.4.6–8 and Figure 1.4.6–9 for the **UMAT** model. It can be seen that the fiber damage in the 0° fiber-reinforced epoxy layer propagates along the ligament above the blunt notch tip (i.e., orthogonal to the loading direction). Figure 1.4.6–10 shows the matrix damage in the 90° layer for the damage model of Linde et al. (2004). There is no fiber damage in the 90° fiber-reinforced epoxy layer prior to the

sudden fracture. Interlaminar damage is most severe between the 0° fiber-reinforced epoxy layer and the aluminum layer. These observations are in agreement with the experimental results of De Vries (2001).

Figure 1.4.6–11 and Table 1.4.6–4 give the load-displacement results for different values of the viscosity parameter, η, obtained using the built-in damage model for fiber-reinforced materials. The same results for the **UMAT** model are given in Figure 1.4.6–12 and Table 1.4.6–6. The smaller the viscosity, the more abrupt the failure and the smaller the failure strength. Although a viscosity of 0.001 seems to overestimate the failure strength by a few percent (Table 1.4.6–4 and Table 1.4.6–6), the convergence is noticeably improved; thus, a viscosity of 0.001 is used for all the other studies in this example. For the built-in damage model for fiber-reinforced materials, only the viscosity in the fiber direction was varied while the viscosity in the matrix direction was kept constant at 0.005. This improved convergence and did not markedly affect the results.

The effect of the loading direction on the blunt notch strength is studied using the three-dimensional element, C3D8R, with the **UMAT** model. Three tests are performed in which the local material orientations in the 0°/90° fiber-reinforced epoxy are rotated by an angle of 15°, 30°, and 45°, respectively. For example, for a loading angle of 15° the fiber orientation in the 0° fiber-reinforced epoxy layer would be at a 15° angle with respect to the X-direction, while the fiber orientation in the 90° fiber-reinforced epoxy layer would be at an angle of −75° with respect to the X-direction (Figure 1.4.6–13). As can be seen in Figure 1.4.6–14, strain hardening is smaller for the larger loading angles. As can be seen in Figure 1.4.6–15, the failure strength decreases with the increasing loading angle and reaches the minimum at the 45° loading angle (the response for even larger loading angles is expected to be approximately symmetric with respect to the 45° angle due to the symmetric nature of the 0°/90° fiber-reinforced epoxy layer). As stated by De Vries (2001), this is expected and reflects the poor shear properties of the fiber-reinforced epoxy layer.

In the above discussions the net blunt notch strength is defined as $P_{max}/l_1 t$, where l_1 is the length of the ligament above the notch and t is the total thickness of the laminate. This example demonstrates that the approach employed in the study can be used to predict the blunt notch strength of the fiber metal laminates.

Abaqus/Explicit results

In the Abaqus/Explicit simulation we only consider loading along the 0° ply. The simulation is conducted without damage stabilization, and no mass scaling is used. However, in order to reduce the computational time, the total loading is applied in a short interval of time (0.001 s). The overall load-displacement curve obtained from the explicit dynamic simulation is compared with the Abaqus/Standard result (with viscosity of 0.001) in Figure 1.4.6–16. The results from the explicit dynamic simulation are presented using an antialiasing filter to remove high frequency noise (see "Filtering output and operating on output in Abaqus/Explicit" in "Output to the output database," Section 4.1.3 of the Abaqus Analysis User's Guide). The overall response compares well with the Abaqus/Standard results with some differences in the peak value of the load and in the post-peak response. Note that damage stabilization is used in the Abaqus/Standard simulation to achieve convergence and is likely to change the overall response (especially in the post-peak portion of the load-displacement curve). On the other hand, the Abaqus/Explicit simulation does not use damage stabilization and is better able to capture the dynamic behavior inherent in the damage and failure

processes. The contour plots of various damage variables in the 0° and 90° plies agree qualitatively with the corresponding plots obtained from the Abaqus/Standard simulation using the built-in damage model.

Python scripts

fml_c3d8r_deg0_vis1_std.py	C3D8R used in the fiber-reinforced epoxy layer, a loading angle of 0°, and a viscosity of 0.001.
fml_c3d8r_deg0_vis2_std.py	C3D8R used in the fiber-reinforced epoxy layer, a loading angle of 0°, and a viscosity of 0.0004.
fml_c3d8r_deg0_vis3_std.py	C3D8R used in the fiber-reinforced epoxy layer, a loading angle of 0°, and a viscosity of 0.00016.
fml_c3d8r_deg0_vis4_std.py	C3D8R used in the fiber-reinforced epoxy layer, a loading angle of 0°, and a viscosity of 0.000064.
fml_c3d8_deg0_vis1_std.py	C3D8 used in the fiber-reinforced epoxy layer, a loading angle of 0°, and a viscosity of 0.001.
fml_sc8r_deg0_vis1_std.py	SC8R used in the fiber-reinforced epoxy layer, a loading angle of 0°, and a viscosity of 0.001.
fml_c3d8r_deg15_vis1_std.py	C3D8R used in the fiber-reinforced epoxy layer, a loading angle of 15°, and a viscosity of 0.001.
fml_c3d8r_deg30_vis1_std.py	C3D8R used in the fiber-reinforced epoxy layer, a loading angle of 30°, and a viscosity of 0.001.
fml_c3d8r_deg45_vis1_std.py	C3D8R used in the fiber-reinforced epoxy layer, a loading angle of 45°, and a viscosity of 0.001.

Input files

Abaqus/Standard input files

fml_frm_sc8r_deg0_vis001_std.inp	SC8R used in the fiber-reinforced epoxy layer, a loading angle of 0°, and a viscosity of 0.001 in the fiber direction (using built-in fiber-reinforced material damage model).
fml_frm_sc8r_deg0_vis0005_std.inp	SC8R used in the fiber-reinforced epoxy layer, a loading angle of 0°, and a viscosity of 0.0005 in the fiber direction (using built-in fiber-reinforced material damage model).
fml_frm_sc8r_deg0_vis00025_std.inp	SC8R used in the fiber-reinforced epoxy layer, a loading angle of 0°, and a viscosity of 0.00025 in the fiber direction (using built-in fiber-reinforced material damage model).
fml_c3d8r_deg0_vis1_std.inp	C3D8R used in the fiber-reinforced epoxy layer, a loading angle of 0°, and a viscosity of 0.001 (using **`UMAT`** model).
fml_c3d8r_deg0_vis2_std.inp	C3D8R used in the fiber-reinforced epoxy layer, a loading angle of 0°, and a viscosity of 0.0004 (using **`UMAT`** model).

fml_c3d8r_deg0_vis3_std.inp	C3D8R used in the fiber-reinforced epoxy layer, a loading angle of 0°, and a viscosity of 0.00016 (using **UMAT** model).
fml_c3d8r_deg0_vis4_std.inp	C3D8R used in the fiber-reinforced epoxy layer, a loading angle of 0°, and a viscosity of 0.000064 (using **UMAT** model).
fml_c3d8_deg0_vis1_std.inp	C3D8 used in the fiber-reinforced epoxy layer, a loading angle of 0°, and a viscosity of 0.001 (using **UMAT** model).
fml_sc8r_deg0_vis1_std.inp	SC8R used in the fiber-reinforced epoxy layer, a loading angle of 0°, and a viscosity of 0.001 (using **UMAT** model).
fml_c3d8r_deg15_vis1_std.inp	C3D8R used in the fiber-reinforced epoxy layer, a loading angle of 15°, and a viscosity of 0.001 (using **UMAT** model).
fml_c3d8r_deg30_vis1_std.inp	C3D8R used in the fiber-reinforced epoxy layer, a loading angle of 30°, and a viscosity of 0.001 (using **UMAT** model).
fml_c3d8r_deg45_vis1_std.inp	C3D8R used in the fiber-reinforced epoxy layer, a loading angle of 45°, and a viscosity of 0.001 (using **UMAT** model).
exa_fml_ortho_damage_umat.f	User subroutine **UMAT** for modeling the damage initiation and evolution in the fiber-reinforced epoxy layers.

Abaqus/Explicit input file

fml_frm_sc8r_deg0_exp.inp	SC8R elements used in the fiber-reinforced epoxy layer and a loading angle of 0° (using built-in fiber-reinforced material damage model).

References

- De Vries, T. J., "Blunt and Sharp Notch Behavior of Glare Laminates," Ph.D dissertation, Delft University Press, 2001.
- Hagenbeek, M., C. Van Hengel, O. J. Bosker, and C. A. J. R. Vermeeren, "Static Properties of Fibre Metal Laminates," Applied Composite Materials, vol. 10, p. 207–222, 2003.
- Linde, P., J. Pleitner, H. De Boer, and C. Carmone, "Modelling and Simulation of Fiber Metal Laminates," ABAQUS Users' Conference, 2004.

Table 1.4.6–1 Isotropic hardening data for aluminum.

Yield stress (MPa)	Plastic strain (%)
300	0.000
320	0.016
340	0.047
355	0.119
375	0.449
390	1.036
410	2.130
430	3.439
450	5.133
470	8.000
484	14.710

Table 1.4.6–2 Orthotropic elastic properties of fiber-reinforced epoxy.

E_L (MPa)	E_T (MPa)	G_{LT} (MPa)	G_{TT} (MPa)	ν_{TT}	ν_{LT}
55000	9500	5500	3000	0.45	0.33

Table 1.4.6–3 Orthotropic damage initiation properties of fiber-reinforced epoxy.

$\sigma_L^{f,t}$ (MPa)	$\sigma_L^{f,c}$ (MPa)	$\sigma_T^{f,t}$ (MPa)	$\sigma_T^{f,c}$ (MPa)	τ_{LT}^{f} (MPa)
2500	2000	50	150	50

Table 1.4.6–4 Net blunt notch strength (MPa) for different values of the viscosity parameter in fiber direction (using built-in fiber-reinforced material damage model, viscosity in the matrix direction η_m=0.005).

Numerical results (SC8R, 0° loading angle)			**Experimental results (De Vries, 2001)**
η_f**=0.001**	η_f**=0.0005**	η_f**=0.00025**	
462.1	456.4	453.2	446

Table 1.4.6–5 Net blunt notch strength (MPa) for different element types used in the fiber-reinforced epoxy layers (using **UMAT** model).

Numerical results (η=0.001, 0° loading angle)			**Experimental results (De Vries, 2001)**
C3D8R	**C3D8**	**SC8R**	
463.7	467.1	458.7	446

Table 1.4.6–6 Net blunt notch strength (MPa) for different values of the viscosity parameter (using **UMAT** model).

Numerical results (C3D8R, 0° loading angle)				**Experimental results (De Vries, 2001)**
η**=0.001**	η**=0.0004**	η**=0.00016**	η**=0.000064**	
463.7	453.8	449.2	448.2	446

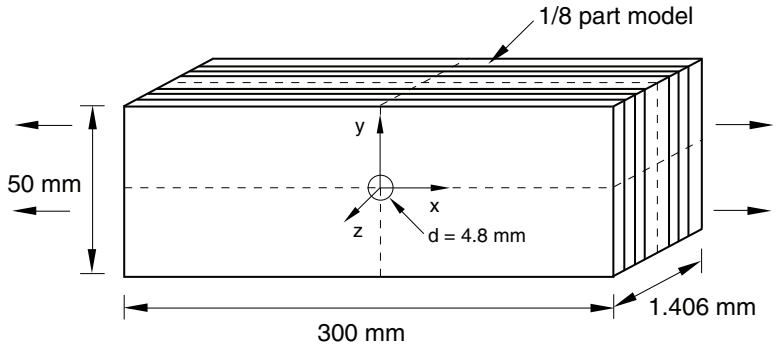


Figure 1.4.6–1 Plate geometry.

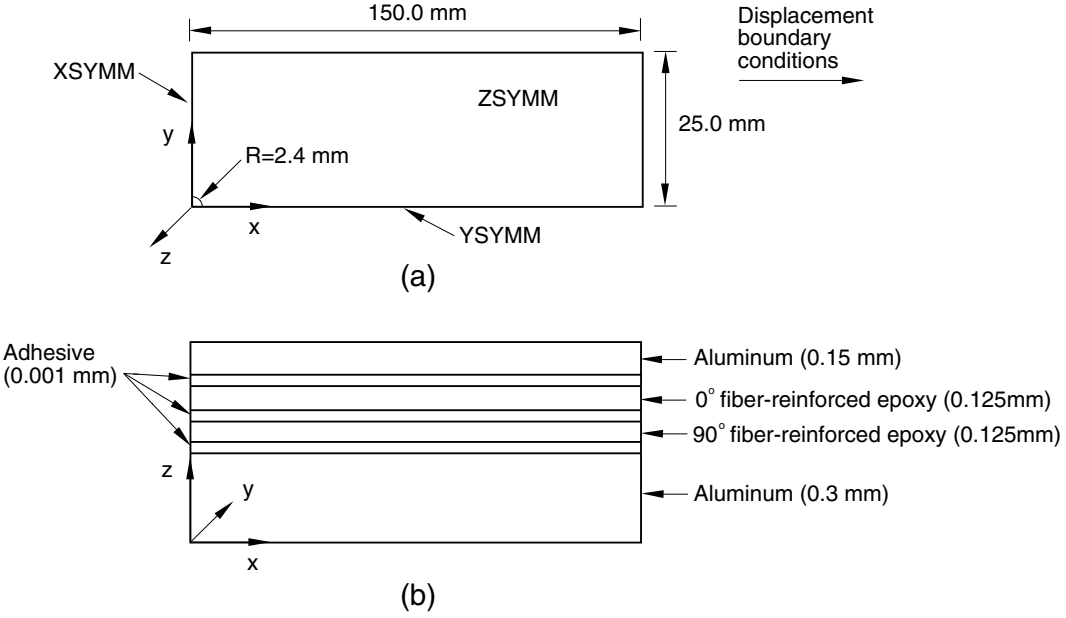


Figure 1.4.6–2 (a) In-plane view of the 1/8 plate; (b) through-thickness lay-up of the 1/8 plate.

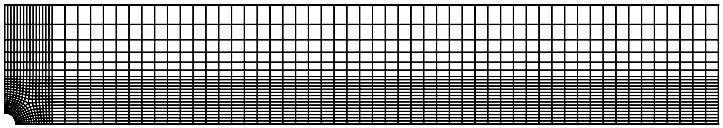


Figure 1.4.6–3 Finite element mesh.

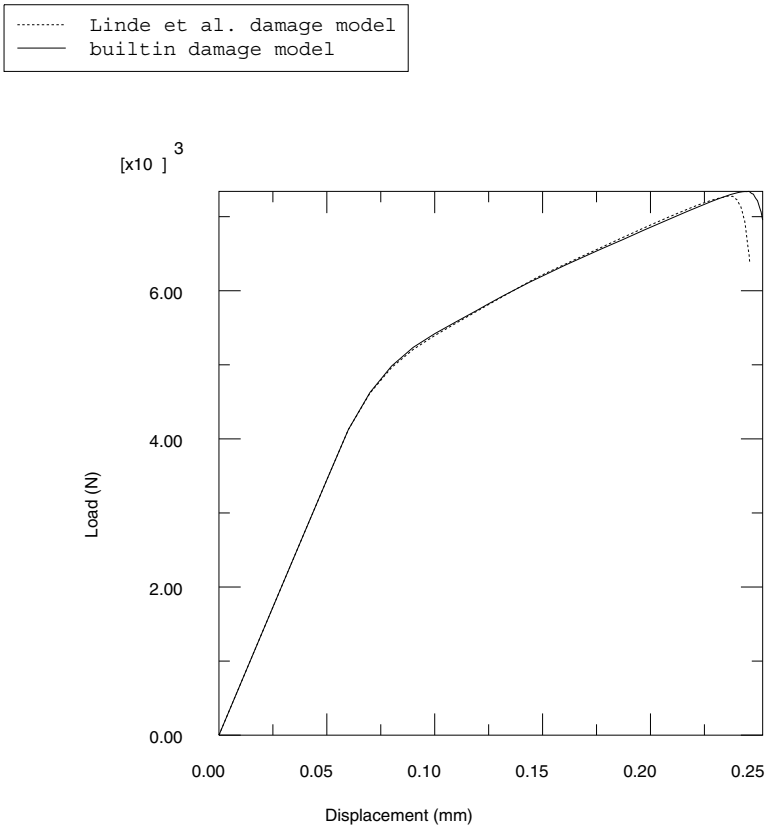


Figure 1.4.6–4 Load-displacement curves for different damage models in the fiber-reinforced epoxy layer for the 0° loading direction, η=0.001.

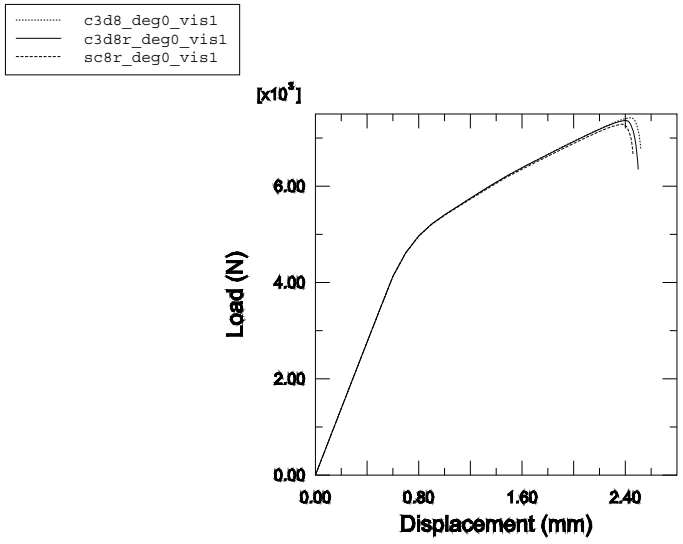


Figure 1.4.6–5 Load-displacement curves for different element types in the fiber-reinforced epoxy layer for the 0° loading direction (using **UMAT** model).

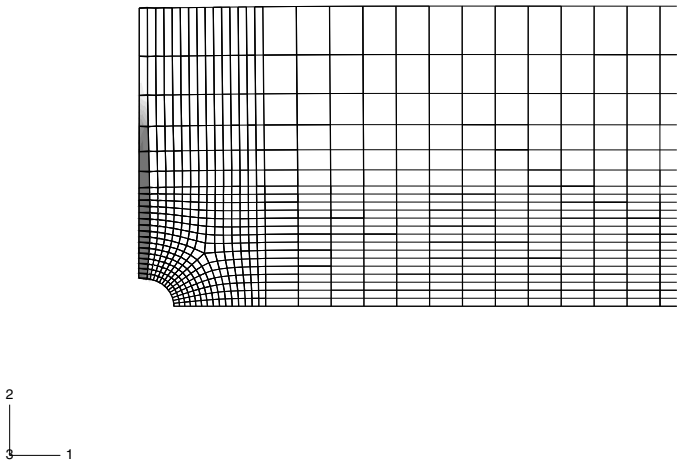


Figure 1.4.6–6 Fiber damage pattern in the 0° fiber-reinforced epoxy layer for the 0° loading direction (using built-in fiber-reinforced material model, DAMAGEFT contour plot).

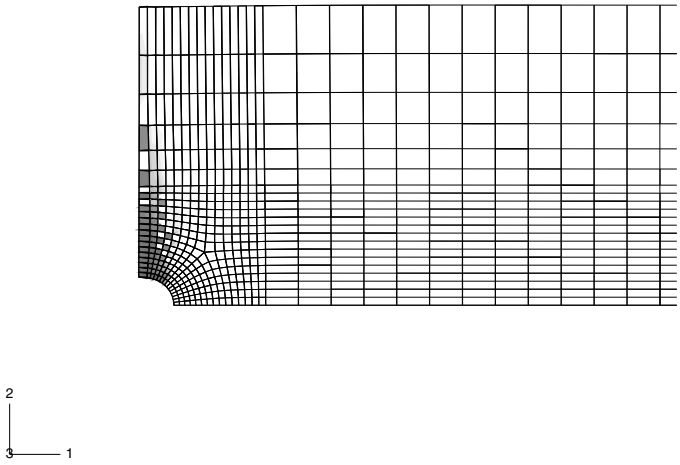


Figure 1.4.6–7 Matrix damage pattern in the 0° fiber-reinforced epoxy layer for the 0° loading direction (using built-in fiber-reinforced material model, DAMAGEMT contour plot).

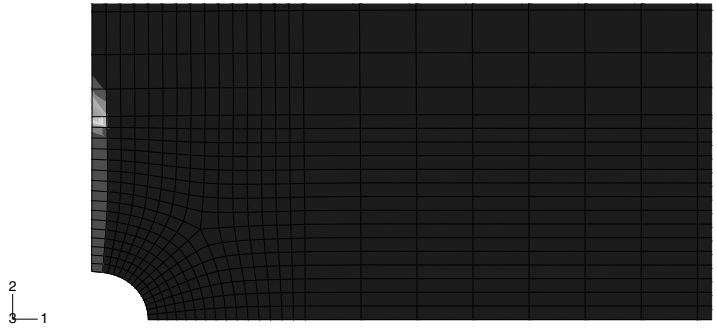


Figure 1.4.6–8 Fiber damage pattern in the 0° fiber-reinforced epoxy layer for the 0° loading direction (using **`UMAT`** model, SDV3 contour plot).

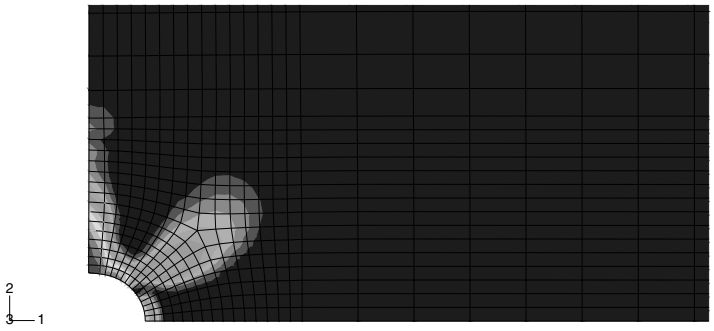


Figure 1.4.6–9 Matrix damage pattern in the 0° fiber-reinforced epoxy layer for the 0° loading direction (using **UMAT** model, SDV4 contour plot).

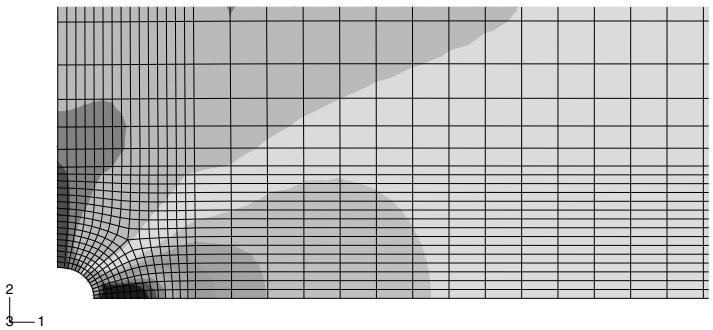


Figure 1.4.6–10 Matrix damage pattern in the 90° fiber-reinforced epoxy layer for the 0° loading direction (using **UMAT** model, SDV4 contour plot).

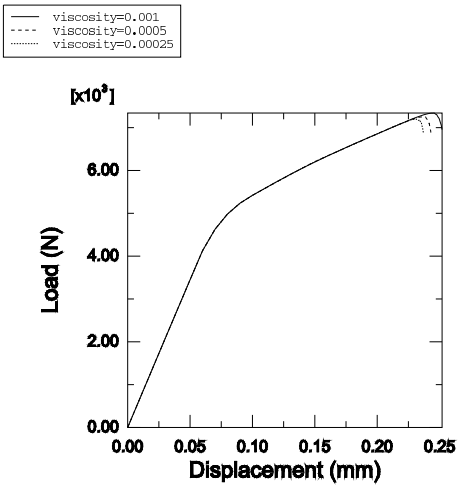


Figure 1.4.6–11 Load-displacement curves for different values of the viscosity parameter for the 0° loading direction (using built-in fiber-reinforced material damage model).

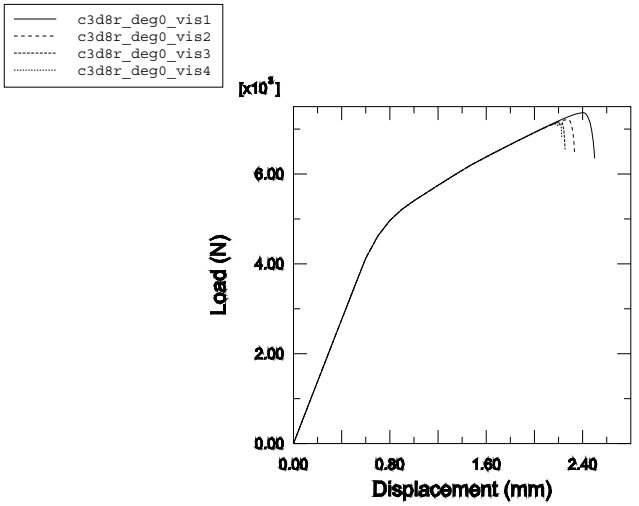


Figure 1.4.6–12 Load-displacement curves for different values of the viscosity parameter for the 0° loading direction (using **UMAT**).

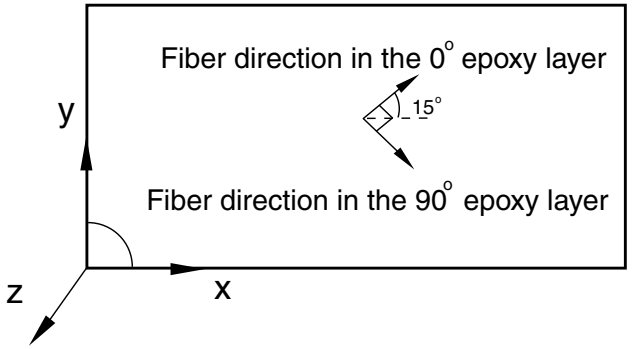


Figure 1.4.6–13 Local material orientations in the fiber-reinforced epoxy layers for the 15° loading direction.

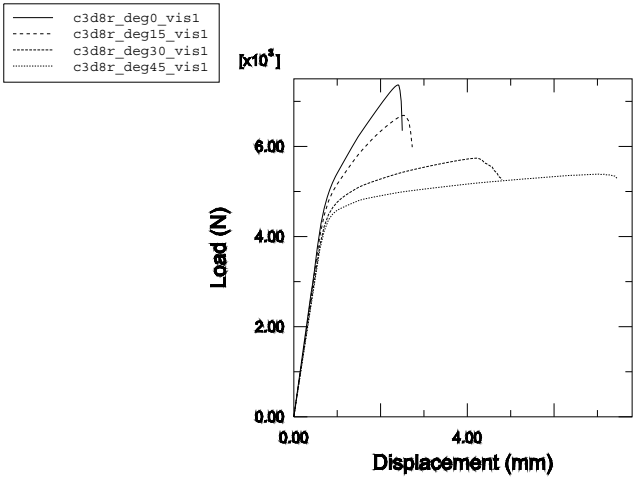


Figure 1.4.6–14 Load-displacement curves for different loading directions (using **UMAT** model).

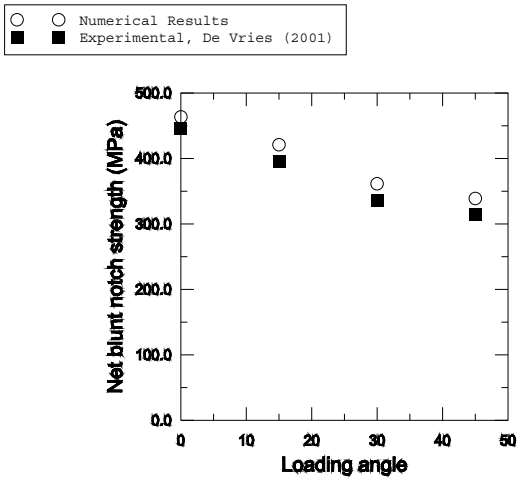


Figure 1.4.6–15 Calculated blunt notch strength for different loading angles in comparison with the experimental results (using **`UMAT`** model).

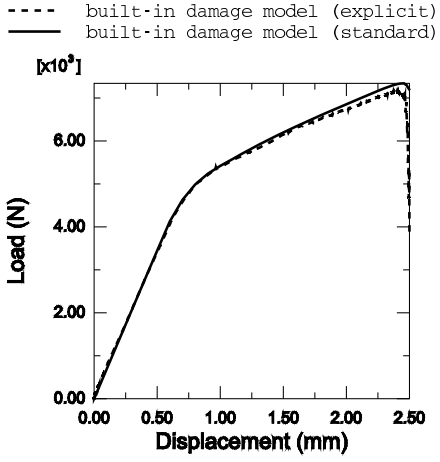


Figure 1.4.6–16 Load-displacement curves for the 0° loading direction: Abaqus/Explicit versus Abaqus/Standard.

1.4.7 DEBONDING BEHAVIOR OF A DOUBLE CANTILEVER BEAM

Products: Abaqus/Standard Abaqus/Explicit

Objectives

This example demonstrates the following Abaqus features and techniques:

- predicting debond growth in a double cantilever beam (DCB) using crack propagation analysis with the VCCT fracture criterion, cohesive elements, and surface-based cohesive behavior in Abaqus/Standard;
- demonstrating the use of contact clearance assignment between surfaces with the VCCT fracture criterion and surface-based cohesive behavior in Abaqus/Explicit to predict debond growth;
- restricting delamination growth and predicting the onset of debonding in models without a predefined crack tip using cohesive elements to supplement the VCCT fracture criterion for modeling the Z-pins and stitches that are commonly used in aerospace applications; and
- predicting progressive delamination growth at the interface in a double cantilever beam subjected to sub-critical cyclic loading using the low-cycle fatigue criterion based on the Paris law.

Application description

This example examines the debonding behavior of a double cantilever beam. Debond onset and growth are predicted for matched meshes in both Abaqus/Standard and Abaqus/Explicit and mismatched meshes in Abaqus/Standard. Different mesh discretizations are also used to investigate their effects on the debonding behavior. The results from Abaqus/Standard are compared with the results obtained using the VCCT-based fracture interface elements in Mabson (2003), as well as the results predicted by theory. The results predicted using VCCT, cohesive elements, and surface-based cohesive behavior in Abaqus/Standard are also compared.

The debonding behavior can also be studied by using the VCCT capability in Abaqus/Explicit. The model used in Abaqus/Explicit is constructed to achieve quasi-static behavior that allows the results obtained to be comparable with those generated using VCCT in Abaqus/Standard.

The same model is analyzed in Abaqus/Standard using the low-cycle fatigue criterion to assess the fatigue life when the model is subjected to sub-critical cyclic loading. The onset and delamination growth are characterized using the Paris law, which relates the relative fracture energy release rate to the crack growth rate. The fracture energy release rate at the crack tip is calculated based on the VCCT technique. The results from Abaqus are compared with those predicted by the theory in Tada (1985).

Geometry

The double cantilever beam in this example has a span of 9.0 in (228.6 mm) with a rectangular cross-section of 1.0 in (25.4 mm) wide × 0.4 in (10.2 mm) deep, as shown in Figure 1.4.7–1.

Boundary conditions and loading

One end of the beam is fixed, and the displacements are applied at the other end, as shown in Figure 1.4.7–1. The maximum displacement is set equal to 0.16 in (4.1 mm) in the monotonic loading cases. In the low-cycle fatigue analysis a cyclic displacement loading with a peak value of 0.05 in (1.3 mm) is specified.

Abaqus modeling approaches and simulation techniques

This example includes several two-dimensional models and one three-dimensional model.

Summary of analysis cases

Case 1 Prediction using matched and mismatched meshes for the two-dimensional DCB model.

Case 2 Comparison using different mesh discretizations for the two-dimensional DCB model.

Case 3 Theoretical and VCCT response prediction for the three-dimensional DCB model.

Case 4 Using cohesive elements with VCCT debond to model crack initiation.

Case 5 Using cohesive elements with VCCT debond to model Z-pins and stitches.

Case 6 Low-cycle fatigue prediction using the same model as in Case 1.

Case 7 Low-cycle fatigue prediction using the same model as in Case 3.

Case 8 Comparison of the results obtained using VCCT, cohesive elements, and cohesive behavior.

Case 9 Using cohesive behavior with VCCT in Abaqus/Explicit to model crack initiation. The model is identical to the model in Case 3.

Analysis types

Static stress analyses are performed for Cases 1–8. Dynamic analysis is used for Case 9.

Case 1 Prediction using matched and mismatched meshes for the two-dimensional DCB model

This case verifies that the simulation results agree with the experimental results and compares the accuracy of the results obtained using matched and mismatched meshes.

Mesh design

Four CPE4 elements are used to model the thickness of each half of the beam. The top and bottom figures in Figure 1.4.7–2 show the configurations of undeformed meshes used here with the initially bonded nodes. The model at the top has matched meshes with 90×4 meshes for each half of the DCB, while the model at the bottom has mismatched meshes with 90×4 meshes for the lower half and 85×4 meshes for the upper half of the DCB.

Results and discussion

The VCCT debond approach in Abaqus predicts the onset of debonding for both the matched and mismatched meshes accurately within 3% of the prediction by the fracture interface elements, as shown in Figure 1.4.7–3. Figure 1.4.7–3 also shows the growth prediction calculated by the VCCT debond approach.

Case 2 Comparison using different mesh discretizations for the two-dimensional DCB model

This case shows the effect of mesh refinement on the response.

Mesh design

Four CPE4 elements are used to model the thickness of each half of the beam, the same as in Case 1. The following mesh discretizations are used along the span of the beam:

1. Matched mesh with 90 × 4 mesh for each half of the DCB.
2. Matched mesh with 180 × 4 mesh for each half of the DCB.
3. Matched mesh with 360 × 4 mesh for each half of the DCB.

The meshes are displayed in Figure 1.4.7–2 (first three models).

Results and discussion

Figure 1.4.7–4 shows the response for different mesh discretizations of the two-dimensional model of the double cantilever beam and clearly illustrates the convergence of the response to the same solution with mesh refinement. The maximum energy release rate cutback tolerance is set to the default value of 0.2 in the input file provided for the two-dimensional model (**`dcb_vcct_2d_1.inp`**). However, to eliminate factors other than mesh size from the results shown in Figure 1.4.7–4, the comparisons were run with the cutback tolerance reduced to 0.1.

Case 3 Theoretical and VCCT response prediction for the three-dimensional DCB model

This case compares the three-dimensional results of Abaqus with the theoretical prediction.

Mesh design

The three-dimensional undeformed mesh and the initially bonded nodes are shown in Figure 1.4.7–5.

Results and discussion

Figure 1.4.7–6 shows a contour plot of the solution-dependent state variable BDSTAT for the three-dimensional model of the double cantilever beam. The debond growth is shown on the right side of the model.

The response predicted by the VCCT debond approach in Abaqus after the onset of debonding closely matches the theoretical results for the three-dimensional model of the double cantilever beam, as shown in Figure 1.4.7–7.

Case 4 Using cohesive elements with VCCT debond to model crack initiation

This case demonstrates how cohesive elements are used to initiate debond growth in a two-dimensional VCCT model.

Mesh design

A zero-thickness cohesive element (shown in Figure 1.4.7–8) is added at $x = 0$ at the interface between the two halves of the DCB. The cohesive element bonds the nodes at $x = 0$ at the contact interface. All the remaining nodes along the contact interface are initially bonded (see "Defining initially bonded crack surfaces in Abaqus/Standard" in "Crack propagation analysis," Section 11.4.3 of the Abaqus Analysis User's Guide).

Material model

The cohesive element properties are chosen such that the energy required for its complete failure equals the fracture toughness of the interface. A damage initiation criterion is specified for the cohesive element.

Case 5 Using cohesive elements with VCCT debond to model Z-pins and stitches

Z-pins and stitches are additional reinforcements used at the bonded interface to retard the delamination growth. These can be modeled using zero-thickness cohesive elements at the debond interface with the appropriate material and damage initiation and damage evolution characteristics of the reinforcement. In the DCB model the nodes at the interface (shown in Figure 1.4.7–9) are initially bonded (see "Defining initially bonded crack surfaces in Abaqus/Standard" in "Crack propagation analysis," Section 11.4.3 of the Abaqus Analysis User's Guide). In addition, a layer of cohesive elements is defined at the interface using the initially bonded nodes to represent the Z-pins and stitches.

Case 6 Low-cycle fatigue prediction using the same model as in Case 1

This case verifies that delamination growth in a two-dimensional DCB model subjected to sub-critical cyclic loading can be predicted using the low-cycle fatigue criterion. The simulation results are compared with the theoretical results.

Mesh design

The model has matched meshes the same as in Case 1 (model at the top of Figure 1.4.7–2).

Results and discussion

The results in terms of crack length versus the cycle number obtained using the low-cycle fatigue criterion in Abaqus are compared with the theoretical results in Figure 1.4.7–10. Reasonably good agreement is obtained.

Case 7 Low-cycle fatigue prediction using the same model as in Case 3

This case verifies that delamination growth in a three-dimensional DCB model subjected to sub-critical cyclic loading can be predicted using the low-cycle fatigue criterion. The simulation results are compared with the theoretical results.

Mesh design

The mesh design is the same as in Case 3 (Figure 1.4.7–5).

Results and discussion

The results in terms of crack length versus the cycle number obtained using the low-cycle fatigue criterion in Abaqus are compared with the theoretical results in Figure 1.4.7–11. Reasonably good agreement is obtained.

Case 8 Comparison of the results obtained using VCCT, cohesive elements, and cohesive behavior

This case compares the results obtained using VCCT, cohesive elements, and surface-based cohesive behavior for both the two- and three-dimensional DCB models.

Mesh design

For the two-dimensional model, four CPE4 elements are used to model the thickness of each half of the beam. The model at the top of Figure 1.4.7–2 shows the configurations of undeformed meshes used here with the initially bonded nodes. The model has matched meshes with 90×4 meshes for each half of the DCB.

The three-dimensional undeformed mesh and the initially bonded nodes are shown in Figure 1.4.7–5.

Results and discussion

Figure 1.4.7–12 compares the results obtained using VCCT, cohesive elements, and surface-based cohesive behavior for the two-dimensional DCB model. Figure 1.4.7–13 shows a similar comparison for the three-dimensional DCB model. All of the methods predicted nearly the same debond growth after debond onset. But the responses predicted using cohesive elements or surface-based cohesive behavior are not strictly linear before debond onset, as compared with the linear behavior predicted by the VCCT debond approach. This is not surprising since the slave nodes for the model using the VCCT debond approach are released one after another, while all the slave nodes for the models using cohesive elements or surface-based cohesive behavior debond simultaneously.

Case 9 Using VCCT in Abaqus/Explicit to model crack initiation

This case verifies that delamination growth in a three-dimensional DCB model can be predicted using the VCCT capability in Abaqus/Explicit. The simulation results are compared with Abaqus/Standard VCCT results.

Mesh design

The mesh design is the same as in Case 3 (Figure 1.4.7–5).

Results and discussion

The debond states obtained from the Abaqus/Explicit and Abaqus/Standard analyses match very well, as depicted in Figure 1.4.7–14. The measured reaction forces from the Abaqus/Explicit analysis compare extremely well with those from Abaqus/Standard, as shown in Figure 1.4.7–15. The debonding time, displacement, and energy release rate are highly consistent between both analyses.

Input files

dcb_vcct_2d_1.inp	Two-dimensional model with matched mesh; 90 × 4 mesh for each half of the DCB.
dcb_mismatch_vcct_2d_1.inp	Two-dimensional model with mismatched meshes.
dcb_coh_init_vcct_2d_1.inp	Two-dimensional model with 4-node two-dimensional cohesive elements (COH2D4) to predict the onset of debonding.
dcb_coh_stitch_vcct_2d_1.inp	Two-dimensional model with 4-node two-dimensional cohesive elements (COH2D4) representing Z-pins and stitches.
dcb_vcct_3d_1.inp	Three-dimensional model with matched meshes.
dcb_vcct_fatigue_2d.inp	Same as dcb_vcct_2d_1.inp but subjected to cyclic displacement loading.
dcb_vcct_fatigue_3d.inp	Same as dcb_vcct_3d_1.inp but subjected to cyclic displacement loading.
dcb_cohelm_2d.inp	Two-dimensional model using cohesive elements with matched mesh; 90 × 4 mesh for each half of the DCB.
dcb_surcoh_2d.inp	Two-dimensional model using surface-based cohesive behavior with matched mesh; 90 × 4 mesh for each half of the DCB.
dcb_cohelm_3d.inp	Three-dimensional model using cohesive elements with matched meshes.
dcb_surcoh_3d.inp	Three-dimensional model using surface-based cohesive behavior with matched meshes.
dcb_vcct_xpl_3d.inp	Abaqus/Explicit three-dimensional model with matched meshes.

References

Abaqus Analysis User's Guide

- "Low-cycle fatigue analysis using the direct cyclic approach," Section 6.2.7 of the Abaqus Analysis User's Guide
- "Crack propagation analysis," Section 11.4.3 of the Abaqus Analysis User's Guide

Abaqus Keywords Reference Guide

- *COHESIVE BEHAVIOR
- *CONTACT CLEARANCE ASSIGNMENT
- *DEBOND
- *DIRECT CYCLIC
- *FRACTURE CRITERION

Other

- Mabson, G, "Fracture Interface Elements," 46th PMC General Session of Mil-17 (Composites Materials Handbook) Organization, Charleston, SC, 2003.
- Tada, H, "The Stress Analysis of Cracks Handbook," Paris Productions Incorporated, 1985.

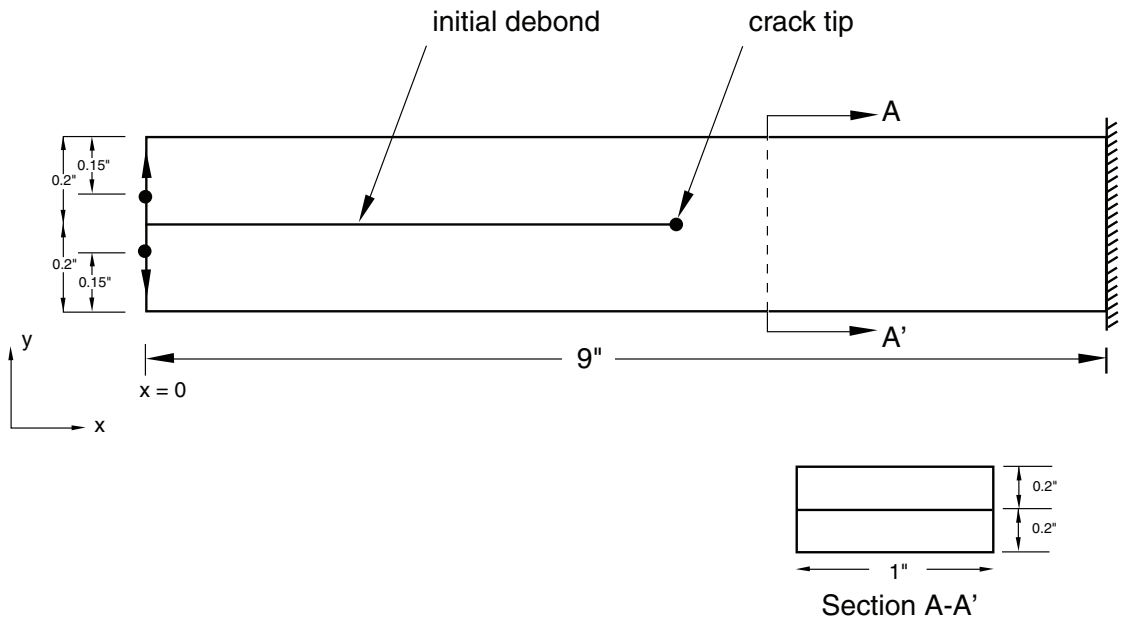


Figure 1.4.7–1 The double cantilever beam (DCB) model.

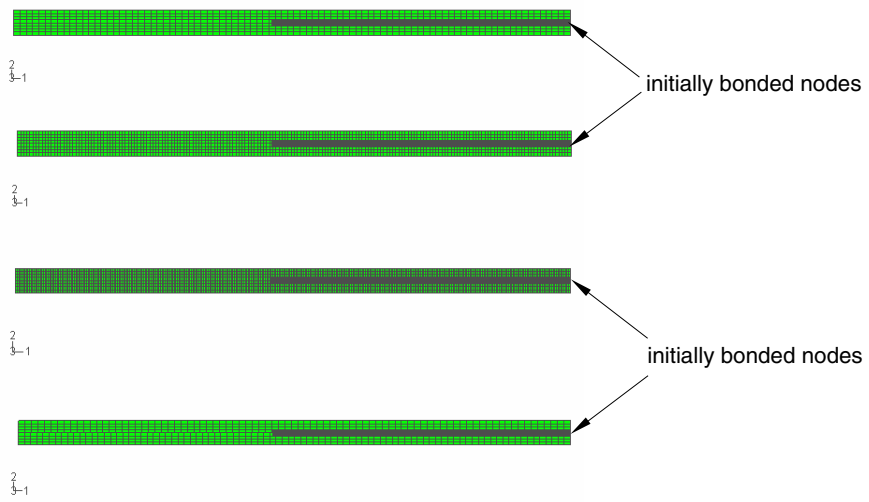


Figure 1.4.7–2 Mesh configurations for the two-dimensional DCB model.

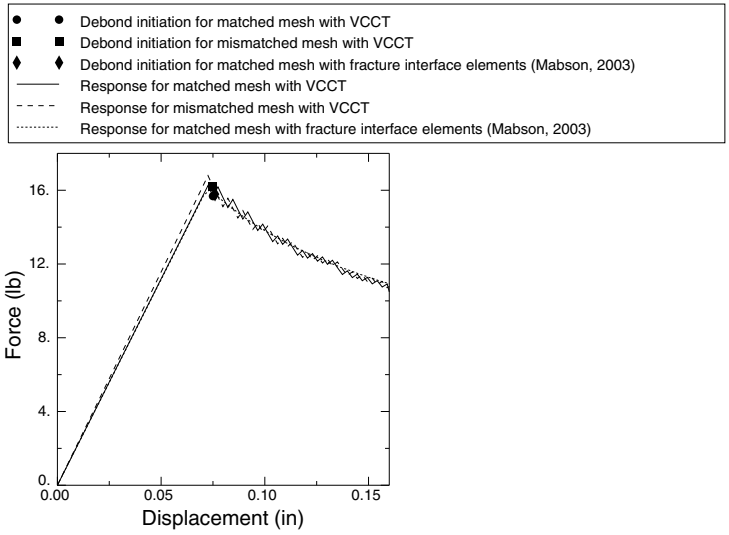


Figure 1.4.7–3 Debond onset and growth prediction for matched and mismatched meshes for the two-dimensional DCB model.

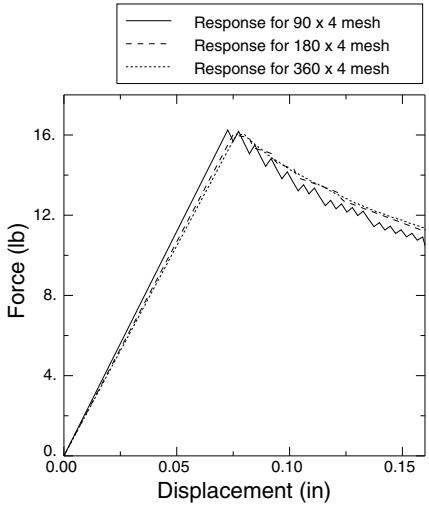


Figure 1.4.7–4 Response for different mesh discretizations of the two-dimensional DCB model.

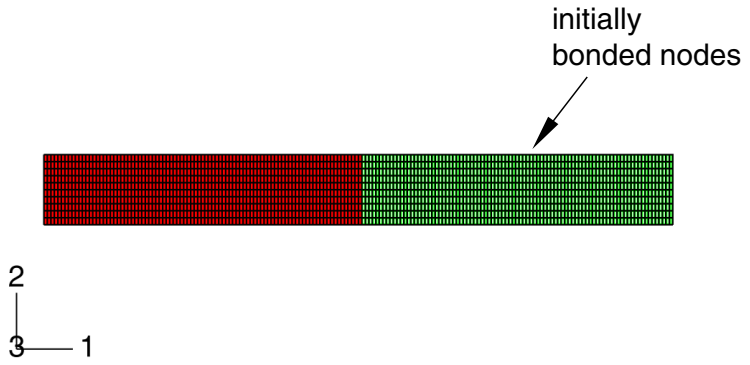


Figure 1.4.7–5 Mesh configuration for the three-dimensional DCB model.

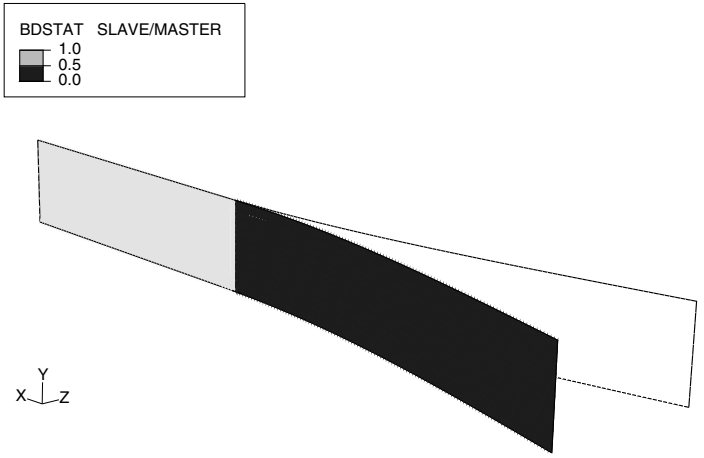


Figure 1.4.7–6 Debond growth for the three-dimensional DCB model.

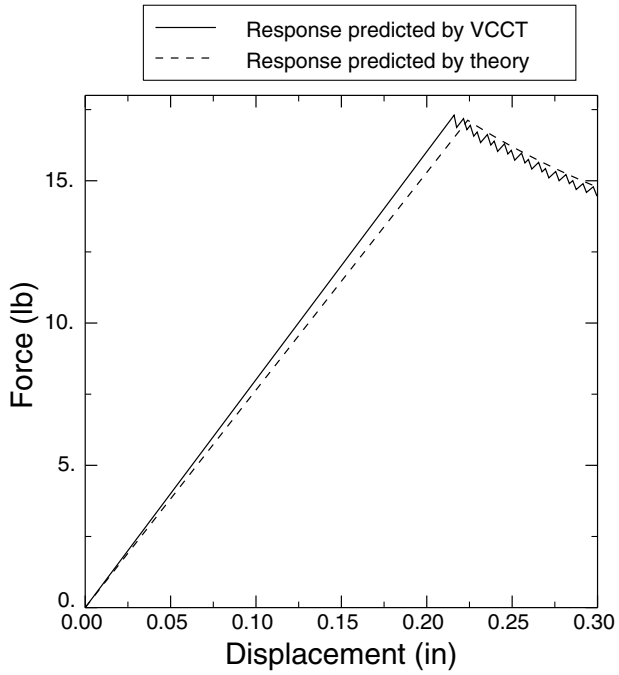


Figure 1.4.7–7 Theoretical and VCCT in Abaqus response prediction for the three-dimensional DCB model.



Figure 1.4.7–8 Using zero-thickness cohesive elements to model crack initiation.



Figure 1.4.7–9 Modeling Z-pins and stitches using cohesive elements.

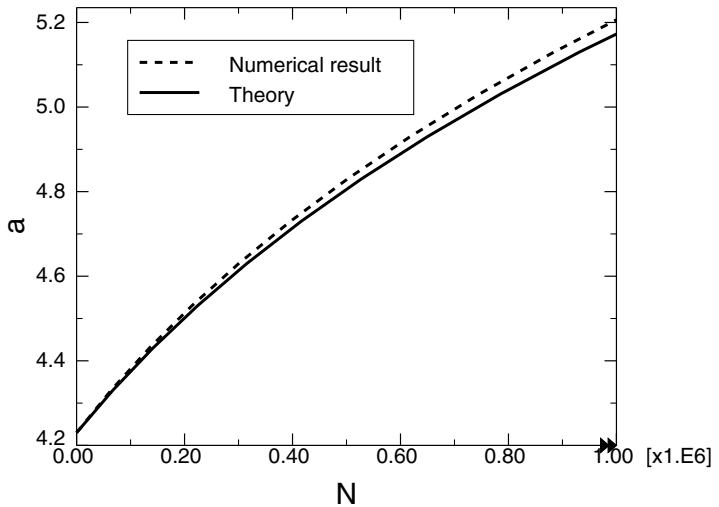


Figure 1.4.7–10 Crack length versus cycle number for the two-dimensional DCB model.

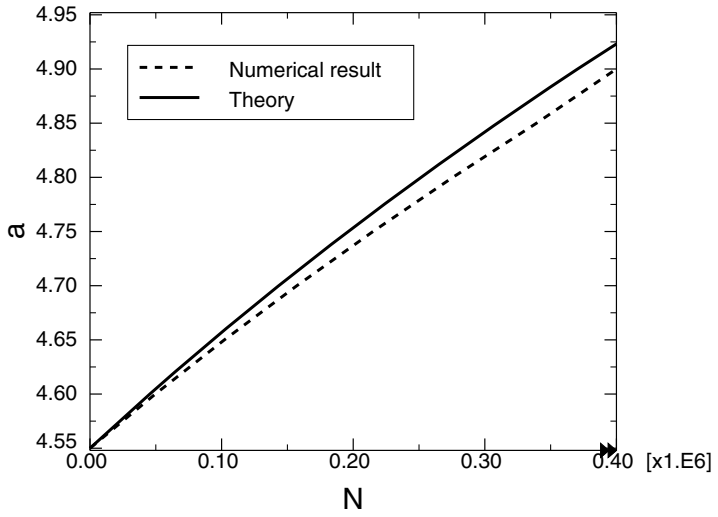


Figure 1.4.7–11 Crack length versus cycle number for the three-dimensional DCB model.

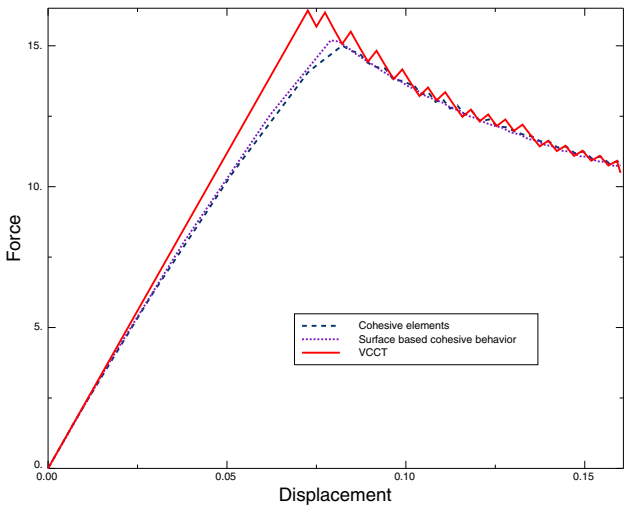


Figure 1.4.7–12 Comparison of the results using VCCT, cohesive elements, and surface-based cohesive behavior for the two-dimensional DCB model.

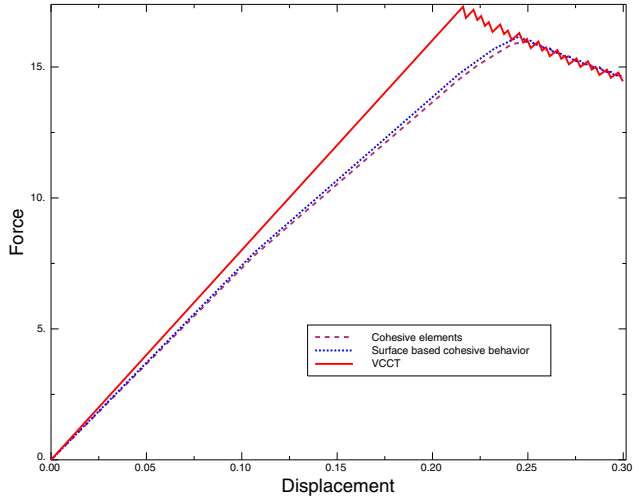


Figure 1.4.7–13 Comparison of the results using VCCT, cohesive elements, and surface-based cohesive behavior for the three-dimensional DCB model.

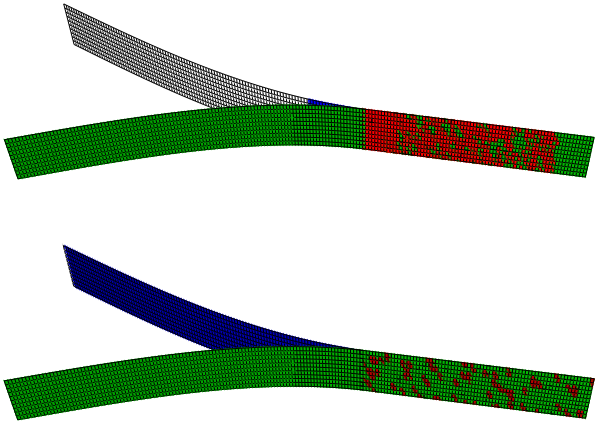


Figure 1.4.7–14 Debond state comparison between Abaqus/Explicit (top) and Abaqus/Standard (bottom).

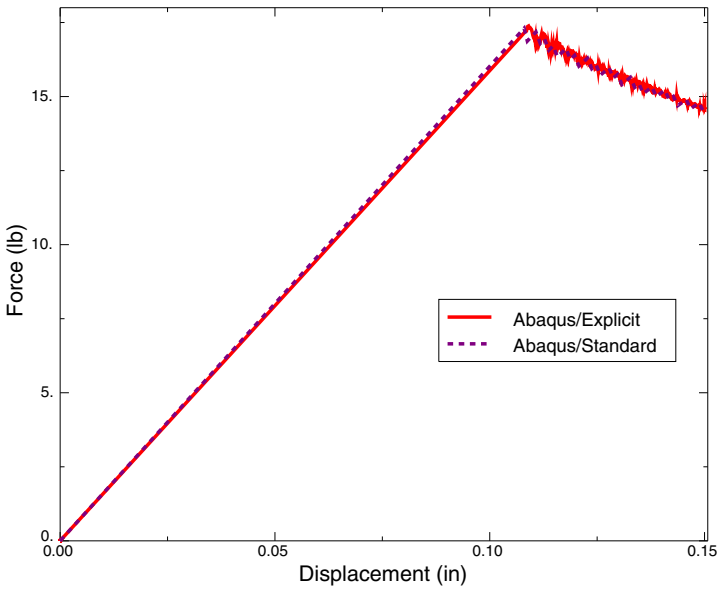


Figure 1.4.7–15 Comparison of the results between Abaqus/Explicit and Abaqus/Standard.

1.4.8 DEBONDING BEHAVIOR OF A SINGLE LEG BENDING SPECIMEN

Products: Abaqus/Standard Abaqus/Explicit

Objectives

This example demonstrates the following Abaqus features and techniques:

- predicting debond growth in a single leg bending (SLB) specimen using crack propagation analysis with the VCCT fracture criterion in Abaqus/Standard;
- predicting debond growth using the VCCT fracture criterion and surface-based cohesive behavior in Abaqus/Explicit; and
- predicting progressive delamination growth at the interface in this specimen subjected to sub-critical cyclic loading using the low-cycle fatigue criterion based on the Paris law.

Application description

The example examines the debonding behavior of a single leg bending specimen and compares the simulation results with the results of the analysis performed using the VCCT-based fracture interface elements discussed in Mabson (2003). Damping is also used in a two-dimensional model to demonstrate how it can stabilize the crack growth.

The same model is analyzed using the low-cycle fatigue criterion to assess the fatigue life when the model is subjected to sub-critical cyclic loading. The onset and delamination growth are characterized using the Paris law, which relates the relative fracture energy release rate to the crack growth rate. The fracture energy release rate at the crack tip is calculated based on the VCCT technique. The results from Abaqus are compared with those predicted by the theory in Davidson (1995).

Geometry

The geometries of the two-dimensional and three-dimensional single leg bending specimens with their corresponding initial crack locations are shown in Figure 1.4.8–1 and Figure 1.4.8–2, respectively.

Boundary conditions and loading

A displacement is applied on the top beam at the location shown in Figure 1.4.8–1 for the two-dimensional model and in Figure 1.4.8–2 for the three-dimensional model. The displacement results in a mixture of opening (mode I) and shearing (mode II) modes. The maximum displacements are set equal to 0.32 in (8.13 mm) in the two-dimensional model and 0.15 in (3.81 mm) in the three-dimensional model for the monotonic loading cases. In the low-cycle fatigue analyses, cyclic displacement loadings with peak values of 0.12 in (3.05 mm) in the two-dimensional model and 0.035 in (0.89 mm) in the three-dimensional model are specified.

Abaqus modeling approaches and simulation techniques

This example includes one two-dimensional model and one three-dimensional model.

Summary of analysis cases

Case 1 Two-dimensional single leg bending model.

Case 2 Three-dimensional single leg bending model.

Case 3 Low-cycle fatigue prediction using the same model as in Case 1.

Case 4 Low-cycle fatigue prediction using the same model as in Case 2.

Case 5 Abaqus/Explicit with VCCT using the same model as in Case 2.

Analysis types

Static analyses are performed for Cases 1–4. Dynamic analysis is used for Case 5.

Case 1 Two-dimensional single leg bending model

This case compares the Abaqus VCCT results for the two-dimensional model with the results of Mabson (2003). The damping effects are also examined.

Mesh design

The model uses a finite element mesh of 4-node bilinear plane strain quadrilateral, incompatible mode elements (CPE4I) for both the long and short beams.

Results and discussion

Figure 1.4.8–3 shows the deformed configuration of the two-dimensional model. Figure 1.4.8–4 shows a contour plot of bond status variable BDSTAT that illustrates the debonding growth in a two-dimensional model. The region of debonding is shown on the right side of the model. Figure 1.4.8–5 compares the results from the two-dimensional analysis with the results of the analysis performed using the VCCT-based fracture interface elements discussed in Mabson (2003).

The damping effect to a two-dimensional simulation is also examined in this case by adding damping at the debond interface (see "Automatic stabilization of rigid body motions in contact problems" in "Adjusting contact controls in Abaqus/Standard," Section 36.3.6 of the Abaqus Analysis User's Guide). The damping stabilizes the crack growth and allows the solution to converge. It is important to assess how much energy has been used for numerical damping by comparing the stabilization energy to the strain energy of the model. Figure 1.4.8–6 shows the comparison between static stabilization energy (ALLSD) and the strain energy of the model (ALLSE). The comparison indicates that the maximum static stabilization energy is less than 3% of the maximum strain energy of the model during the analysis. This value is reasonable and indicates that the solution has not been affected by the addition of artificial numerical damping.

Case 2 Three-dimensional single leg bending model

This case compares the Abaqus VCCT results for the three-dimensional model with a theoretical prediction.

Mesh design

The model uses fully integrated first-order shell elements (S4) for both the long and short beams.

Results and discussion

Figure 1.4.8–7 shows a contour plot of bond status variable BDSTAT for the three-dimensional model. Figure 1.4.8–8 shows a comparison between the results from the three-dimensional analysis and the results presented in Mabson (2003).

Case 3 Low-cycle fatigue prediction using the same model as in Case 1

This case verifies that delamination growth in a two-dimensional single leg bending model subjected to sub-critical cyclic loading can be predicted using the low-cycle fatigue criterion. The simulation results are compared with the theoretical results.

Mesh design

The mesh design is the same as in Case 1.

Results and discussion

The results in terms of crack length versus the cycle number obtained using the low-cycle fatigue criterion in Abaqus are compared with the theoretical results in Figure 1.4.8–9. Reasonably good agreement is obtained.

Case 4 Low-cycle fatigue prediction using the same model as in Case 2

This case verifies that delamination growth in a three-dimensional single leg bending model subjected to sub-critical cyclic loading can be predicted using the low-cycle fatigue criterion. The simulation results are compared with the theoretical results.

Mesh design

The mesh design is the same as in Case 2.

Results and discussion

The results in terms of crack length versus the cycle number obtained using the low-cycle fatigue criterion in Abaqus are compared with the theoretical results in Figure 1.4.8–10. Reasonably good agreement is obtained.

Case 5 Using VCCT in Abaqus/Explicit to model crack initiation

This case verifies that delamination growth can be predicted using Abaqus/Explicit. The simulation results are compared with results obtained from Abaqus/Standard. To reduce inertia effects and allow a better comparison between Abaqus/Standard and Abaqus/Explicit results, the material density was lowered and the loading was ramped on with a smooth step definition.

Mesh design

The mesh design is the same as in Case 2.

Results and discussion

The results obtained using Abaqus/Explicit with VCCT show reasonably good agreement with those obtained from Abaqus/Standard, as depicted in Figure 1.4.8–11 and Figure 1.4.8–12. Due to the thin layer of elements and the specified boundary conditions in the model, inertia effects are clearly observed in the measured reaction forces. However, the peak forces at debond onset, debond time, and other VCCT output quantities are consistent between the two analyses. The reaction forces obtained in Abaqus/Explicit were filtered with a Butterworth filter with a cutoff frequency of 500 Hz to reduce high-frequency oscillations from the response curve.

Input files

slb_vcct_2d_1.inp	Two-dimensional model of the SLB specimen.
slb_vcct_3d_1.inp	Three-dimensional model of the SLB specimen.
slb_vcct_fatigue_2d.inp	Same as slb_vcct_2d_1.inp but subjected to cyclic displacement loading.
slb_vcct_fatigue_3d.inp	Same as slb_vcct_3d_1.inp but subjected to cyclic displacement loading.
slb_vcct_xpl_3d.inp	Three-dimensional model of the SLB specimen using Abaqus/Explicit with VCCT.

References

Abaqus Analysis User's Guide

- "Low-cycle fatigue analysis using the direct cyclic approach," Section 6.2.7 of the Abaqus Analysis User's Guide
- "Crack propagation analysis," Section 11.4.3 of the Abaqus Analysis User's Guide

Abaqus Keywords Reference Guide

- *COHESIVE BEHAVIOR
- *CONTACT CLEARANCE ASSIGNMENT
- *DEBOND

- *DIRECT CYCLIC
- *FRACTURE CRITERION
- *NODAL ENERGY RATE

Other

- Mabson, G, "Fracture Interface Elements," 46th PMC General Session of Mil-17 (Composites Materials Handbook) Organization, Charleston, SC, 2003.
- Davidson, B. D., R. Kruger, and M. Konig, "Three-Dimensional Analysis of Center-Delaminated Unidirectional and Multidirectional Single-Leg Bending Specimens," Composites Science and Technology, vol. 54, pp. 385–394, 1995.

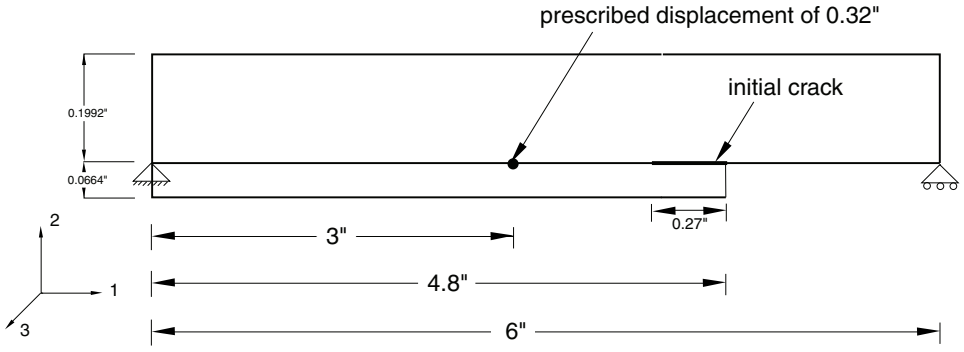


Figure 1.4.8–1 The two-dimensional single leg bending (SLB) specimen.

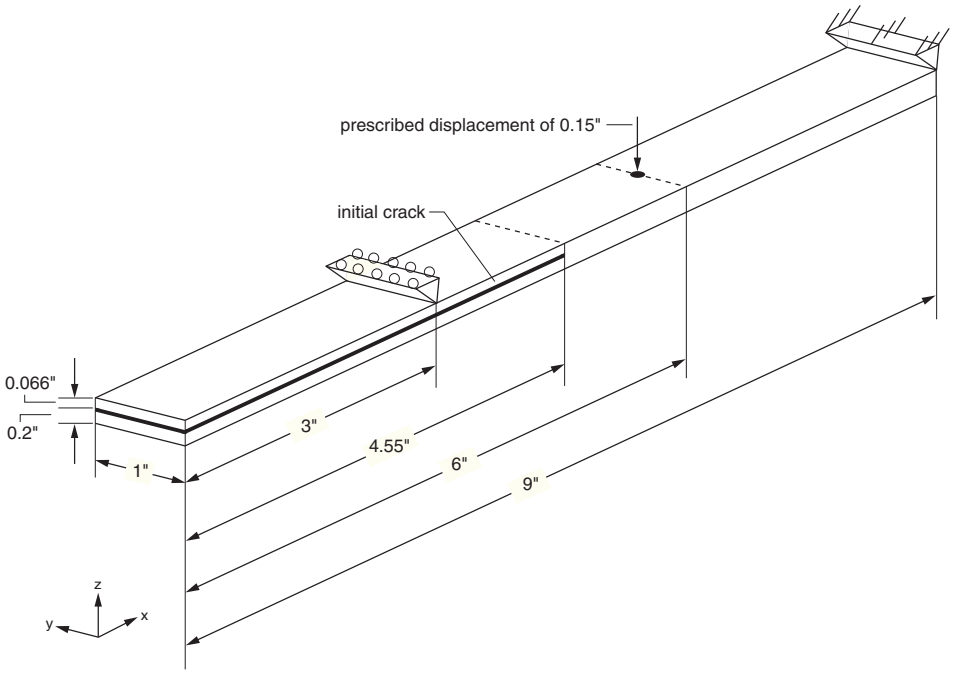


Figure 1.4.8–2 The three-dimensional single leg bending (SLB) specimen.

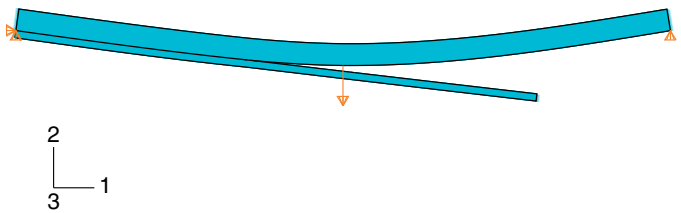


Figure 1.4.8–3 The deformed shape of the two-dimensional model showing boundary conditions and prescribed displacements.

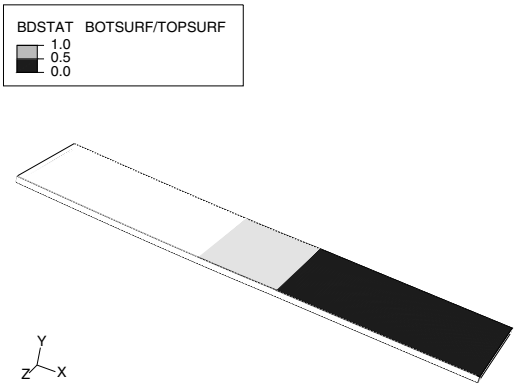


Figure 1.4.8–4 The prediction of debonding growth for the two-dimensional SLB model.

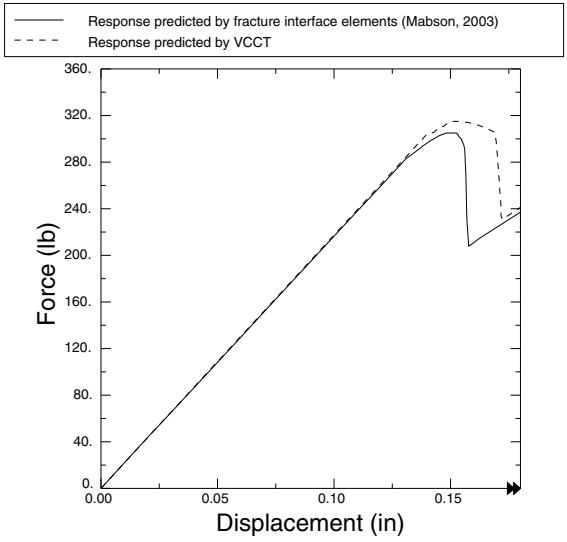


Figure 1.4.8–5 Response prediction for the two-dimensional SLB model.

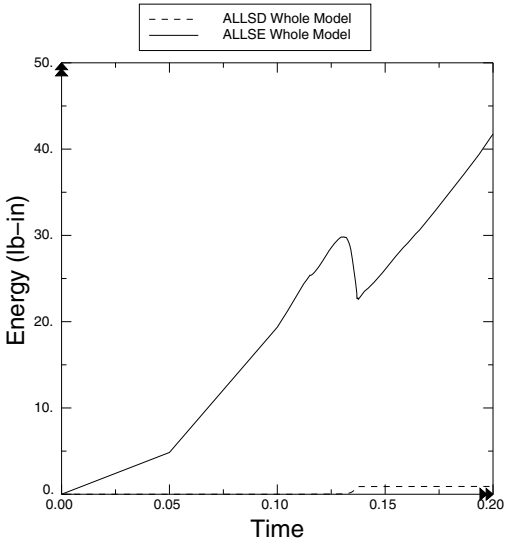


Figure 1.4.8–6 A comparison between ALLSD and ALLSE for the two-dimensional model.

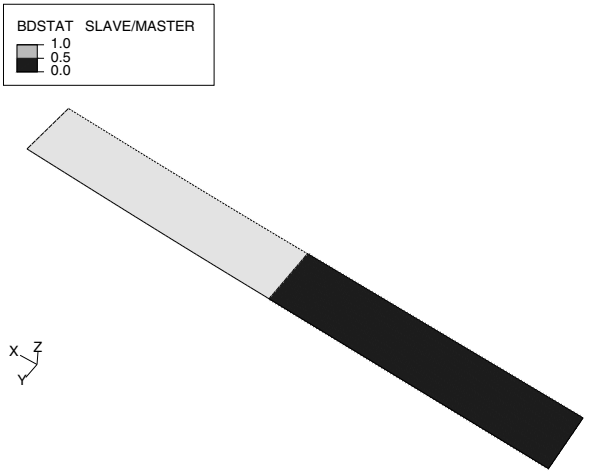


Figure 1.4.8–7 The prediction of debonding growth for the three-dimensional SLB model.

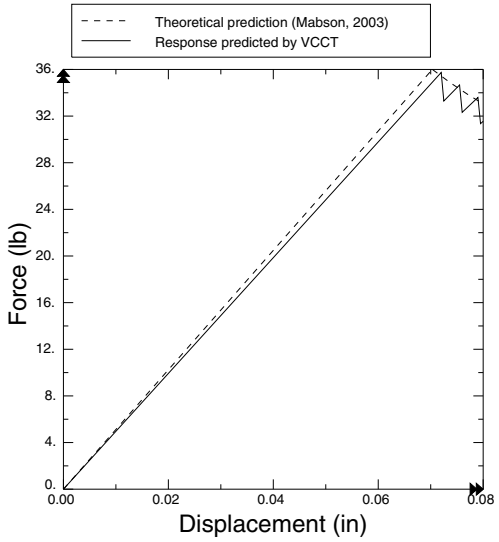


Figure 1.4.8–8 Response prediction for the three-dimensional SLB model.

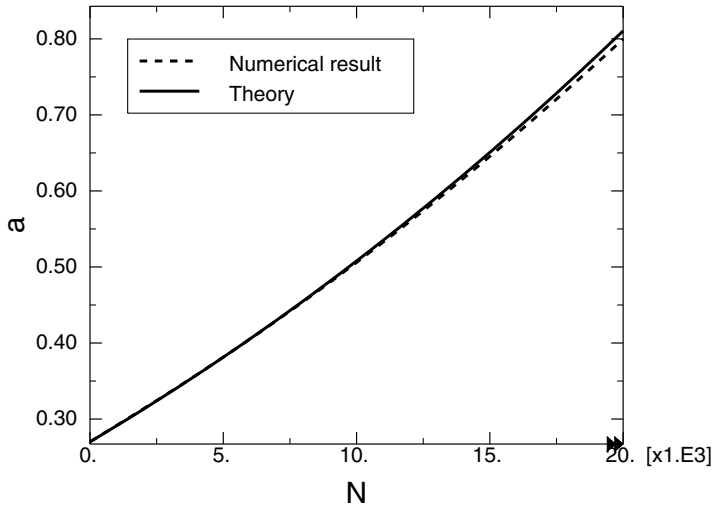


Figure 1.4.8–9 Crack length versus cycle number for the two-dimensional SLB model.

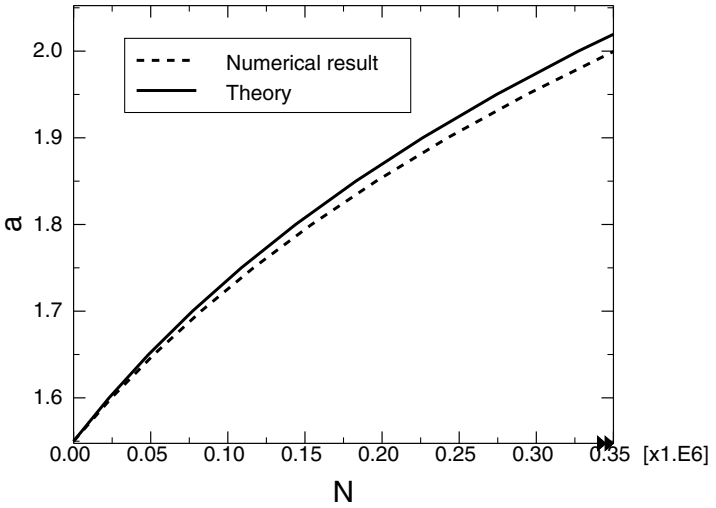


Figure 1.4.8–10 Crack length versus cycle number for the three-dimensional SLB model.

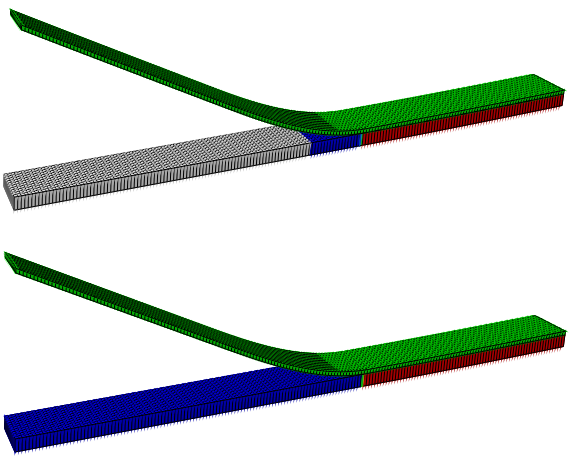


Figure 1.4.8–11 Debond state comparison between Abaqus/Explicit (top) and Abaqus/Standard (bottom).

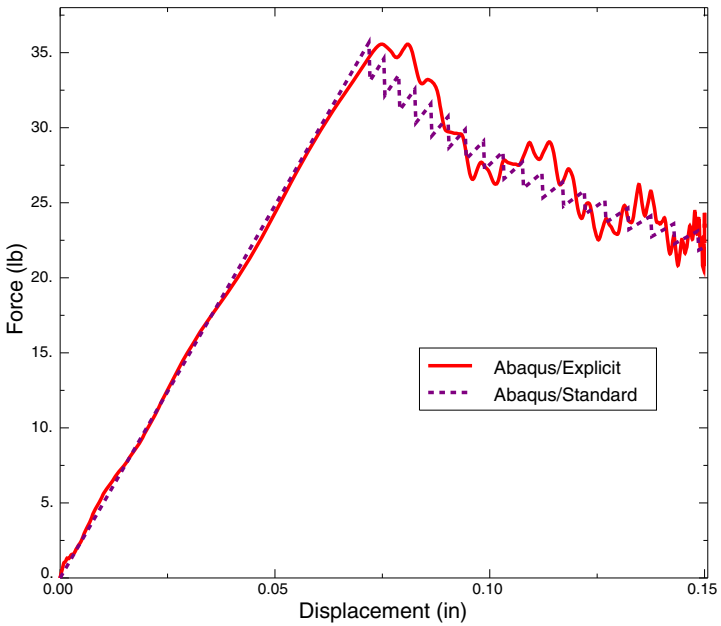


Figure 1.4.8–12 Comparison of the results between Abaqus/Explicit and Abaqus/Standard.

1.4.9 POSTBUCKLING AND GROWTH OF DELAMINATIONS IN COMPOSITE PANELS

Products: Abaqus/Standard Abaqus/Explicit

Objectives

This example illustrates the application of the use of VCCT fracture criterion in both Abaqus/Standard and Abaqus/Explicit to predict the postbuckling response, onset, and growth of delaminations in laminated composite panels.

Application description

Delaminations are a primary failure mode for laminated composite materials. The delamination growth is more prominent under compressive loading since it results in buckling of a sublaminate leading to the delamination growth. The particular problem considered here is described in Reeder (2002). The results from the VCCT debond approach in Abaqus are compared to the experimental results.

Geometry

A flat 9.0 in (228.6 mm) × 4.5 in (114.3 mm) composite panel with a centrally located 2.5 in (63.5 mm) diameter delamination is studied in this example, as shown in Figure 1.4.9–1.

Materials

The panel is made of a AS4/3501-6 graphite/epoxy composite material system for which the typical lamina properties are given in Table 1.4.9–1. The laminate stacking sequence for the panel is $[(\pm 45/90/0)_2/\pm 60/\pm 15]_S$. The critical fracture toughness for Modes I, II, and III at the delamination interface are also given in Table 1.4.9–1.

Boundary conditions and loading

The panel is subjected to compressive loading along its long axis. The overall dimensions of the model with boundary conditions and loading can be seen in Figure 1.4.9–1.

Abaqus modeling approaches and simulation techniques

The delamination is placed at the interface between the 5th and 6th ply (between 45° and −45°). The delamination region is modeled using two superimposed shell elements with contact constraints defined to prevent penetration of elements. The Benzeggagh-Kenane mixed-mode failure criterion (Benzeggagh and Kenane, 1996) is used to determine the growth of delamination based on the strain energy release rate computed using VCCT.

Analysis types

Both static and dynamic analyses are performed.

Mesh design

The finite element model is created with fully integrated first-order shell elements (S4). The finite element mesh for the model is shown in Figure 1.4.9–2 with the circular delamination at the center.

Loading

The loading consists of a prescribed displacement of 0.03 in (0.76 mm) at the top edge of the panel.

Solution controls

To achieve a stable delamination growth in Abaqus/Standard, a small amount of damping is specified for the interface where delamination growth occurs (see "Automatic stabilization of rigid body motions in contact problems" in "Adjusting contact controls in Abaqus/Standard," Section 36.3.6 of the Abaqus Analysis User's Guide). There are no solution controls specified in the Abaqus/Explicit analysis.

Results and discussion

The final deformed configuration of the composite panel obtained from Abaqus/Standard is shown in Figure 1.4.9–3. The postbuckling in the panel section cut along the long axis is illustrated in Figure 1.4.9–4. A contour plot of the bond status variable BDSTAT obtained from Abaqus/Standard indicating the growth of delamination is shown in Figure 1.4.9–5. The load-strain predictions are compared with the experimental data presented by Reeder (2002) in Figure 1.4.9–6. The predictions from VCCT are in agreement with the experimental results. The onset of delamination growth predicted by the VCCT debond approach in Abaqus is within 10% of the experimental data. The energy dissipated to stabilize the delamination growth is less than 4% of the total strain energy, as shown in Figure 1.4.9–7.

The deformed configurations obtained from Abaqus/Explicit are shown in Figure 1.4.9–8 and Figure 1.4.9–9. Delamination growth obtained from the Abaqus/Explicit analysis is shown in Figure 1.4.9–10. The force-displacement responses differ slightly from the Abaqus/Standard results, as shown in Figure 1.4.9–11, but show reasonable agreement.

Input files

nasa_postbuckle_vcct_1.inp	Postbuckling analysis of a composite plate using Abaqus/Standard.
nasa_postbuckle_xpl_vcct.inp	Postbuckling analysis of a composite plate using Abaqus/Explicit.

References

Abaqus Analysis User's Guide

- "Crack propagation analysis," Section 11.4.3 of the Abaqus Analysis User's Guide

Abaqus Keywords Reference Guide

- *COHESIVE BEHAVIOR
- *CONTACT CLEARANCE ASSIGNMENT

- *DEBOND
- *FRACTURE CRITERION

Other

- Benzeggagh, M., and M. Kenane, "Measurement of Mixed-Mode Delamination Fracture Toughness of Unidirectional Glass/Epoxy Composites with Mixed-Mode Bending Apparatus," Composite Science and Technology, vol. 56, p. 439, 1996.
- Reeder, J., S. Kyongchan, P. B. Chunchu, and D. R. Ambur, "Postbuckling and Growth of Delaminations in Composite Plates Subjected to Axial Compression," 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Denver, Colorado, vol. 1746, p. 10, 2002.

Table 1.4.9–1 Properties for AS4/3501-6 graphite/epoxy material.

Property	Value
E_1	18.500×10^6 lb/in^2 (127.554 kN/mm^2)
E_2	1.640×10^6 lb/in^2 (11.307 kN/mm^2)
G_{12}	0.871×10^6 lb/in^2 (6.005 kN/mm^2)
G_{13}	0.871×10^6 lb/in^2 (6.005 kN/mm^2)
G_{23}	0.522×10^6 lb/in^2 (3.599 kN/mm^2)
ν_{12}	0.30
G_{IC}	0.46863 lb/in (0.08207 N/mm)
G_{IIC}	3.171825 lb/in (0.55546 N/mm)
G_{IIIC}	3.171825 lb/in (0.55546 N/mm)

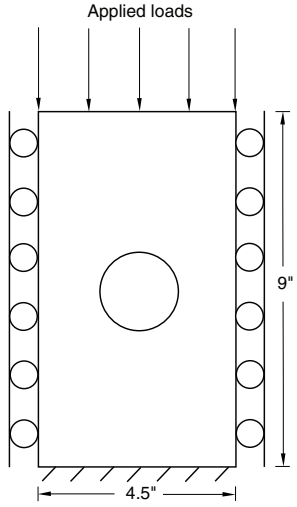


Figure 1.4.9–1 The flat composite panel.

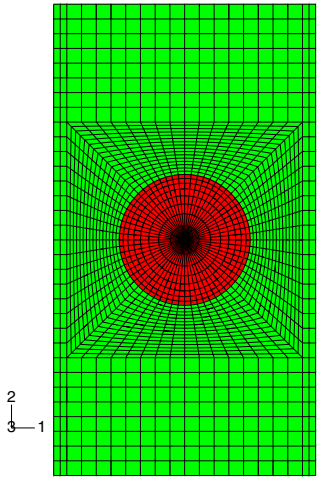


Figure 1.4.9–2 The meshed NASA panel model.

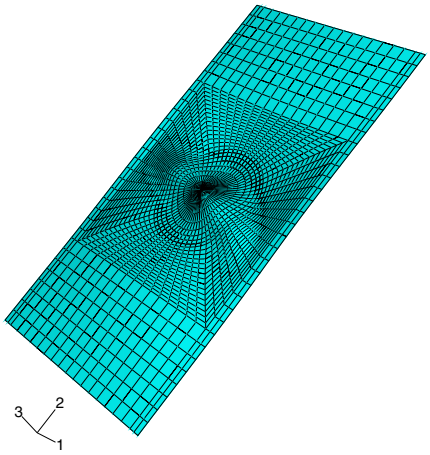


Figure 1.4.9–3 The final deformed configuration (Abaqus/Standard).

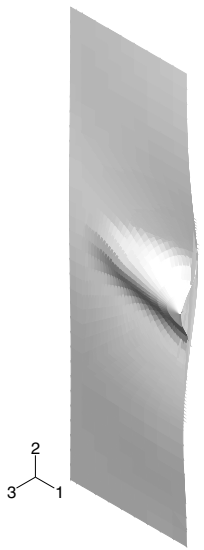


Figure 1.4.9–4 The postbuckling in the panel section (Abaqus/Standard).

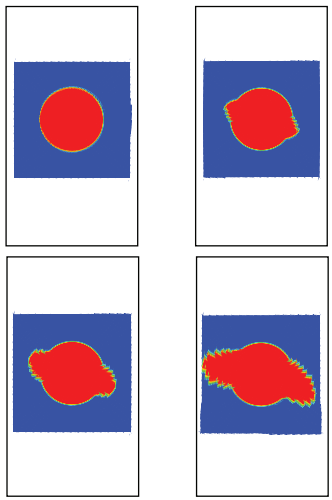


Figure 1.4.9–5 The growth of delamination (Abaqus/Standard).

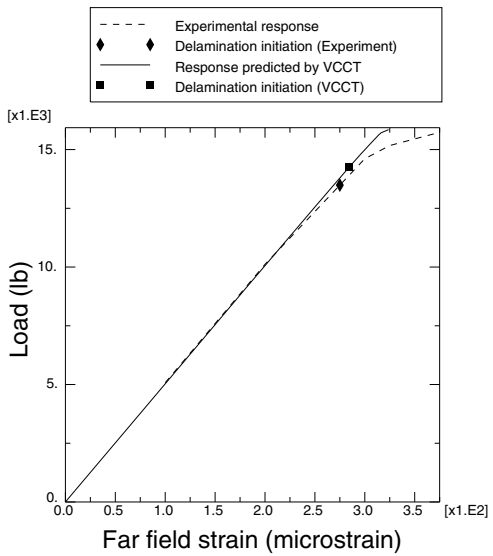


Figure 1.4.9–6 The load-strain predictions compared with experimental data.

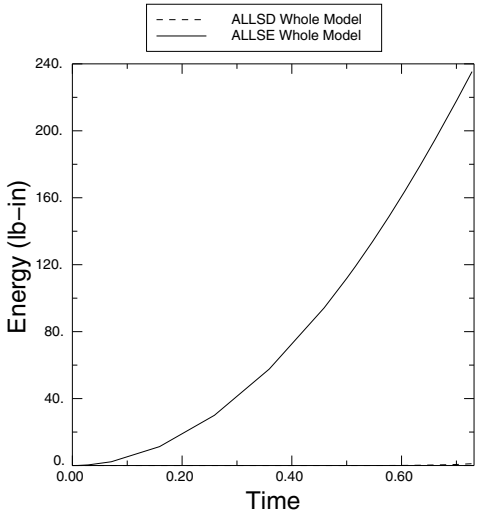


Figure 1.4.9–7 The energy dissipated to stabilize the delamination growth.

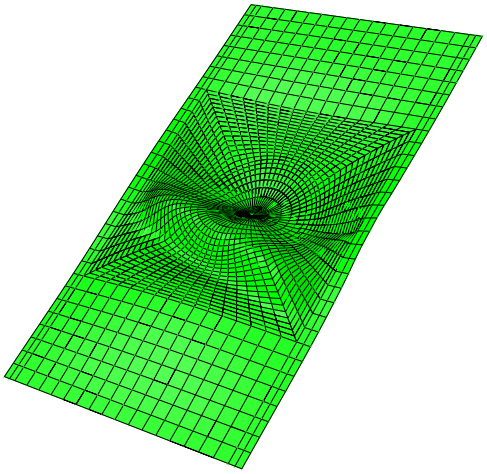


Figure 1.4.9–8 The final deformed configuration (Abaqus/Explicit).

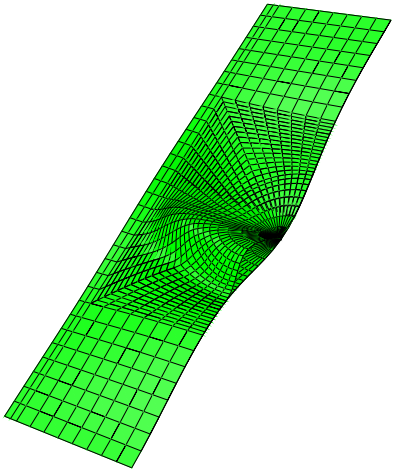


Figure 1.4.9–9 The postbuckling in the panel section (Abaqus/Explicit).

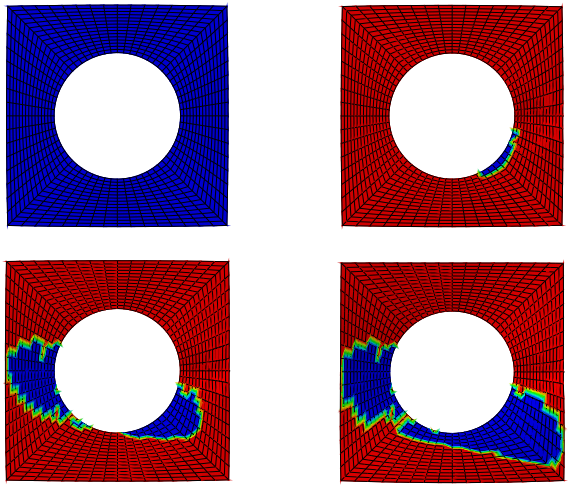


Figure 1.4.9–10 The growth of delamination (Abaqus/Explicit).

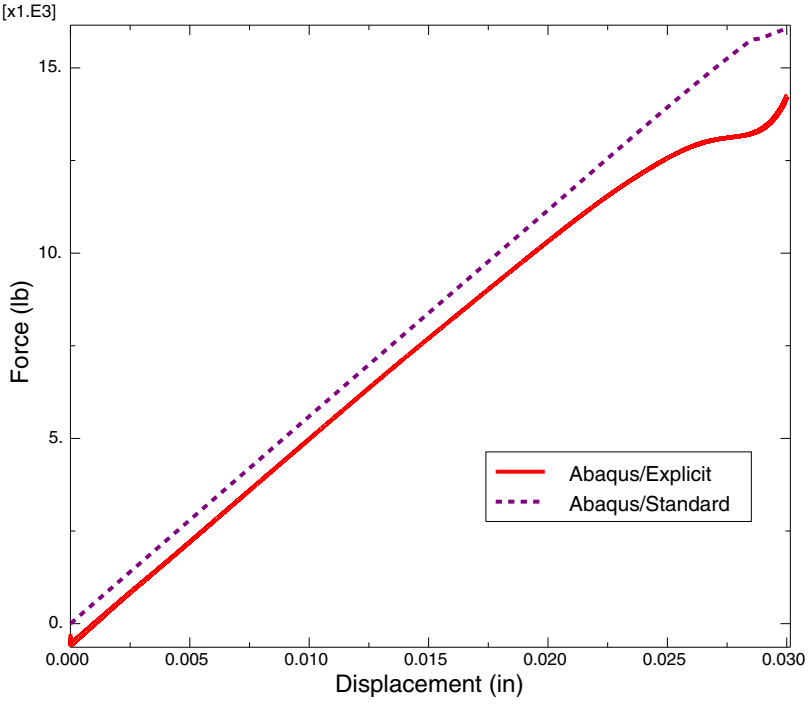


Figure 1.4.9–11 The force-displacement response comparison between Abaqus/Explicit and Abaqus/Standard.

1.5 Import analyses

- “Springback of two-dimensional draw bending,” Section 1.5.1
- “Deep drawing of a square box,” Section 1.5.2

1.5.1 SPRINGBACK OF TWO-DIMENSIONAL DRAW BENDING

Products: Abaqus/Standard Abaqus/Explicit

This example illustrates the forming and springback analysis of a two-dimensional draw bending process. The forming analysis is performed using Abaqus/Explicit, and the springback analysis is run with Abaqus/Standard using an import analysis.

Problem description

The example described here is one of the benchmark tests reported at the Numisheet '93 Conference. The benchmark contains a series of six problems performed with three different materials and two different blank holder forces. One of the six problems is described here. The simulations for all the problems are described in the paper by Taylor et al. (1993).

The blank initially measures 350 mm by 35 mm and is 0.78 mm thick. The problem is essentially a plane strain problem (the out-of-plane dimension for the blank is 35 mm). A cross-section of the geometry of the die, the punch, the blank holder, and the blank is shown in Figure 1.5.1–1. The total blank holder force is 2.45 kN, and a mass of 5 kg is attached to the blank holder. A coefficient of friction of 0.144 is used for all interacting surfaces.

The blank is made of mild steel. The material is modeled as an elastic-plastic material with isotropic elasticity, using the Hill anisotropic yield criterion for the plasticity. The following material properties are used:

Young's modulus = 206.0 GPa

Poisson's ratio = 0.3

Density = 7800.

Yield stress σ_0 = 167.0 MPa

Anisotropic yield criterion: R_{11}=1.0, R_{22}=1.0402, R_{33}=1.24897, R_{12}=1.07895, R_{13}=1.0, R_{23}=1.0

The problem is symmetric about a plane through the center of the punch, and only half of the problem is modeled. The blank is modeled with a single row of 175 first-order shell elements. Symmetry boundary conditions are applied on the plane of symmetry, and boundary conditions are applied on all the nodes of the blank to simulate the plane strain conditions. The out-of-plane dimension for the blank in the model is 5 mm; thus, the blank holder force is scaled appropriately.

The forming process is simulated in two steps with Abaqus/Explicit. The blank holder force is applied in the first step of the analysis. The force is ramped on with a smooth step definition to minimize inertia effects. In the second step of the analysis the punch is moved down 70 mm by prescribing the velocity of the rigid body reference node for the punch. The velocity is applied with a triangular smooth step amplitude function, starting and ending with zero velocity, and with a peak velocity occurring at the middle of the time period.

A significant amount of springback occurs in this case. Because the blank is very flexible and the fundamental mode of vibration is low, it would take a long simulation to obtain a quasi-static solution of the springback analysis in Abaqus/Explicit.

The springback analysis is performed with Abaqus/Standard using an import analysis. The results from the forming simulation in Abaqus/Explicit are imported into Abaqus/Standard, and a static analysis calculates the springback. During this step an artificial stress state that equilibrates the imported stress state is applied automatically by Abaqus/Standard and gradually removed during the step. The displacement obtained at the end of the step is the springback, and the stresses give the residual stress state.

Settings in the import analysis determine the reference configuration. When you choose to update the reference configuration in an import analysis, the deformed sheet with its material state at the end of the Abaqus/Explicit analysis is imported into Abaqus/Standard and the deformed configuration becomes the reference configuration. This procedure is most convenient if, during postprocessing, the displacements due to springback need to be displayed. When you do not update the reference configuration, the material state, displacements, and strains of the deformed sheet at the end of the Abaqus/Explicit analysis are imported into Abaqus/Standard, and the original configuration remains as the reference configuration. This procedure should be used if it is desirable to obtain a continuous displacement solution.

In this two-dimensional draw bending problem significant springback occurs, and large-displacement effects are included in the calculations by considering geometric nonlinearities in the step definition.

Further details of the import capability are discussed in "Transferring results between Abaqus/Explicit and Abaqus/Standard," Section 9.2.2 of the Abaqus Analysis User's Guide.

Results and discussion

The optimum peak velocity for the punch (the value that gives quasi-static results at least cost) is determined by running the explicit analysis with peak velocities of 30 m/s, 15 m/s, and 5 m/s. The energy histories are shown in Figure 1.5.1–2, Figure 1.5.1–3, and Figure 1.5.1–4, respectively. From these results it is evident that the amount of kinetic energy in the model is too large at a peak velocity of 30 m/s for the analysis to simulate the quasi-static forming process, while at a peak velocity of 5 m/s the kinetic energy is virtually zero. A peak velocity for the punch of 15 m/s is chosen for the forming analysis, as the kinetic energy for this case is considered low enough not to affect the results significantly. For accurate springback analysis it is important that stresses are not influenced by inertia effects.

The blank at the end of the Abaqus/Explicit forming analysis is shown in Figure 1.5.1–5. The shape after springback is shown in Figure 1.5.1–6. The results compare well with the reported experimental data. In the numerical results the angle between the outside flange and the horizontal axis is 22° in the Abaqus/Explicit analysis and 17.1° in the Abaqus/Standard analysis. The differences in the results are due to differences in the contact calculations. In Abaqus/Explicit the change in shell thickness is accounted for during contact calculations, and the surface-to-surface contact formulation in Abaqus/Standard does the same. However, for the node-to-surface contact formulation in Abaqus/Standard when a shell is pinched between two surfaces, it is necessary to use "softened"

contact to account for the shell thickness and, consequently, an approximation is made. A modified Abaqus/Explicit analysis that uses softened contact and zero shell thickness (with a zero-thickness surface) has been set up to compare the results from Abaqus/Explicit and Abaqus/Standard directly. The predicted results match closely. The average angle measured in the experiments is 17.1°, with a range from 9° to 23° in the experimental results. The results of the springback analysis when the reference configuration is updated are nearly identical to the results when the reference configuration is not updated.

Input files

springback_exp_form.inp	Forming analysis in Abaqus/Explicit with a punch velocity of 15 m/s using contact pairs.
springback_exp_form_gcont.inp	Forming analysis in Abaqus/Explicit with a punch velocity of 15 m/s using general contact.
springback_std_importyes.inp	Springback analysis in Abaqus/Standard with the *IMPORT, UPDATE=YES option.
springback_std_importno.inp	Springback analysis in Abaqus/Standard with the *IMPORT, UPDATE=NO option.
springback_std_both.inp	Input data used with Abaqus/Standard for both the forming and the springback analyses.
springback_std_both_surf.inp	Input data used with Abaqus/Standard for both the forming and the springback analyses using the surface-to-surface contact formulation.
springback_exp_form_soft.inp	Modified forming analysis in Abaqus/Explicit with a punch velocity of 15 m/s using softened contact and contact pairs as in springback_std_both.inp.
springback_exp_punchv30.inp	Forming analysis in Abaqus/Explicit with a punch velocity of 30 m/s using contact pairs.
springback_exp_punchv5.inp	Forming analysis in Abaqus/Explicit with a punch velocity of 5 m/s using contact pairs.
springback_exp_punchv30_gcont.inp	Forming analysis in Abaqus/Explicit with a punch velocity of 30 m/s using general contact.
springback_exp_punchv5_gcont.inp	Forming analysis in Abaqus/Explicit with a punch velocity of 5 m/s using general contact.

Reference

- Taylor, L. M., J. Cao, A. P. Karafillis, and M. C. Boyce, “Numerical Simulations of Sheet Metal Forming,” Proceedings of 2nd International Conference, NUMISHEET 93, Isehara, Japan, Ed. A. Makinovchi, et al.

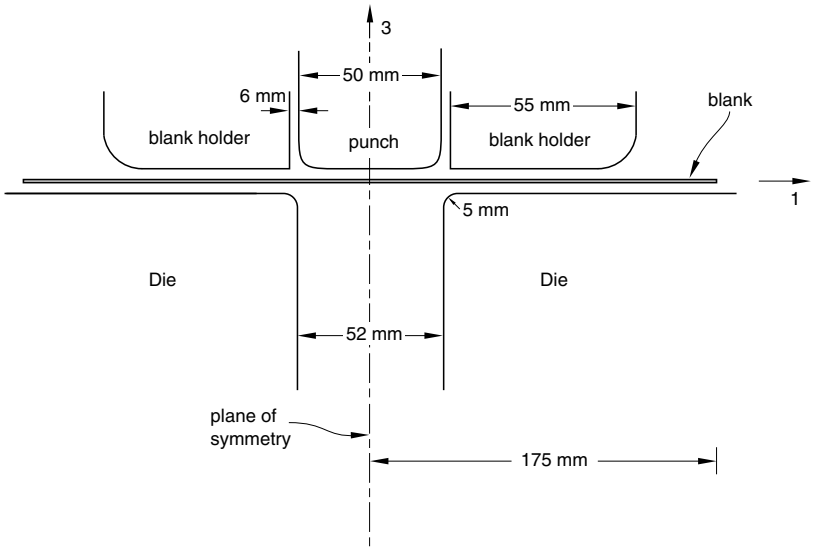


Figure 1.5.1–1 Cross-section showing the geometry of the die, the punch, the blank holder, and the blank.

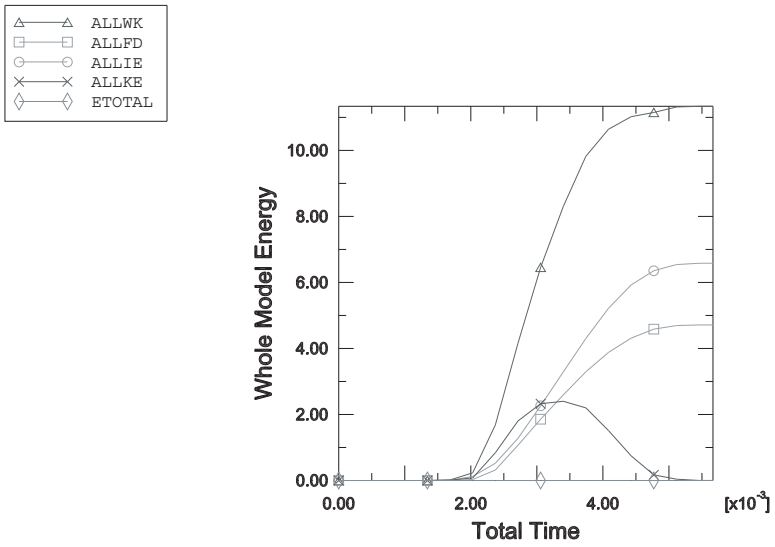


Figure 1.5.1–2 Energy history for forming analysis: 30 m/s peak velocity of punch.

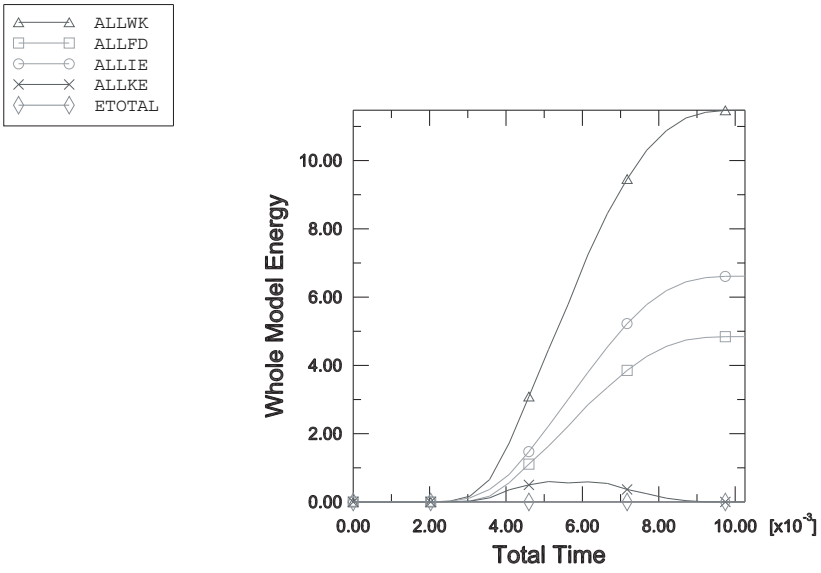


Figure 1.5.1–3 Energy history for forming analysis: 15 m/s peak velocity of punch.

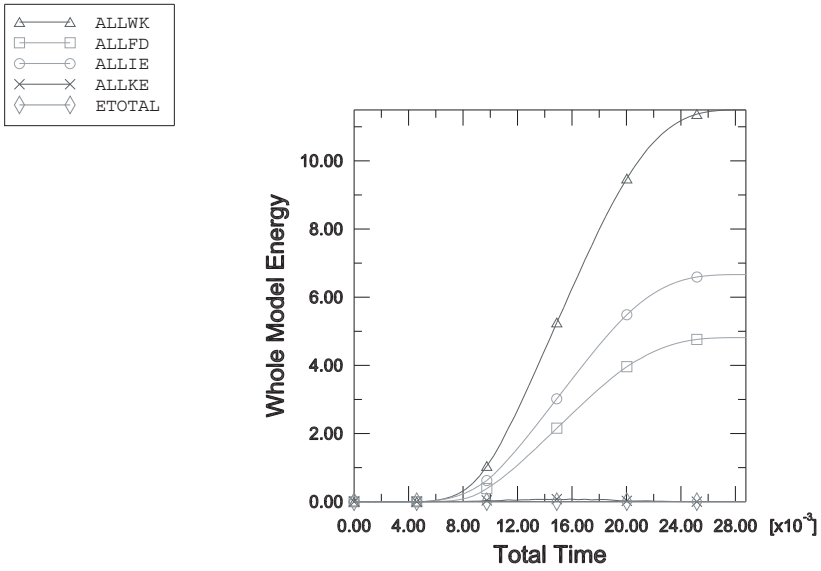


Figure 1.5.1–4 Energy history for forming analysis: 5 m/s peak velocity of punch.

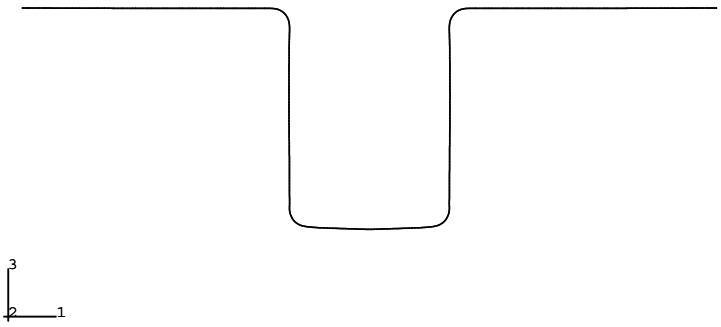


Figure 1.5.1–5 Blank at the end of the forming analysis in Abaqus/Explicit.

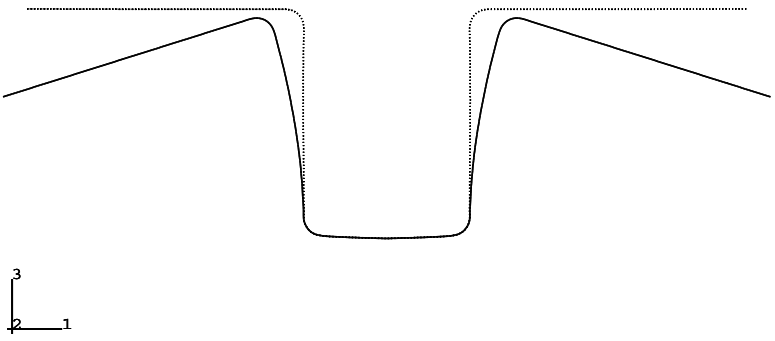


Figure 1.5.1–6 Blank after springback in Abaqus/Standard.

1.5.2 DEEP DRAWING OF A SQUARE BOX

Products: Abaqus/Standard Abaqus/Explicit

Objectives

This example problem demonstrates the following Abaqus features and techniques:

- transferring results from Abaqus/Explicit to Abaqus/Standard using the import analysis technique;
- comparing results from an analysis sequence that uses Abaqus/Explicit for a forming step and Abaqus/Standard for a springback analysis with results obtained using Abaqus/Standard for both the forming and springback steps; and
- comparing characteristics of different contact formulations with finite sliding, especially with regard to the treatment of surface thickness.

Application description

This example illustrates the forming of a three-dimensional shape by a deep drawing process. In general, the forming procedure involves a forming step followed by a springback that occurs after the blank is removed from the tool. The goal of analyzing the forming procedure is to determine the final deformed shape after springback.

Geometry

The blank is initially square, 200 mm by 200 mm, and is 0.82 mm thick. The rigid die is a flat surface with a square hole 102.5 mm by 102.5 mm, rounded at the edges with a radius of 10 mm. The rigid square punch measures 100 mm by 100 mm and is rounded at the edges with the same 10 mm radius. The rigid blank holder can be considered a flat plate, since the blank never comes close to its edges. The geometry of these rigid parts is illustrated in Figure 1.5.2–1.

Materials

The blank is made of aluminum-killed steel, which is assumed to satisfy the Ramberg-Osgood relation between true stress and logarithmic strain,

$$\epsilon = (\sigma/K)^{1/n},$$

with a reference stress value (K) of 513 MPa and a work-hardening exponent (n) of 0.223. Isotropic elasticity is assumed, with a Young's modulus of 211 GPa and a Poisson's ratio of 0.3. An initial yield stress of 91.3 MPa is obtained from these data. The stress-strain behavior is defined by piecewise linear segments matching the Ramberg-Osgood curve up to a total (logarithmic) strain level of 107%, with Mises yield, isotropic hardening, and no rate dependence.

Boundary conditions and loading

Given the symmetry of the problem, it is sufficient to model only a one-eighth sector of the box. However, for easier visualization we have employed a one-quarter model. Symmetry boundary conditions are applied at the quarter edges of the blank. The punch and the blank holders are allowed to move only in the vertical direction. Allowing vertical motion of the blank holders accommodates changes in the blank thickness during forming.

Interactions

Contact interaction is considered between the blank and the punch with a friction coefficient of 0.25 and between the blank and the die with a friction coefficient of 0.125. The contact interaction between the blank and the blank holders is assumed to be frictionless.

Abaqus modeling approaches and simulation techniques

The most efficient way to analyze this type of problem is to analyze the forming step using Abaqus/Explicit and to import the results in Abaqus/Standard to analyze the springback. For verification purposes the complete analysis is also carried out with Abaqus/Standard. However, this is computationally more expensive and will be prohibitively more expensive for simulation of the forming of realistic, complex components.

This problem is used in Nagtegaal and Taylor (1991) where implicit and explicit finite element techniques for forming problems are compared. The computer time involved in running the simulation using explicit time integration with a given mesh is directly proportional to the time period of the event, since the stable time increment size is a function of the mesh size (length) and the material stiffness. Thus, it is usually desirable to run the simulation at an artificially high speed compared to the physical process. If the speed in the simulation is increased too much, the solution does not correspond to the low-speed physical problem; i.e., inertial effects begin to dominate. In a typical forming process the punch may move at speeds on the order of 1 m/sec, which is extremely slow compared to typical wave speeds in the materials to be formed (the wave speed in steel is approximately 5000 m/sec). In general, inertia forces will not play a dominant role for forming rates that are considerably higher than the nominal 1 m/sec rates found in the physical problem. Therefore, explicit solutions are obtained with punch speeds of 10, 30, and 100 m/sec for comparison with the static solution obtained with Abaqus/Standard. In the results presented here, the drawing process is simulated by moving the reference node for the punch downward through a total distance of 36 mm in 0.0036 seconds. A detailed comparison of analyses of various metal forming problems using explicit dynamic and static procedures is discussed in the paper by Nagtegaal and Taylor (1991).

Although this example does not contain rate-dependent material properties, it is common in sheet metal forming applications for this to be a consideration. If the material is rate-dependent, the velocities cannot be artificially increased without affecting the material response. Instead, the analyst can use the technique of mass scaling to adjust the effective punch velocity without altering the material properties. “Rolling of thick plates,” Section 1.3.6, contains an explanation and an example of the mass scaling technique.

Summary of analysis cases

Forming analysis with Abaqus/Explicit.	Case 1a	Using the general contact capability.
	Case 1b	Using the kinematic contact pairs.
	Case 1c	Using penalty contact pairs.
	Case 1d	Forming analysis of a fine mesh case using the general contact capability (included for the sole purpose of testing the performance of the Abaqus/Explicit code).
	Case 1e	Forming analysis of a fine mesh case using kinematic contact pairs (included for the sole purpose of testing the performance of the Abaqus/Explicit code).
Springback analysis with Abaqus/Standard.	Case 2a	Abaqus/Standard springback analysis using an import analysis with no update of the reference configuration.
	Case 2b	Abaqus/Standard springback analysis using an import analysis with update of the reference configuration.
	Case 2c	Springback analysis of a fine mesh case (included for the sole purpose of testing the performance of the Abaqus/Standard code) using an import analysis with update of the reference configuration.
Forming and springback analysis with Abaqus/Standard.	Case 3a	Using the surface-to-surface contact formulation.
	Case 3b	Using the node-to-surface contact formulation.

Analysis types

As described earlier, the import capability in Abaqus is utilized to run the forming step as an explicit dynamic analysis followed by a static stress analysis using Abaqus/Standard for calculating the springback. For comparison, results from a complete static stress analysis using Abaqus/Standard for both the forming and the springback steps are presented.

Analysis techniques

The import feature in Abaqus is used for transferring results from Abaqus/Explicit to Abaqus/Standard.

Mesh design

The blank is modeled with 4-node, bilinear finite-strain elements (type S4R); while the punch, die, and the blank holder are meshed using 4-node, three-dimensional rigid surface elements (type R3D4). The mesh design for the various parts is shown in Figure 1.5.2–1 and Figure 1.5.2–2.

Loading

The blank is held between the blank holders by applying a concentrated load of 22.87 kN. Further loading on the blank is applied by contact forces with the punch in the forming step.

Analysis steps

Using Abaqus/Explicit for the forming procedure involves a single forming step where the rigid punch is pushed against the blank while the blank is held by the blank holders by applying a concentrated load. This description applies to Cases 1a–1e. For the import analysis in Abaqus/Standard a single step is used to calculate the springback as in Cases 2a–2c. For the complete analysis in Abaqus/Standard as in Cases 3a and 3b, the following steps are adopted:

- First step: the blank holders are brought in contact with the blank by applying a small displacement to the reference point of one of the rigid blank holders.
- Second step: a concentrated load is applied to the reference point of the blank holder to hold the blank in place while maintaining contact.
- Following steps: the forming is effected by pushing the rigid punch against the blank.
- Final two steps: the springback is analyzed by deactivating the contact pairs.

Output requests

The output variables STH for shell thickness and PEEQ for equivalent plastic strain are specifically requested along with preselected variables. Further, the history of reaction force and displacement for the punch is also requested.

Case 1a: Explicit forming analysis using general contact

This analysis pertains only to the forming step. For the complete analysis the forming step in this case needs to be followed by a springback analysis (either Case 2a or Case 2b).

Interactions

General contact is used (see the general contact specification) to define contact interactions in this case. This allows very simple definitions of contact with very few restrictions on the types of surfaces involved (see "Defining general contact interactions in Abaqus/Explicit," Section 36.4.1 of the Abaqus Analysis User's Guide). However, general contact does not account for changes in shell thickness by default. Consequently, the general contact surface property assignment must account for thinning of the blank.

Case 1b: Explicit forming analysis using kinematic contact pairs

This analysis again pertains only to the forming step. For the complete analysis the forming step needs to be followed by a springback analysis (either Case 2a or Case 2b).

Interactions

Contact pairs are defined to include blank interaction with the punch, die, and the blank holder separately with appropriate friction behavior as previously specified. The contact pair algorithm, which is specified in the contact pair definition, has more restrictions on the types of surfaces involved and often requires more careful definition of contact (see "Defining contact pairs in Abaqus/Explicit," Section 36.5.1 of the Abaqus Analysis User's Guide). Contact interactions are defined between all element-based surfaces in the model.

Case 1c: Explicit forming analysis using penalty contact pairs

This analysis pertains only to the forming step. The springback calculations have to be done separately (Case 2a or Case 2b).

Interactions

Penalty contact is specified for contact pairs to include blank interaction with the punch, die, and the blank holder separately with appropriate friction behavior.

Case 1d: Explicit forming analysis using general contact with a refined mesh

In this case the mesh for the blank is uniformly refined so that the number of elements in each direction is twice the number in the previous cases. This case is run to purely benchmark the efficiency of performing an explicit analysis.

Interactions

The contact interactions are exactly the same as in Case 1a.

Case 1e: Explicit forming analysis using kinematic contact pairs with a refined mesh

In this case the refined mesh defined in Case 1d is utilized for performing the explicit forming analysis.

Interactions

The contact interactions are exactly the same as in Case 1b.

Case 2a: Static springback analysis with no update of the reference configuration during import

For running this case, a prior explicit forming analysis (Case 1a, Case 1b, or Case 1c) should have been completed for importing results into Abaqus/Standard. By specifying an import analysis with no update of the reference configuration, the displacements are the total values relative to the original reference

configuration before the forming analysis. This makes it easy to compare the results with the analysis in which both the forming and springback are analyzed with Abaqus/Standard.

Boundary conditions

Boundary conditions are imposed in the Abaqus/Standard analysis to prevent rigid body motion and for symmetry. The node at the center of the box is fixed in the z-direction.

Interactions

No contact interactions are used in this analysis once the deformed sheet with its material state at the end of Abaqus/Explicit is imported.

Case 2b: Static springback analysis with update of the reference configuration during import

Similar to Case 2a, a prior explicit forming analysis (Case 1a, Case 1b, or Case 1c) should have been completed for importing results into Abaqus/Standard. However, specifying an import analysis with update of the reference configuration implies that the displacements are relative to the deformed configuration at the end of the forming analysis. The boundary conditions and interactions are exactly the same as Case 2a.

Case 2c: Static springback analysis using a refined mesh with update of the reference configuration during import

For running this case, Case 1d or Case 1e for explicit forming analysis should have been completed for importing results into Abaqus/Standard. Here again, specifying an import analysis with update of the reference configuration implies that the displacements are relative to the deformed configuration at the end of the forming analysis. The boundary conditions and interactions are exactly the same as Case 2a.

Case 3a: Static analysis of forming and springback using surface-to-surface contact

In this analysis both the forming and the springback steps are analyzed in Abaqus/Standard.

Interactions

In this case the surface-to-surface contact formulation is invoked. Since double-sided surfaces are not available in Abaqus/Standard, two single-sided surfaces are used to model the blank when the forming step is modeled in Abaqus/Standard: one surface to model the top of the blank and one to model the bottom of the blank. The surface-to-surface contact formulation considers the original shell thickness by default throughout the analysis. There is no option to consider the current shell thickness instead of the original shell thickness.

Solution controls

Contact stabilization is used to avoid chattering between the blank and the rigid surfaces it is in contact with. In addition, the adaptive automatic stabilization scheme is applied to improve the robustness of the static analysis.

Case 3b: Static analysis of forming and springback using node-to-surface contact

As in Case 3a, both the forming and the springback steps are analyzed in Abaqus/Standard.

Interactions

In this case the node-to-surface contact formulation is used. Since, shell thickness cannot be considered by node-to-surface finite-sliding contact, "softened" contact is used to approximate the thickness (see the modified contact pressure-overclosure relationship).

Discussion of results and comparison of cases

Figure 1.5.2–3, Figure 1.5.2–5, and Figure 1.5.2–4 show contours of shell thickness in the blank at the end of the forming step before springback in Abaqus/Explicit (Case 1a) and Abaqus/Standard analyses (Case 3a and Case 3b), respectively. Figure 1.5.2–6, Figure 1.5.2–7, and Figure 1.5.2–8 show contours of equivalent plastic strain in the blank in the final deformed shape for the Abaqus/Explicit and the two Abaqus/Standard analyses, respectively. The predicted results are very similar. The Abaqus/Explicit results match the surface-to-surface contact formulation in Abaqus/Standard more closely than the node-to-surface results in Abaqus/Standard. This observation is true for both the equivalent plastic strain contours and shell thickness contours and is a consequence of the intrinsic differences between the various contact formulations. The node-to-surface formulation in Abaqus/Standard accounts for the shell thickness indirectly by using carefully specified pressure-overclosure relationships (soft contact). The other analyses use contact formulations that account for shell thickness directly. Despite the fact that the surface-to-surface formulation in Abaqus/Standard uses the original shell thickness throughout the analysis, the results correlate well.

Closer inspection of the results reveals that the corners of the box are formed by stretching, whereas the sides are formed by drawing action. This effect leads to the formation of shear bands that run diagonally across the sides of the box, resulting in a nonhomogeneous wall thickness. The material draws unevenly from the originally straight sides of the blank. Applying a more localized restraint near the midedges of the box (for example, by applying drawbeads) and relaxing the restraint near the corners of the box is expected to increase the quality of the formed product.

Figure 1.5.2–9 shows the reaction force on the punch, and Figure 1.5.2–10 shows the thinning of an element at the corner of the box. Here again, the results from the surface-to-surface formulation in Abaqus/Standard match those from Abaqus/Explicit better than the node-to-surface contact formulation in Abaqus/Standard. In spite of the approximate treatment of surface thickness via the pressure-overclosure relationship for the node-to-surface formulation, the shell thicknesses predicted by Abaqus/Explicit and the node-to-surface formulation in Abaqus/Standard differ only by about 4%, reflecting the overall quality of the results.

The springback analysis runs in 6 increments for both of the contact formulations in Abaqus/Standard. Most of the springback occurs in the z-direction, and the springback is not significant. The corner of the outside edge of the formed box drops approximately 0.35 mm, while the vertical side of the box rises by approximately 0.26 mm. Figure 1.5.2–11 shows a contour plot of

the displacements in the z-direction obtained from the springback analysis using the node-to-surface formulation.

The analysis with no reference configuration update yields similar results. However, in this case the displacements are interpreted as total values relative to the original configuration.

Files

Case 1a: Explicit forming analysis using general contact

deepdrawbox_exp_form.inp	Input file for the explicit forming step.

Case 1b: Explicit forming analysis using kinematic contact pairs

deepdrawbox_exp_form_cpair.inp	Input file for the explicit forming step.

Case 1c: Explicit forming analysis using penalty contact pairs

deepdrawbox_exp_form_plty_cpair.inp	Input file for the explicit forming step.

Case 1d: Explicit forming analysis using general contact with a refined mesh

deepdrawbox_exp_finemesh.inp	Input file for the explicit forming step.

Case 1e: Explicit forming analysis using kinematic contact pairs with a refined mesh

deepdrawbox_exp_finemesh_cpair.inp	Input file for the explicit forming step.

Case 2a: Static springback analysis with UPDATE=NO during import

deepdrawbox_std_importno.inp	Input file for the static springback step.

Case 2b: Static springback analysis with UPDATE=YES during import

deepdrawbox_std_importyes.inp	Input file for the static springback step.

Case 2c: Static springback analysis using a refined mesh with UPDATE=YES during import

deepdrawbox_std_finesprngback.inp	Input file for the static springback step with a refined mesh for the blank.

Case 3a: Static analysis of forming and springback using surface-to-surface contact

deepdrawbox_std_both_surf.inp	Input file for the complete static analysis.
deepdrawbox_std_both_surf_stabil_adap.inp	Input file for the complete static analysis with adaptive stabilization.

Case 3b: Static analysis of forming and springback using node-to-surface contact

deepdrawbox_std_both.inp	Input file for the complete static analysis.

References

Abaqus Analysis User's Guide

- "Defining general contact interactions in Abaqus/Explicit," Section 36.4.1 of the Abaqus Analysis User's Guide
- "Defining contact pairs in Abaqus/Standard," Section 36.3.1 of the Abaqus Analysis User's Guide

Abaqus Keywords Reference Guide

- *CONTACT
- *CONTACT PAIR

Other

- Nagtegaal J. C. and L. M. Taylor, "Comparison of Implicit and Explicit Finite Element Methods for Analysis of Sheet Forming Problems," VDI Berichte No. 894, 1991.

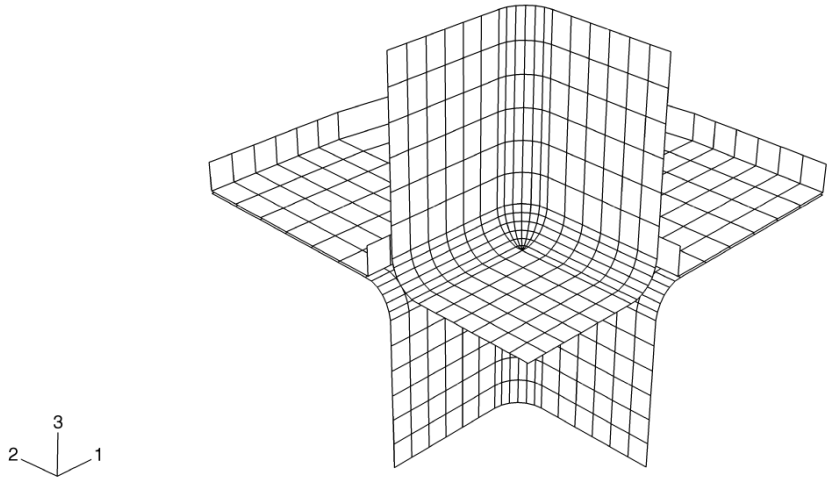


Figure 1.5.2–1 Meshes for the die, punch, and blank holder.

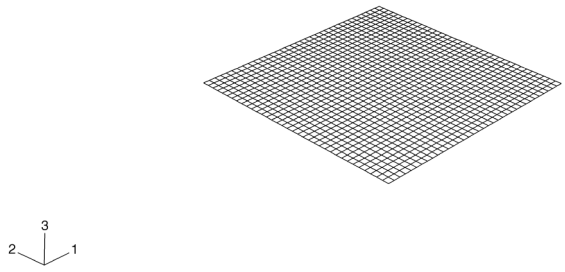


Figure 1.5.2–2 Undeformed mesh for the blank.

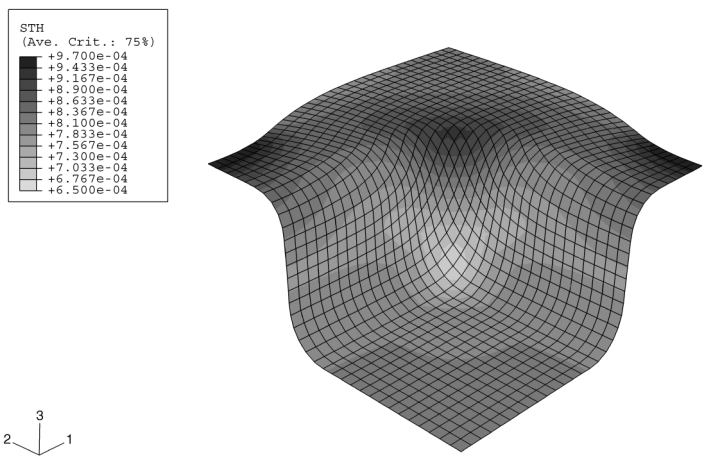


Figure 1.5.2–3 Contours of shell thickness with Abaqus/Explicit.

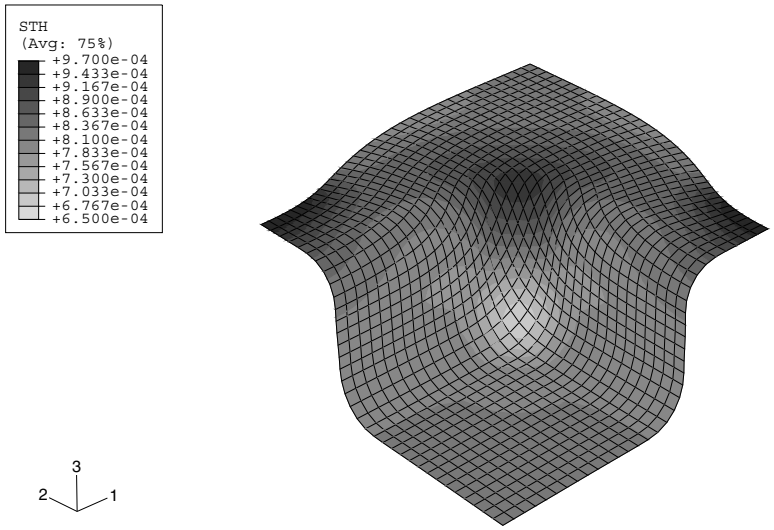


Figure 1.5.2–4 Contours of shell thickness with Abaqus/Standard using surface-to-surface contact formulation.

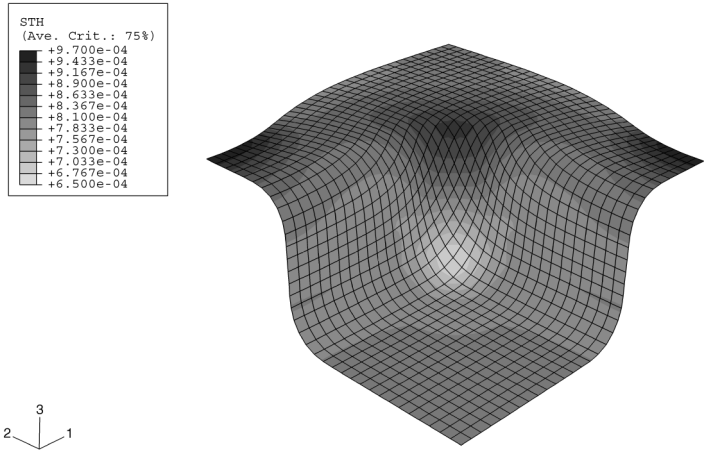


Figure 1.5.2–5 Contours of shell thickness with Abaqus/Standard using node-to-surface contact formulation.

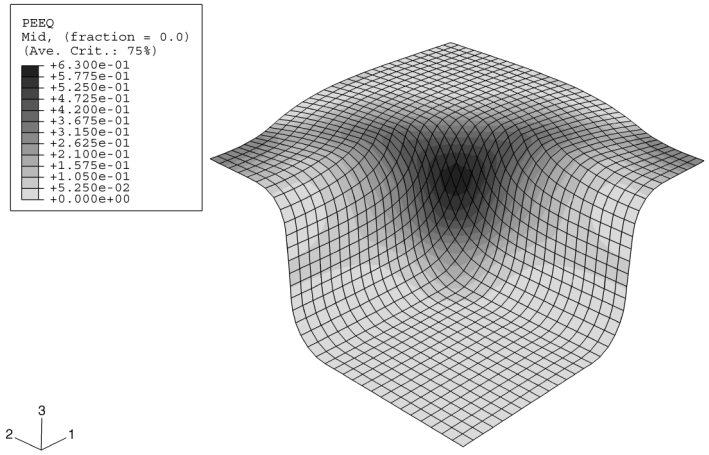


Figure 1.5.2–6 Contours of equivalent plastic strain with Abaqus/Explicit.

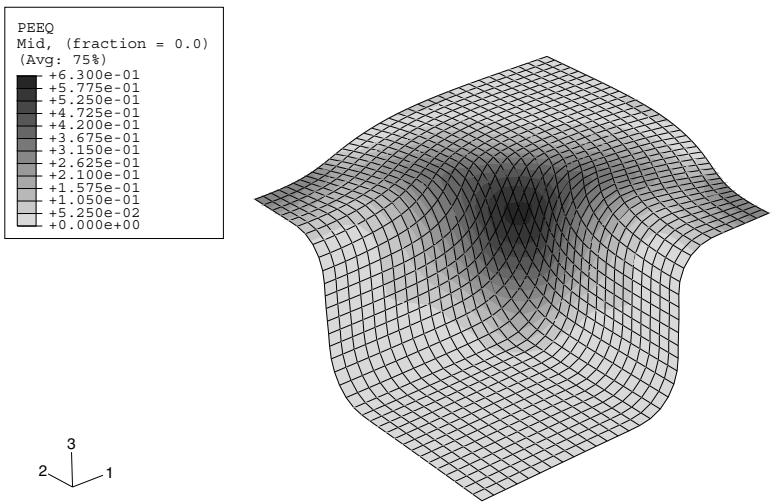


Figure 1.5.2–7 Contours of equivalent plastic strain with Abaqus/Standard using surface-to-surface contact formulation.

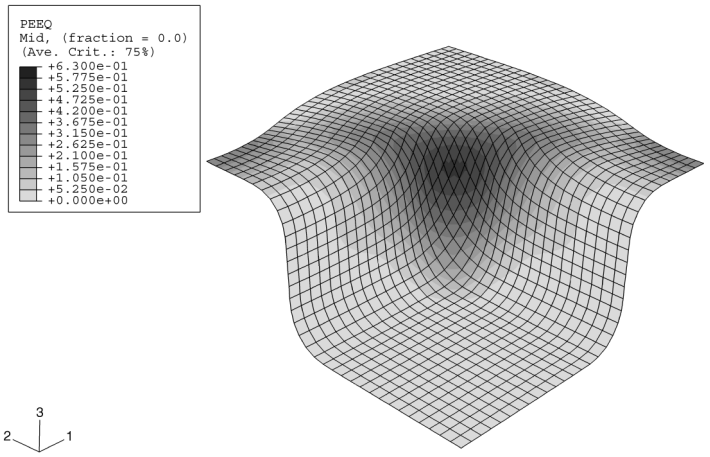


Figure 1.5.2–8 Contours of equivalent plastic strain with Abaqus/Standard using node-to-surface contact formulation.

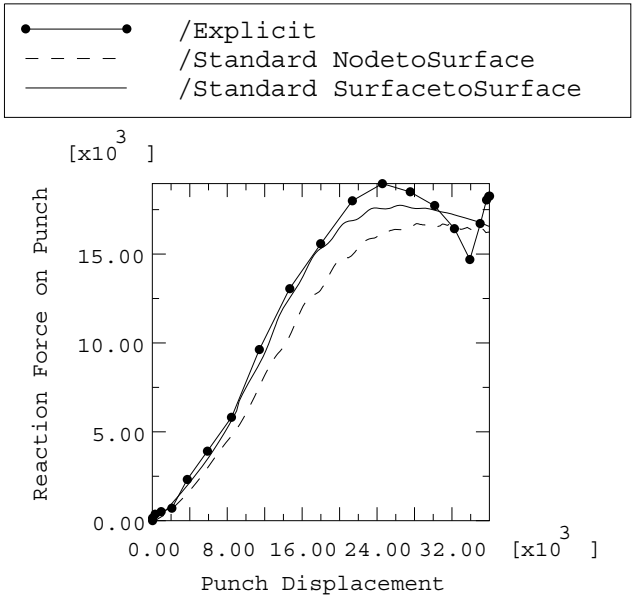


Figure 1.5.2–9 Reaction force on the punch versus punch displacement.

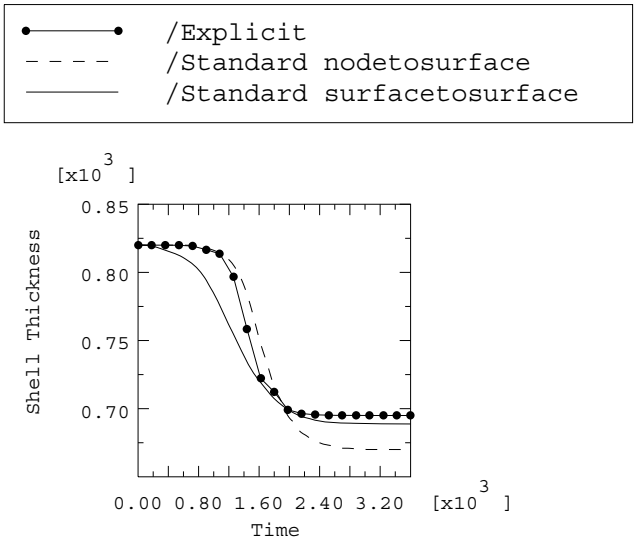


Figure 1.5.2–10 Shell thickness of the thinnest part of the blank versus time.

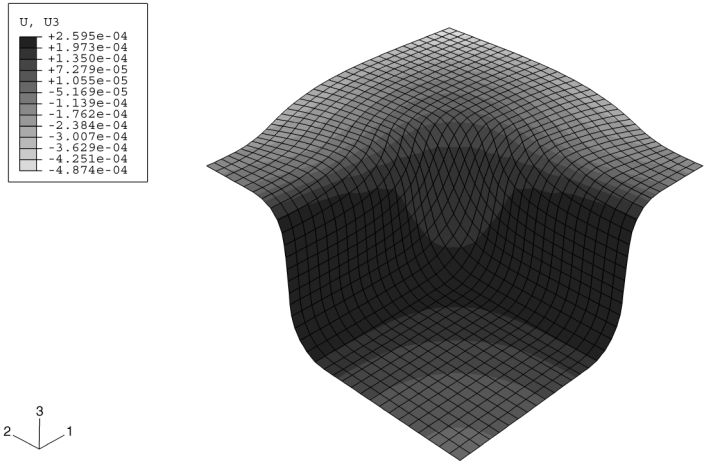


Figure 1.5.2–11 Contour plot showing the springback in the *z*-direction.

2. Dynamic Stress/Displacement Analyses

- "Dynamic stress analyses," Section 2.1
- "Mode-based dynamic analyses," Section 2.2
- "Eulerian analyses," Section 2.3
- "Co-simulation analyses," Section 2.4

2.1 Dynamic stress analyses

- "Nonlinear dynamic analysis of a structure with local inelastic collapse," Section 2.1.1
- "Detroit Edison pipe whip experiment," Section 2.1.2
- "Rigid projectile impacting eroding plate," Section 2.1.3
- "Eroding projectile impacting eroding plate," Section 2.1.4
- "Tennis racket and ball," Section 2.1.5
- "Pressurized fuel tank with variable shell thickness," Section 2.1.6
- "Modeling of an automobile suspension," Section 2.1.7
- "Explosive pipe closure," Section 2.1.8
- "Knee bolster impact with general contact," Section 2.1.9
- "Crimp forming with general contact," Section 2.1.10
- "Collapse of a stack of blocks with general contact," Section 2.1.11
- "Cask drop with foam impact limiter," Section 2.1.12
- "Oblique impact of a copper rod," Section 2.1.13
- "Water sloshing in a baffled tank," Section 2.1.14
- "Seismic analysis of a concrete gravity dam," Section 2.1.15
- "Progressive failure analysis of thin-wall aluminum extrusion under quasi-static and dynamic loads," Section 2.1.16
- "Impact analysis of a pawl-ratchet device," Section 2.1.17
- "High-velocity impact of a ceramic target," Section 2.1.18

2.1.1 NONLINEAR DYNAMIC ANALYSIS OF A STRUCTURE WITH LOCAL INELASTIC COLLAPSE

Product: Abaqus/Standard

This example illustrates an inexpensive approach to the prediction of the overall response of a structure that exhibits complex local behavior. The case studied is an unrestrained pipe whip example, where an initially straight pipe undergoes so much motion that the pipe section collapses. A two-stage technique is used to predict the response. First, the collapse of the section is studied under static conditions using a generalized plane strain model. This analysis defines the moment-curvature relationship for the section under conditions of pure bending. It also shows how the section deforms as it collapses. This information can be used to judge whether the deformation is reasonable with respect to possible failure (fracture) of the section. In addition, this first stage analysis can be used to calculate the change in the cross-sectional area enclosed by the pipe as a function of the curvature of the pipe. In a pipe whip case the driving force is caused by fluid jetting from a break in the pipe; and, if the pipe does undergo such large motion, a section may be deformed sufficiently to choke the flow. The second stage of the analysis is to predict the overall dynamic response of the pipe, using the moment-curvature response of the section that has been obtained in the first analysis to define the inelastic bending behavior of the beam. This two-stage approach provides a straightforward, inexpensive method of evaluating the event. The method is approximate and may give rise to significant errors. That aspect of the approach is discussed in the last section below.

Modeling

The problem is shown in Figure 2.1.1–1. To investigate the static collapse of the section, we consider a unit length of an initially straight pipe subjected to a pure bending moment and assume that plane sections remain plane. We can think of this unit length of pipe as being bounded at its ends by rigid walls and imagine the bending to be achieved by rotation of the walls relative to each other, the end sections being allowed to distort only in the plane of the walls (see Figure 2.1.1–2). With this idealization the pipe section can be modeled and discretized using generalized plane strain elements, as shown in Figure 2.1.1–2. Bending occurs about the x-axis, and symmetry conditions are prescribed along the y-axis. There will not be symmetry about the x-axis because of the Poisson's effect. To remove rigid body motion in the y-direction, point A is fixed in that direction. Symmetry implies no x-displacement at $x = 0$ and no rotation of the section about the y-axis. 4-node and 8-node generalized plane strain elements are used. In addition to the four or eight regular nodes used for interpolation, these elements require one extra reference node that is common to all elements in the model. Degree of freedom 3 at the reference node is the relative displacement between the boundary planes, while degrees of freedom 4 and 5 are the relative rotations of these planes.

Since the problem involves bending the pipe cross-section, regular fully integrated 4-node elements will not provide accurate results, especially when the pipe is fairly thin, because they will suffer from "shear locking"—they will not provide the bending deformation because to do so requires that they shear at their integration points and this shearing requires an unrealistically large amount of strain energy. This problem is avoided by integrating the elements only at their centroids but the elements then exhibit

singular modes—modes of deformation that do not cause strain. Abaqus uses orthogonal hourglass generalized strains and associated stiffness to avoid such spurious singular mode behavior. Although these techniques are not always reliable, they can work well and do so in this example. The problem is also modeled with the fully integrated incompatible mode element CPEG4I and the modified triangular element CPEG6M. CPEG4I elements do not have any hourglass modes and perform well in bending. For additional discussion of these points see "Performance of continuum and shell elements for linear analysis of bending problems," Section 2.3.5 of the Abaqus Benchmarks Guide.

For the dynamic analysis of the pipe whip event the pipeline is modeled with 10 beam elements of type B21. These are planar beam elements that use linear interpolation of displacement and rotation. The moment-curvature relation obtained from the static analysis (shown in Figure 2.1.1–5) is used in the nonlinear general cross-section to define the bending behavior of the beams. A definition for the axial force versus strain behavior of the beams is also required and is provided by conversion of the uniaxial stress-strain relation given in Figure 2.1.1–1 into force versus strain by multiplying the stress by the current area, A, of the cross-section. This current area is computed from the original cross-sectional area A_0 by assuming that the material is incompressible, so $A = A_0 l_0 / l$, where l is the current length and l_0 is the original length.

This definition of the beam section behavior provides for no interaction between the bending and axial stretching, although in most real cases there will actually be some interaction. However, this approximation is probably reasonable in this particular problem since the response is predominantly bending and no appreciable error is introduced by the little stretching that does occur.

Loading and solution control

In the large-displacement static analysis of the inelastic collapse of the section, rotation of the boundary planes about the x-axis is prescribed at degree of freedom 4 of the generalized plane strain reference node. The Riks procedure is used: this method usually provides rapid convergence in such cases, especially when unstable response occurs.

In the large-displacement dynamic analysis the blowdown force is treated as a follower force. During the first 0.06 seconds of the event it has a constant magnitude of 30 kN (this is about three times the load required to produce maximum moment in the static response of the section). After that time the load is zero. The response is computed for a time period of 0.4 seconds, using automatic time incrementation. A half-increment residual tolerance (HAFTOL) of 30 kN (which is the magnitude of the applied load) is used. Since we expect considerable plastic deformation, high frequency response should be damped quickly in the actual event, so that this value of HAFTOL should be adequate to give reasonably accurate results.

Results and discussion

Figure 2.1.1–3 shows a series of contour plots of equivalent plastic strain (plotted on the deformed configuration) from the static analysis using element type CPEG8R, and Figure 2.1.1–4 shows the same plots for the analysis using element type CPEG4R. Figure 2.1.1–3 clearly shows that the discretization is too coarse or should be rezoned later in the deformation, but it is judged that this is not critical to

the overall moment-rotation response prediction. Figure 2.1.1–5 shows the moment-curvature responses predicted by the analyses. The unstable nature of the response is clearly illustrated.

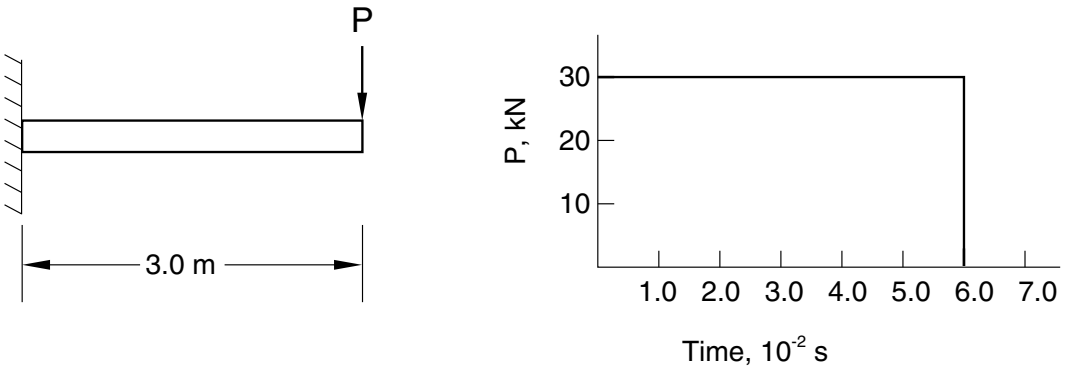
Figure 2.1.1–6 shows a series of deformed configuration plots from the dynamic analysis. After the shutdown of the force at 0.06 seconds, the momentum of the pipe is enough to cause localization of the deformation at the root of the cantilever as the section collapses there: the pipe whips around this hinge in a full circle and beyond its initial configuration. As well as this major hinge at the root, permanent plastic deformation develops throughout most of the pipe, leaving it bent into an arc. Time history plots of the tip displacement are shown in Figure 2.1.1–7 and of the curvature strain at the localized hinge in Figure 2.1.1–8. Figure 2.1.1–9 shows the moment-curvature response for the element at the support and shows the elastic unloading and reloading that takes place during and at the end of the event. Figure 2.1.1–10 shows the history of the energy content during the dynamic analysis and clearly shows the initial build-up of kinetic energy, which is then converted almost entirely to plastic dissipation.

This two-stage approach to the problem has the advantages of being simple and computationally inexpensive. It contains some obvious approximations. One is that interaction effects between bending, axial, and torsional behavior are neglected. This lack of interaction between the various modes of cross-sectional response is a basic approximation of the nonlinear beam general section option. In reality, axial or torsional strain will have the effect of reducing the strength of the section in bending. This effect is unlikely to be significant in a case that is dominated by bending, but it can be important if large axial or torsional loadings occur. The approach also neglects the effect of the axial gradient of the cross-sectional behavior on the response. This may be a significant error, but its evaluation would require a detailed, three-dimensional analysis for comparison; and that exercise is beyond the scope of this example. Another possibly significant error is the neglect of rate effects on the response. The cross-sectional collapse involves large strains, which occur in a very short time in the dynamic loadings, so high strain rates arise. It is likely that the material will exhibit strain rate dependence in its yield behavior and will, therefore, be rather stiffer than the static analysis predicts it to be. This should have the effect of spreading the hinge along the pipe and reducing the localization (because the strain rates increase at the section where most deformation is occurring, and that increased strain rate increases the resistance of the section). The magnitude of this effect can be estimated from the solution we have obtained. From Figure 2.1.1–3 we see that typical strains in the section are about 10–20% when the section is far into collapse; and Figure 2.1.1–8 shows that, in the dynamic event, it takes about 0.2 seconds for this to occur. This implies average gross strain rates of about 1.0 per second in that period of the response. In typical piping steels such a strain rate might raise the yield stress 5–10% above its static value. This is not a large effect, so the mitigation of localization by rate effects is probably not a major aspect of this event. Again, a more precise assessment of this error would require a fully three-dimensional analysis. Overall it seems likely that this simple and computationally inexpensive two-stage approach to the problem is providing results that are sufficiently realistic to be used in design, although it would be most desirable to compare these results with physical experimental data or data from a full, detailed, three-dimensional analysis to support that statement. Finally, it should be noted that the section considered here is relatively thick (R/t =3.5). In pipes with thin walls (R/t >20) it is to be expected that the behavior will be affected strongly by internal fluid pressure in the pipe and by the interaction between axial and bending forces. Such thin-walled pipes could be modeled at relatively low cost by using ELBOW elements directly in

the dynamic analysis instead of this two-stage approach. An additional concern with very thin pipes is that they are more likely to tear and leak, rather than choke the flow.

Input files

nonlindyncollapse_cpeg8r.inp	Static analysis of the elastic-plastic collapse of the pipe section using CPEG8R elements.
nonlindyncollapse_nonlingsect.inp	Dynamic analysis of the inelastic pipe whip response using nonlinear beam general section definitions for the axial and bending behaviors of the pipe.
nonlindyncollapse_cpeg4i.inp	Static analysis using element type CPEG4I.
nonlindyncollapse_cpeg4r.inp	Static analysis using element type CPEG4R.
nonlindyncollapse_cpeg4r_eh.inp	Static analysis using element type CPEG4R with enhanced hourglass control.
nonlindyncollapse_cpeg6m.inp	Static analysis using element type CPEG6M.
nonlindyncollapse_postoutput1.inp	*POST OUTPUT analysis.
nonlindyncollapse_postoutput2.inp	*POST OUTPUT analysis.



Geometry:

Initial outside radius = 38.05 mm
Initial thickness = 9.5 mm

Material:

Young's modulus = 208 GPa
Poissons ratio = 0.3

Yield behavior:

$\sigma = 895\ \varepsilon_p^{0.26}$ MPa for $\sigma > 316$ MPa
ε_p = plastic strain

Figure 2.1.1–1 Elastic-plastic pipe subjected to rupture force.

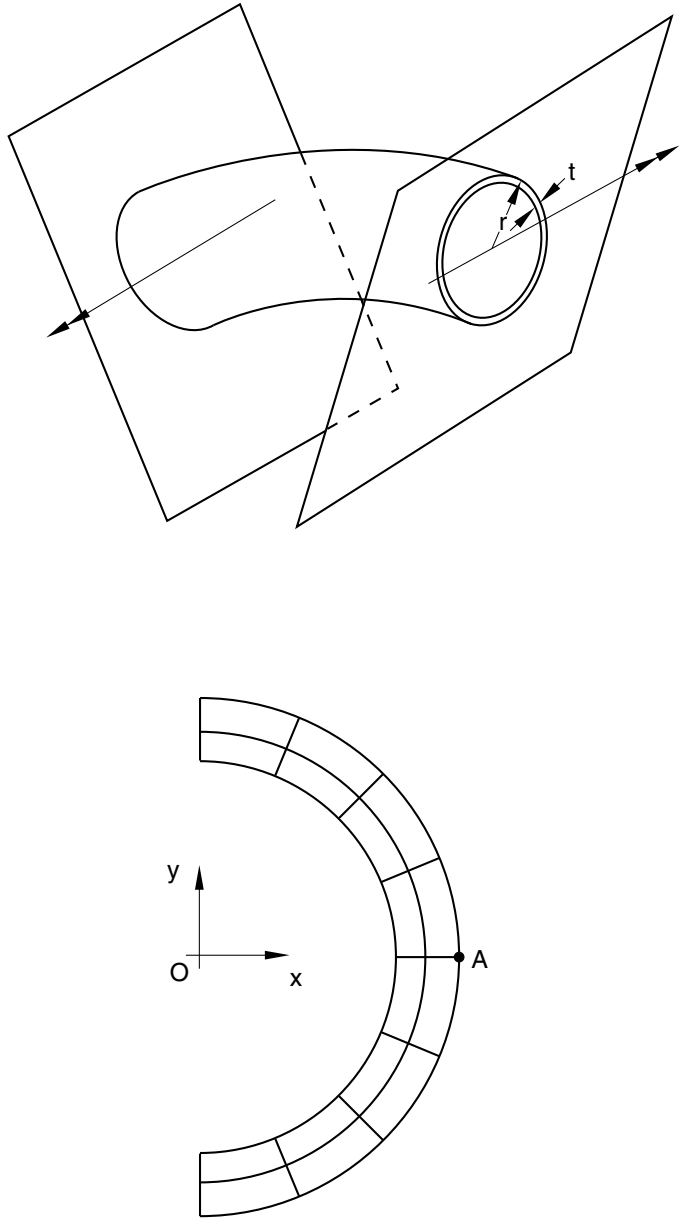


Figure 2.1.1–2 Initially straight pipe collapsing under pure bending; generalized plane strain model.

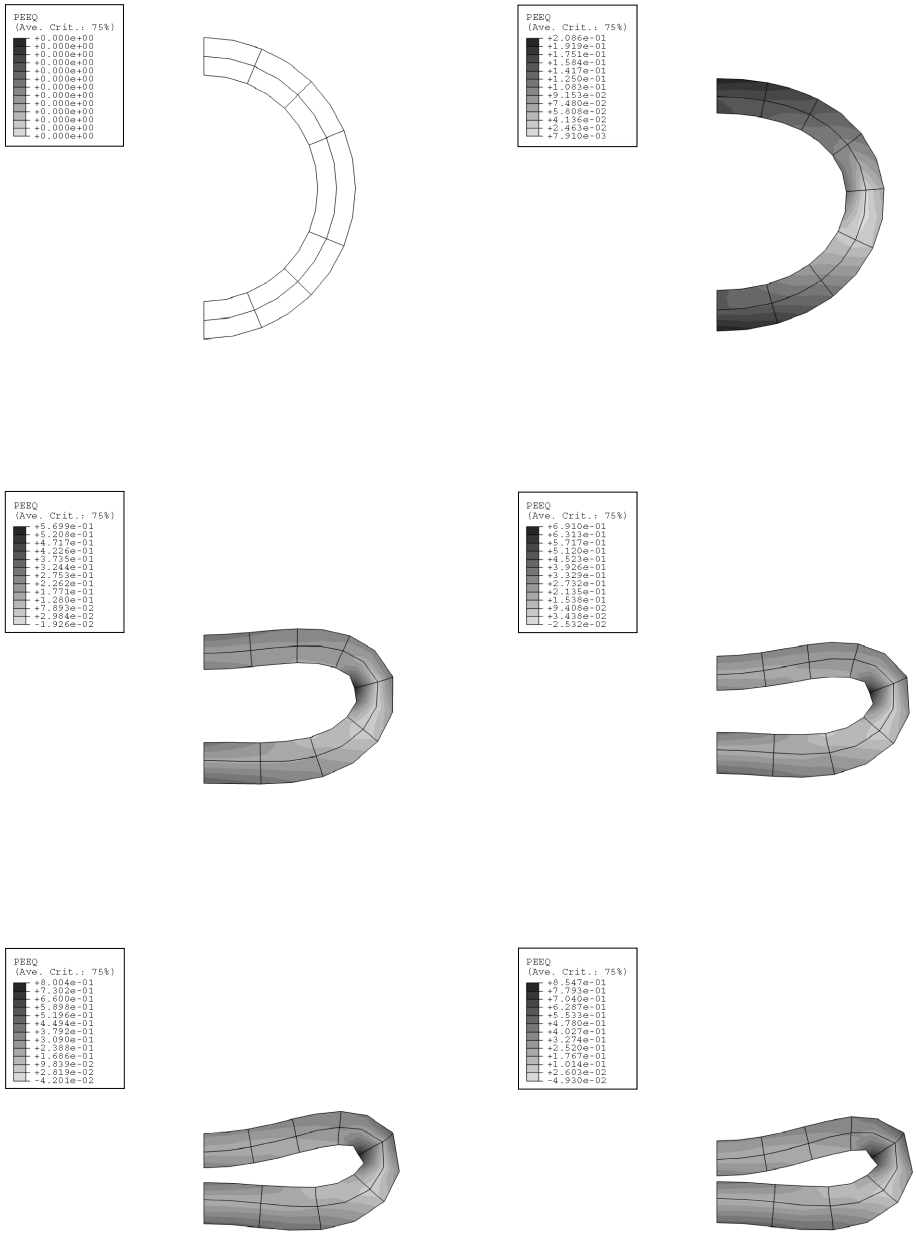


Figure 2.1.1–3 Equivalent plastic strain contours in collapsing pipe section, element type CPEG8R.

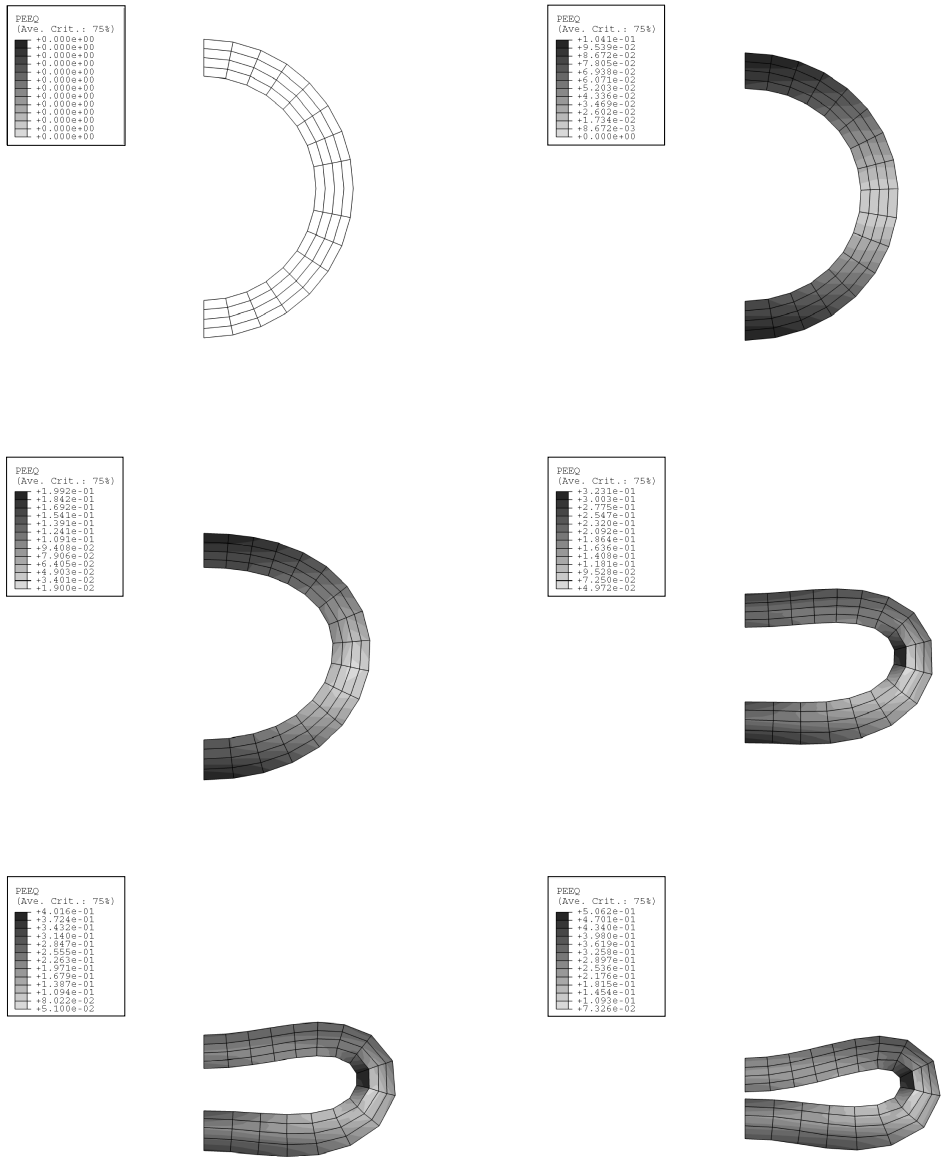


Figure 2.1.1–4 Equivalent plastic strain contours in collapsing pipe section, element type CPEG4R.

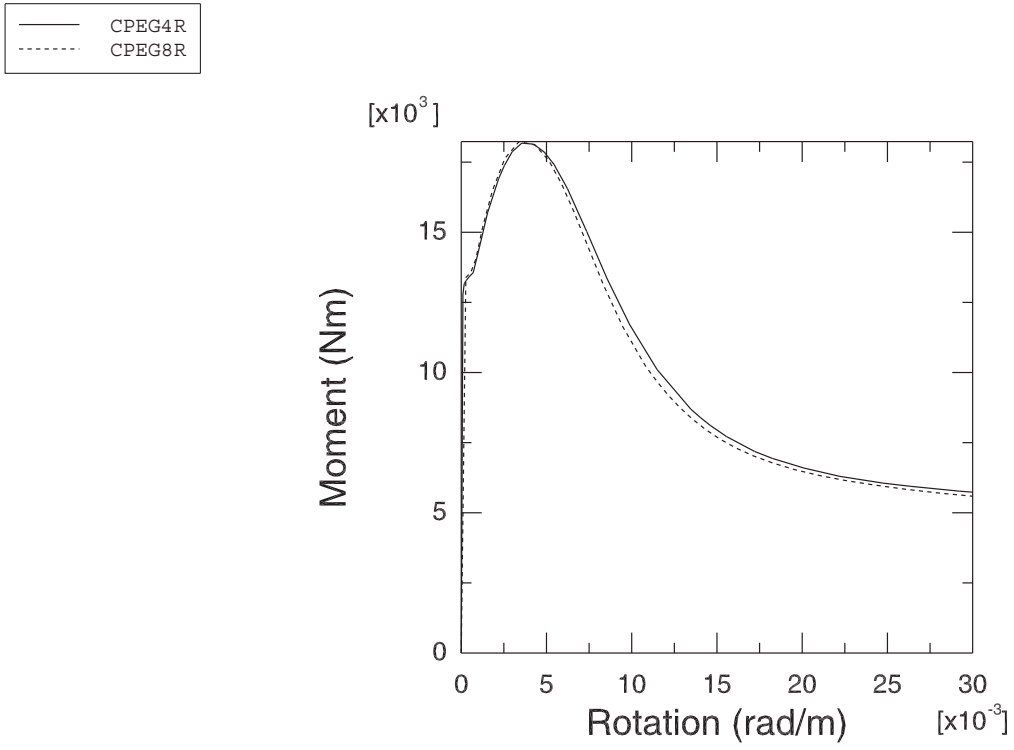


Figure 2.1.1–5 Moment-curvature response predicted for collapsing section under pure bending.



Figure 2.1.1–6 Displaced positions of pipe, every 20 increments. Initial increments in top figure, final increments in bottom figure.

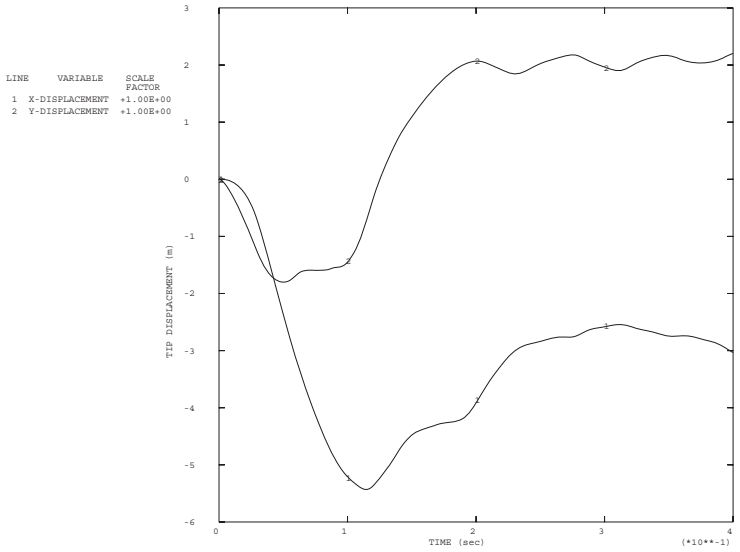


Figure 2.1.1–7 Tip displacement history.

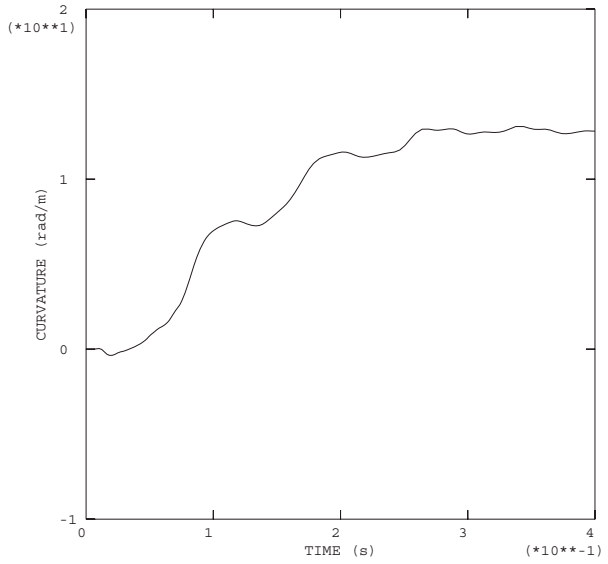


Figure 2.1.1–8 Curvature-time history for the element at the support.

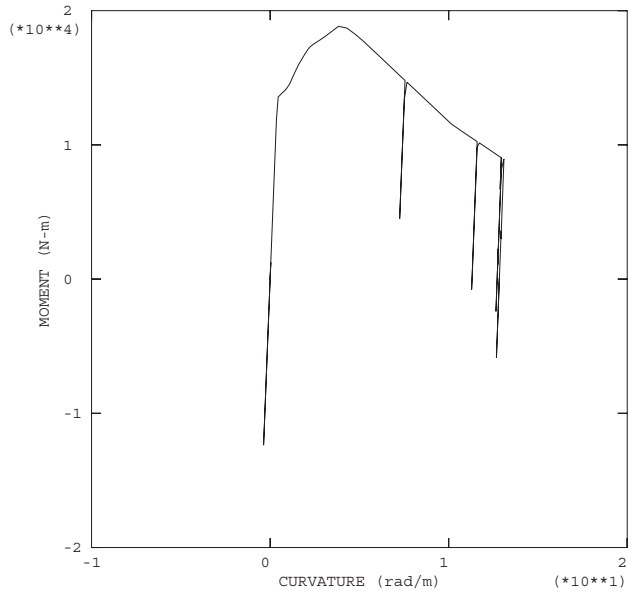


Figure 2.1.1–9 Moment versus curvature in the element at the support.

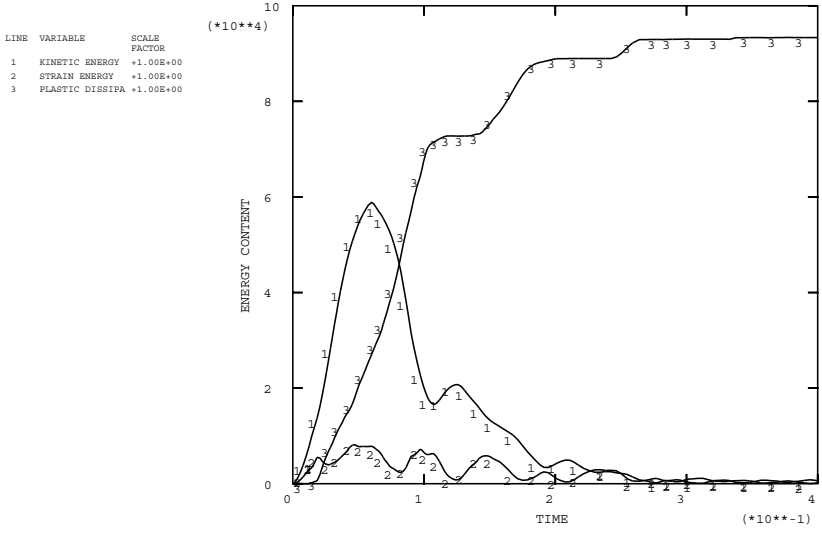


Figure 2.1.1–10 Energy history for the beam.

2.1.2 DETROIT EDISON PIPE WHIP EXPERIMENT

Product: Abaqus/Standard

This example is a model of a simple, small-displacement pipe whip experiment conducted by the Detroit Edison Company and reported by Esswein et al. (1978). The problem involves rather small displacements but provides an interesting case because some (limited) experimental results are available. It is a typical pipe whip restraint design case. It is a rather straightforward analysis because the restraint limits the motion and the geometry is so simple.

Geometry and model

The geometry and loading are shown in Figure 2.1.2–1. The pipe has a straight run of length 2.286 m (90 in), a very stiff elbow, and a cantilever "stick" 482.6 mm (19 in) long. A bursting diaphragm is installed at the end of the "stick" to initiate the blowdown. The restraint is a set of three U-bolts coupled together. The blowdown force history measured in the experiment is also shown in Figure 2.1.2–1. All dimensions, material properties, and this force history are taken from Esswein et al. (1978).

The horizontal pipe run is modeled with eight elements of type B23 (cubic interpolation beam with planar motion), and the stick is modeled with two elements of the same type. The elbow is treated as a fully rigid junction, so the node at the elbow is shared between the two branches. The bursting diaphragm structure is modeled as a lumped mass of 106.8 kg (0.61 lb s^2/in). The restraint is modeled as a single truss element. For the pipe the Young's modulus is 207 GPa (30×10^6 lb/in^2), the initial yield stress is 214 MPa (31020 lb/in^2), and the work hardening modulus is $846\varepsilon^{0.2}$ MPa ($122700\varepsilon^{0.2}$ lb/in^2) after yield. The restraint has an elastic stiffness of 131.35 MN/m (750000 lb/in), a yield force of 16681 N (3750 lb), and—when yielding—a force-displacement response $F = 2.2716\,\delta^{0.235}$ MN/m ($12971\delta^{0.235}$ lb/in). These values are taken from Esswein et al. (1978), where it is stated that they are based on measurements of static values with the stresses and forces increased by 50% in the plastic range to account for strain-rate effects. When it is known that strain-rate effects are important to the response it is preferable to model them directly, using a rate-dependent viscoplastic model. This has not been done in this case because the actual material is not specified.

Isotropic hardening is assumed for both the pipe and the restraint since the plastic flows are presumed to be in the large flow regime and not just incipient plasticity (where the Bauschinger effect can be important). The cross-section of the pipe is integrated with a seven-point Simpson rule: this should be of sufficient accuracy for this problem. Generally, in beam-like problems without repeated large magnitude excitation, a higher-order integration scheme would show only significantly different results at late times in the response, and then the differences are not too important in models of this rather unrefined level.

Esswein et al. (1978) provide the blowdown force-time history shown in Figure 2.1.2–1. This is applied as a point load at the end of the stick. In reality the fluid force during blowdown occurs at the piping elbows; but, since the displacements remain small, this detail is not important.

Solution control

Automatic time stepping is used, with an initial time increment of 100 μsec and the value of the half-increment residual tolerance, set to 4448 N (1000 lb). This value is based on actual force values expected (in this case, the blowdown force): HAFTOL is chosen to be about 10% of peak real forces. This should give good accuracy in the dynamic integration.

Results and discussion

The displacement of the node that hits the restraint is shown in Figure 2.1.2–2, and the force between the pipe and restraint is shown in Figure 2.1.2–3. Some experimental results from Esswein et al. (1978) are shown in Figure 2.1.2–3.

The analysis appears to predict the closure time and the peak force between the pipe and restraint quite well. However, the numerical solution (like the numerical solution given by Esswein et al., 1978) shows a slower force rise time than the experiment. A possible explanation may be the material model, where viscoplastic (strain-rate-dependent yield) effects have been modeled as enhanced yield values, as discussed above: this means that, at the high strain rate that occurs just after impact, the actual material can carry higher stresses than the model, and so will respond more stiffly. The oscillation in the gap force in Figure 2.1.2–3 after the initial loading of the restraint is presumably caused by the difference in the basic natural frequencies of the restraint and the pipe: this oscillation is sufficiently severe to cause two slight separations.

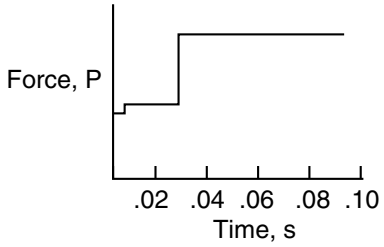
Input files

detroitedison.inp	Input data for this analysis.
detroitedison_postoutput.inp	*POST OUTPUT analysis.

Reference

- Esswein, G., S. Levy, M. Triplet, G. Chan, and N. Varadavajan, *Pipe Whip Dynamics,* ASME Special Publication, 1978.

P		time
N	lb	s
32027	7200	0- 0.00045
32348	7272	0.00045 -0.0291
35230	7920	0.0291-



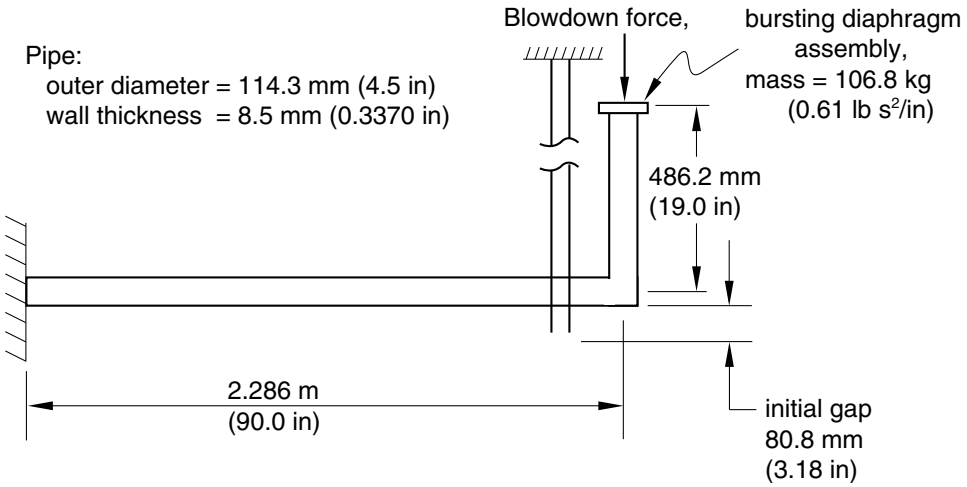


Figure 2.1.2–1 Detroit Edison experiment.

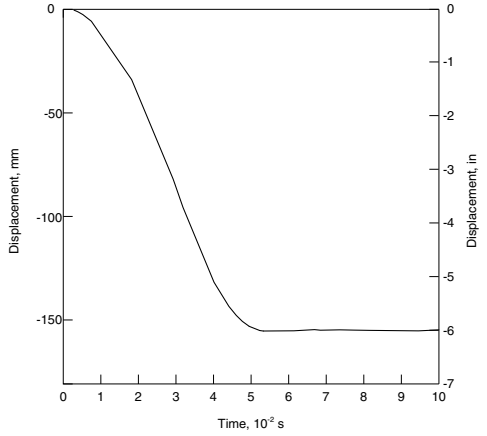


Figure 2.1.2–2 Displacement history at constrained end.

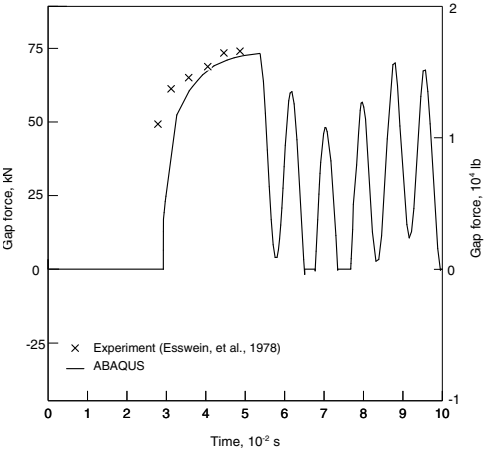


Figure 2.1.2–3 Gap force history.

2.1.3 RIGID PROJECTILE IMPACTING ERODING PLATE

Product: Abaqus/Explicit

This example simulates the oblique impact of a rigid spherical projectile onto a flat armor plate at a velocity of 1000 m/sec. A failure model is used for the plate, thus allowing the projectile to perforate the plate. The example illustrates impact, progressive failure, and the use of infinite elements.

Problem description

The armor plate has a thickness of 10 mm and is assumed to be semi-infinite in size compared to the projectile. This is accomplished by using CIN3D8 infinite elements around the perimeter of the plate. The plate is modeled using 4480 C3D8R elements. The armor plate material has Young's modulus of 206.8 GPa, Poisson's ratio of 0.3, density of 7800 kg/m^3, yield stress of 1220 MPa, and a constant hardening slope of 1220 MPa. The material definition also includes a progressive failure model, which causes Abaqus/Explicit to remove elements from the mesh as they fail. Failure is assumed to occur at an equivalent plastic strain of 100%, at which point the element is removed from the model instantaneously. (The value of the failure strain is chosen somewhat arbitrarily; it is not intended to model any particular material.)

The sphere has a diameter of 20 mm and is assumed to be rigid, with a mass corresponding to a uniform material with a density of 37240 kg/m^3. The rotary inertia of the sphere is not needed in the model because we assume there is no friction between the sphere and the plate. Boundary conditions are applied to constrain the motion of the sphere in the y-direction. Two approaches for modeling the surface of the sphere are tested: using an analytical rigid surface and using R3D4 rigid elements. Analytical rigid surfaces are the preferred means for representing simple rigid geometries such as this in terms of both accuracy and computational performance. However, more complex three-dimensional surface geometries that occur in practice must be modeled with surfaces formed by element faces. Results for the faceted representations are presented here. The element formulation for the C3D8R elements is modified with the section controls. The advocated formulation for this problem uses the centroid kinematic formulation and a linear combination of stiffness and viscous hourglass control. Additional combinations of kinematic formulation and hourglass control are included for comparison.

Only half of the plate is modeled, using appropriate symmetry boundary conditions in the x–z plane. The model is shown in Figure 2.1.3–1. The complete sphere is modeled for visualization purposes. There are 17094 degrees of freedom in the model.

Since elements in the plate will fail and be removed from the model, nodes in the interior of the plate will be exposed to contact with the surface of the rigid sphere. Thus, contact must be modeled between the surface of the sphere, defined as an element-based surface, and a node-based surface that contains all of the nodes in the plate within a radius of 20 mm of the point of impact. (See "Eroding projectile impacting eroding plate," Section 2.1.4, for an example in which element-based surfaces are used to model erosion.) In the primary input files contact pairs are used to define contact between the surface of the sphere and any of the nodes contained in the node set. Input files that use the general contact algorithm are also provided.

Results and discussion

The spherical projectile impacts the plate at 1000 m/sec at an angle of 30° to the normal to the plate. Deformed shapes at different stages of the analysis are shown in Figure 2.1.3–2 through Figure 2.1.3–4 for the centroid kinematic and the combined (viscous-stiffness form) hourglass section controls (analysis case pl3d_erode_ccs). Early in the analysis, shown in Figure 2.1.3–2, a relatively small amount of material has been eroded from the surface of the plate and the plate is still deforming under the sphere. In Figure 2.1.3–3 the plate has been perforated and the projectile is still in contact with the edge of the hole. In Figure 2.1.3–4 the projectile has exited the plate and is moving away with a constant velocity. Figure 2.1.3–5 and Figure 2.1.3–6 show the history of the projectile's velocity (Table 2.1.3–1 shows the analysis options used to obtain these results). The results show close agreement.

In Figure 2.1.3–2 through Figure 2.1.3–4 the failed elements have been eliminated by creating a display group in Abaqus/CAE that contains only the active elements.

Input files

pl3d_erode_ccs.inp	Model using the CENTROID kinematic and COMBINED hourglass section control options.
pl3d_erode_ccs_gcont.inp	Model using the CENTROID kinematic and COMBINED hourglass section control options and the general contact capability.
pl3d_erode_ces.inp	Model using the CENTROID kinematic and ENHANCED hourglass section control options.
pl3d_erode_ces_gcont.inp	Model using the CENTROID kinematic and ENHANCED hourglass section control options and the general contact capability.
sphere_n.inp	External file referenced in this input.
sphere_e.inp	External file referenced in this input.
pl3d_erode.inp	Model using the default section controls.
pl3d_erode_gcont.inp	Model using the default section controls and the general contact capability.
pl3d_erode_ale.inp	Model using the default section controls and the *ADAPTIVE MESH option.
pl3d_erode_ocs.inp	Model using the ORTHOGONAL kinematic and COMBINED hourglass section control options.
pl3d_erode_ocs_gcont.inp	Model using the ORTHOGONAL kinematic and COMBINED hourglass section control options and the general contact capability.
pl3d_erode_oes.inp	Model using the ORTHOGONAL kinematic and ENHANCED hourglass section control options.

pl3d_erode_oes_gcont.inp — Model using the ORTHOGONAL kinematic and ENHANCED hourglass section control options and the general contact capability.

pl3d_erode_anl.inp — Model using an analytical rigid surface and the default section controls.

Table 2.1.3–1 Analysis section controls tested.

Analysis File	Relative CPU Time	Section Controls	
		Kinematic Formulation	Hourglass Control
pl3d_erode	1.0	average strain	integral viscoelastic
pl3d_erode_ocs	0.86	orthogonal	combined
pl3d_erode_oes	0.88	orthogonal	enhanced
pl3d_erode_ccs	0.73	centroid	combined
pl3d_erode_ces	0.75	centroid	enhanced

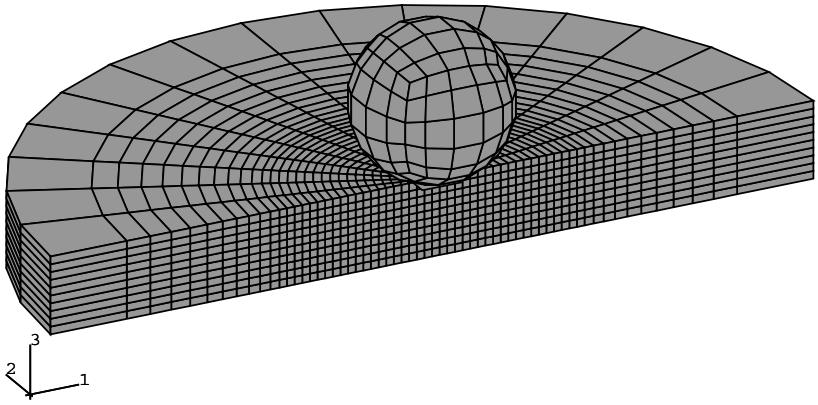


Figure 2.1.3–1 Undeformed mesh.

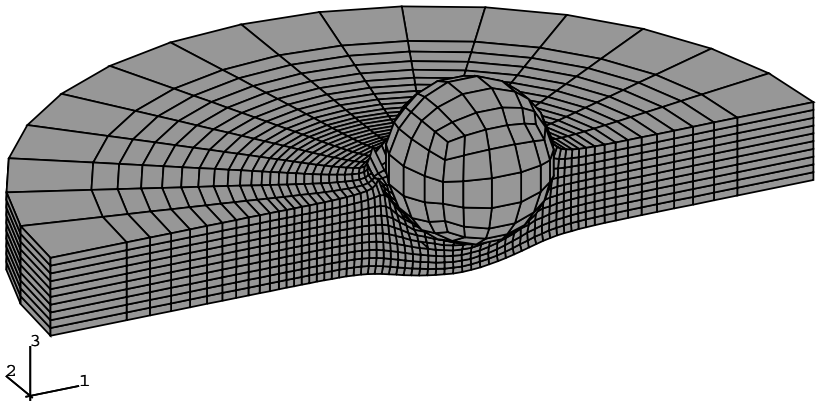


Figure 2.1.3–2 Deformed shape at 10 microseconds (analysis using the centroid kinematic and combined hourglass section controls).

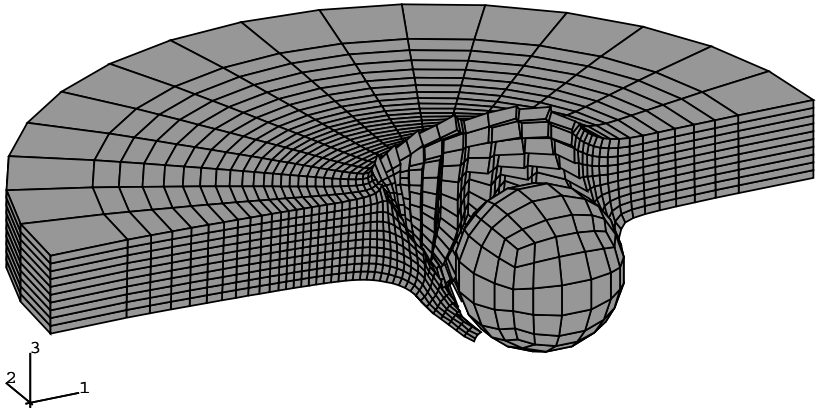


Figure 2.1.3–3 Deformed shape at 30 microseconds (analysis using the centroid kinematic and combined hourglass section controls).

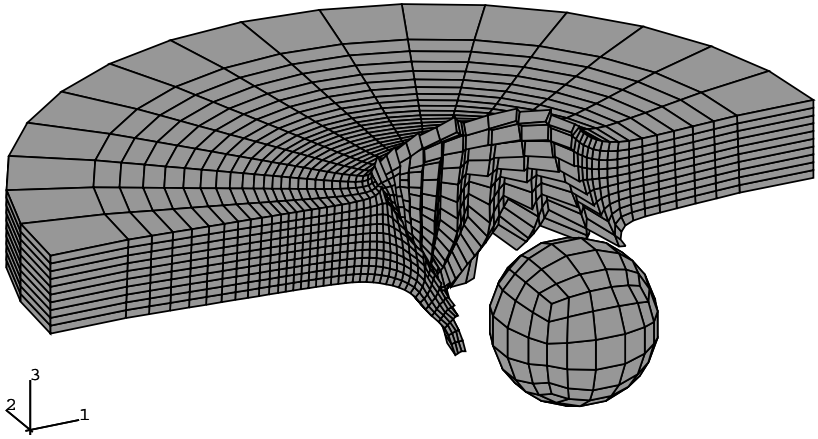


Figure 2.1.3–4 Deformed shape at 40 microseconds (analysis using the centroid kinematic and combined hourglass section controls).

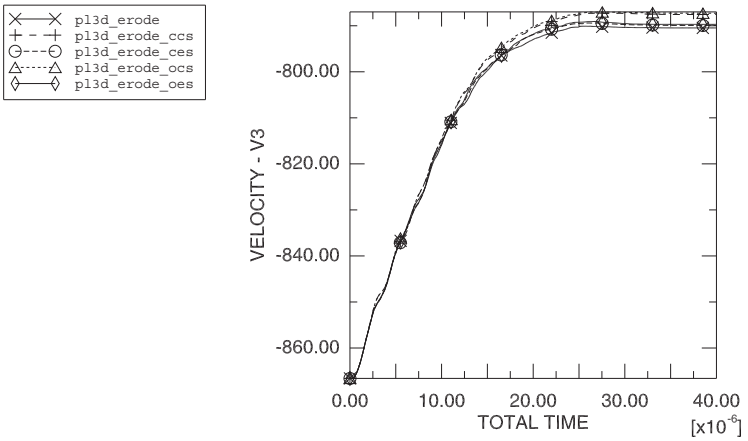


Figure 2.1.3–5 Vertical component of the projectile velocity.

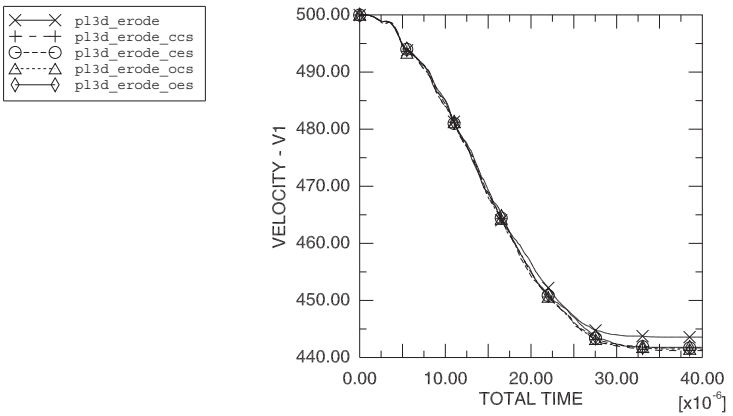


Figure 2.1.3–6 Horizontal component of the projectile velocity.

2.1.4 ERODING PROJECTILE IMPACTING ERODING PLATE

Product: Abaqus/Explicit

This example simulates the oblique impact of a cylindrical projectile onto a flat armor plate at a velocity of 2000 m/sec. The same material model, which includes a failure model with progressive damage, is used for both the projectile and plate. This example demonstrates the ability of the general contact algorithm to model surface erosion on multiple contacting bodies during high-speed impact.

Problem description

The undeformed mesh is shown in Figure 2.1.4–1. The armor plate has a thickness of 3 mm. A relatively small rectangular region of the plate is modeled for simplicity, with fully fixed boundary conditions specified on three cutting planes and y-axis symmetry specified on one cutting plane. The projectile, which is 10 mm in length and has a radius of 1 mm, has an initial speed of 2000 m/sec. The cylindrical axis of the projectile is 20° from perpendicular to the plate, and the initial velocity of the projectile is aligned with its cylindrical axis. Half of the projectile is modeled, with y-axis symmetry specified on the cutting plane. The plate and projectile material properties are identical, with Young's modulus of 210 GPa, Poisson's ratio of 0.3, and density of 7800 kg/m^3. The yield stress of the material is specified as a function of the equivalent plastic strain at different equivalent plastic strain rates. The material definition also includes failure models with progressive damage, which causes Abaqus/Explicit to remove elements from the mesh as they fail. Both the ductile and shear initiation criteria are used: the ductile criterion is specified in terms of the plastic strain at the onset of damage as a tabular function of the stress triaxiality; the shear criterion is specified in terms of the plastic strain at the onset of damage as a tabular function of the shear stress ratio. The damage evolution energy is assumed to be 500 N/m.

During the analysis elements from both bodies fail, which calls for the use of element-based surfaces that can adapt to the exposed surfaces of the current non-failed elements. The general contact algorithm supports element-based surfaces that evolve in this manner (whereas the contact pair algorithm does not). To model eroding contact, the user must include in the contact domain all surface faces that may become exposed during the analysis, including faces that are originally in the interior of bodies. Only the interior faces that are expected to participate in contact are included in the contact domain in this analysis to minimize the memory use (including interior faces for all elements in the model would more than double the memory use).

By default, the general contact algorithm does not include nodal erosion, so contact nodes will still take part in the contact calculations even after all of the surrounding elements have failed. These nodes act as free-floating point masses that can experience contact with the active contact faces. For comparison purposes, the analysis is also conducted including nodal erosion, which causes the nodes to be removed from the contact calculations once all surrounding elements have failed (and can result in computational savings). The momentum transfer associated with free-flying nodes is expected to be significant in this example, so nodal erosion is not recommended.

Results and discussion

Deformed shapes at different stages of the analysis are shown in Figure 2.1.4–2 through Figure 2.1.4–5 (only active elements are shown in these figures). As shown in Figure 2.1.4–5, the projectile eventually perforates the plate, with approximately the leading half of the projectile elements failing during the analysis. Some broken-off fragments with active elements can be observed in Figure 2.1.4–5. The nodes and exposed faces of such fragments can take part in contact. Nodes no longer attached to any active elements can take part in contact only in the analysis without nodal erosion (which corresponds to the primary input file). Figure 2.1.4–6 compares total kinetic energy histories for the analyses conducted with and without nodal erosion, respectively. Approximately 32% of the initial kinetic energy is absorbed by the impact for the model without nodal erosion; whereas approximately 26% of the initial kinetic energy is absorbed for the model with nodal erosion.

Input files

erode_material.inp	External file referenced in this input (material definition).
erode_proj_and_plate.inp	Model of impact of eroding projectile into eroding plate without nodal erosion.
erode_proj_and_plate2.inp	Model of impact of eroding projectile into eroding plate with nodal erosion.

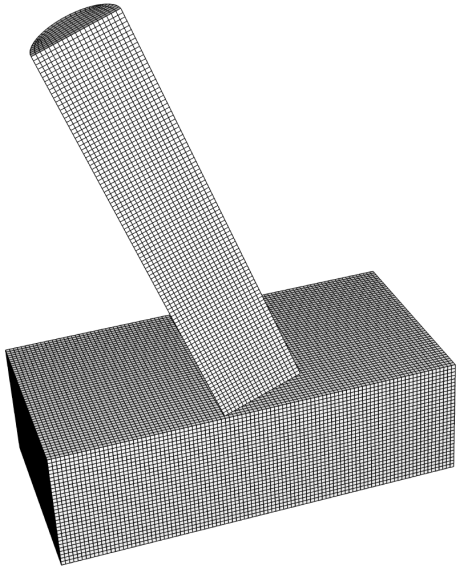


Figure 2.1.4–1 Undeformed mesh.

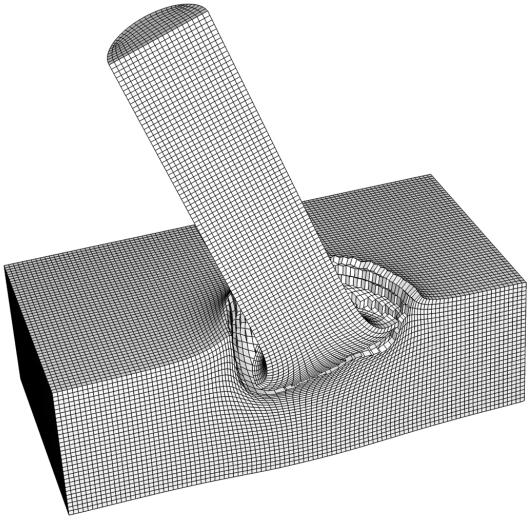


Figure 2.1.4–2 Deformed shape at 1.5 microseconds.

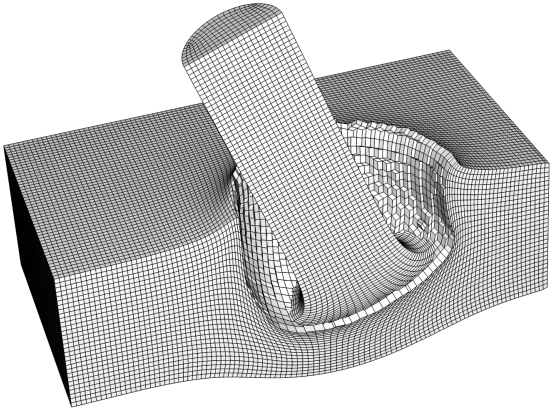


Figure 2.1.4–3 Deformed shape at 3 microseconds.

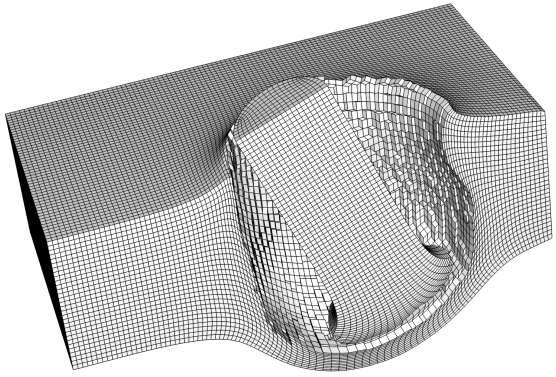


Figure 2.1.4–4 Deformed shape at 4.5 microseconds.

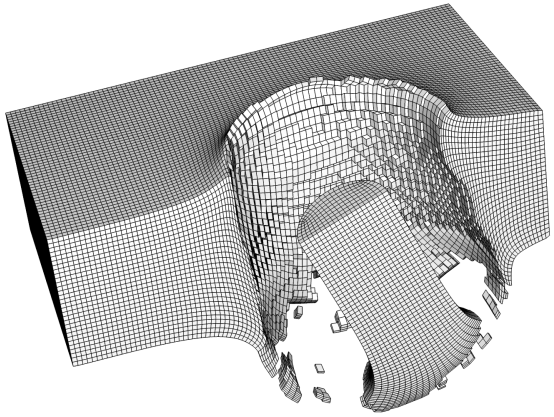


Figure 2.1.4–5 Deformed shape at 6 microseconds.

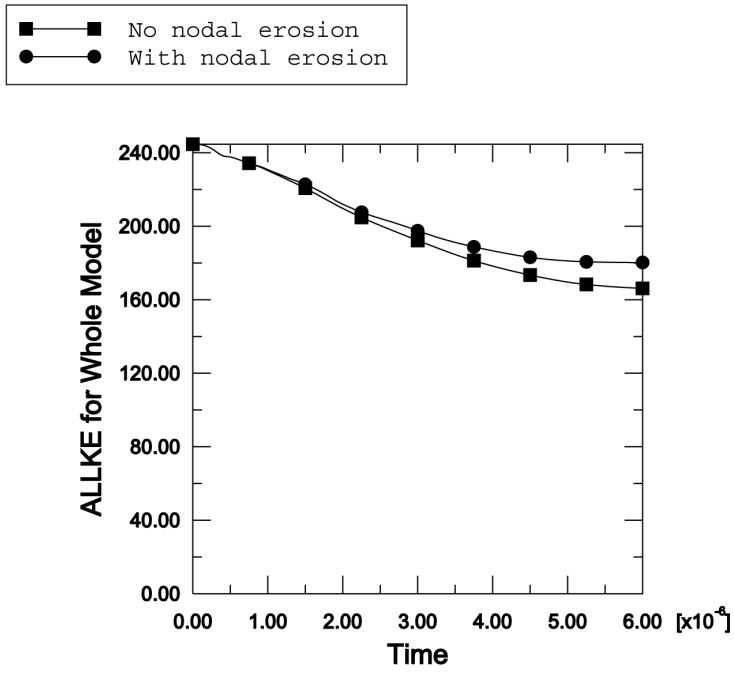


Figure 2.1.4–6 Kinetic energy history.

2.1.5 TENNIS RACKET AND BALL

Product: Abaqus/Explicit

This example simulates the oblique impact of a tennis ball onto a racket at 6.706 m/sec (264 in/sec). The example illustrates contact between a deforming surface and a node set, the definition of initial stresses, and modeling of a fluid cavity filled with a compressible gas using the surface-based fluid cavity capability.

Problem description

The strings on the tennis racket are modeled using T3D2 truss elements. They are assumed to be linear elastic, with Young's modulus of 6.895 GPa (1.0×10^6 psi), Poisson's ratio of 0.3, and density of 1143 kg/m^3 (1.07×10^{-4} lb sec^2 in^{-4}). The strings are under an initial tension of 44.48 N (10 lb), which is specified using initial stress conditions.

The frame is assumed to be rigid and is modeled using R3D4 elements. The nodes of the strings (truss elements) around the perimeter are the same nodes as those used for the R3D4 elements. The reference node for the rigid frame has boundary conditions applied to constrain all six degrees of freedom on the rigid body so that the frame does not move.

The tennis ball is modeled as a sphere, using 150 S4R shell elements. It is assumed to be made of rubber, modeled with general hyperelastic material properties as a Mooney-Rivlin material with the constants $C_{10} = 0.690$ MPa (100 lb/in^2) and $C_{01} = 0.173$ MPa (25 lb/in^2). Abaqus/Explicit requires some compressibility for hyperelastic materials. In the results shown here, $D_1 = 0.0145$ MPa^{-1} (10^{-4} psi^{-1}). This gives an initial bulk modulus ($K_0 = 2/D_1$) that is 80 times the initial shear modulus $2(C_{10} + C_{01})$. This ratio is lower than the ratio for typical rubbers, but the results are not particularly sensitive to this value in this case because the rubber is unconfined. A more accurate representation of the material's compressibility would be needed if the rubber were confined by stiffer adjacent components or reinforcement. Decreasing D_1 by an order of magnitude (thus increasing the initial bulk modulus by a factor of 10) has little effect on the overall results but causes a reduction in the stable time increment by a factor of $\sqrt{10}$ due to the increase in the bulk modulus. The density of the tennis ball is 1068 kg/m^3 (1.07×10^{-4} lb sec^2 in^{-4}).

The tennis ball is under an initial internal pressure of 41 kPa (6 psi) in addition to the ambient atmospheric pressure of 100 kPa (14.7 psi). An element-based surface is defined on the inside of the tennis ball. This surface is used to define a fluid cavity filled with gas. The properties of the gas inside the tennis ball, molecular weight and molar heat capacity, are defined as part of the fluid behavior of a fluid cavity. The molecular weight and molar heat capacity of the gas are arbitrarily chosen as 0.062 kg (0.1367 lb) and 28.110 J/kg °K (112.847 lb in/lbm °K). Since the ball is impermeable to gas, the pressure of the gas will rise when the volume of the ball decreases, and vice versa. Static equilibrium gives the value of the initial biaxial membrane stresses in the shell elements of the sphere as $pr/2t = 155$ kPa (22.5 psi) to balance the internal pressure (here p is the internal gas pressure, r is the radius of the sphere, and t is the tennis ball thickness). This initial state of stress in the ball is defined using initial stress conditions.

A coefficient of friction of 0.1 is specified between the ball and the strings. The ball impacts on the strings at 6.706 m/sec (264 in/sec) at an angle of 15°.

No attempt has been made to generate an accurate model of the ball and strings: the model parameters are chosen simply to provide a "soft" ball relative to the strings to illustrate contact effects.

The complete model is shown in Figure 2.1.5–1. There are 2241 degrees of freedom in the model.

An element-based surface is defined on the tennis ball. Since the truss elements are line elements, they do not form a planar surface. A node-based surface is defined that contains all the nodes of the strings. Contact pairs are then used to define contact between the element-based surface of the ball and any of the nodes defined in the node-based surface. An input file that uses the general contact algorithm is also provided.

Results and discussion

Figure 2.1.5–2 shows the position of the ball with respect to the strings in the undeformed configuration. The deformed shapes at different stages of the analysis are shown in Figure 2.1.5–3 through Figure 2.1.5–7. Figure 2.1.5–8 shows a time history of the energies for the model. These include the total internal energy (ALLIE), the kinetic energy (ALLKE), the viscous dissipation (ALLVD), the energy dissipated by friction (ALLFD), the external work (ALLWK), and the total energy balance for the model (ETOTAL). The total energy is seen to remain almost constant during the analysis, as it should. Figure 2.1.5–9 and Figure 2.1.5–10 give the history of pressure inside the ball and the history of the actual volume of the ball. It can be seen that both the gas pressure inside the ball and the ball volume stabilize after 10 msec.

Input files

tennis_surfcav.inp	Analysis using the contact pair approach.
tennis_gcont_surfcav.inp	Analysis using the general contact capability.
tennis_ef1.inp	External file referenced in all analyses.
tennis_ef2.inp	External file referenced in all analyses.

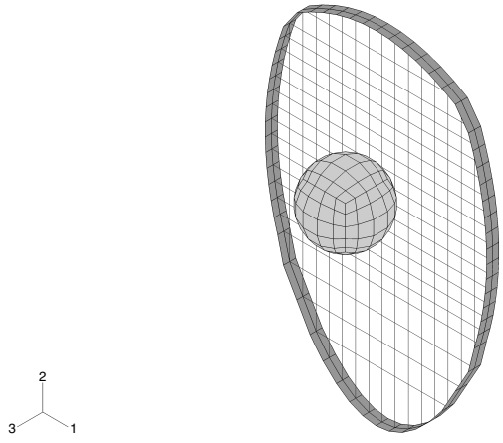


Figure 2.1.5–1 Undeformed mesh.

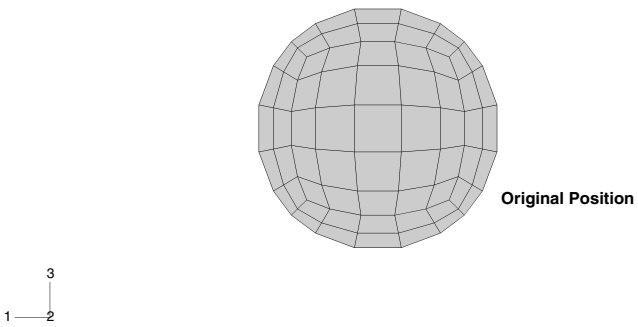


Figure 2.1.5–2 Original position of ball and strings.

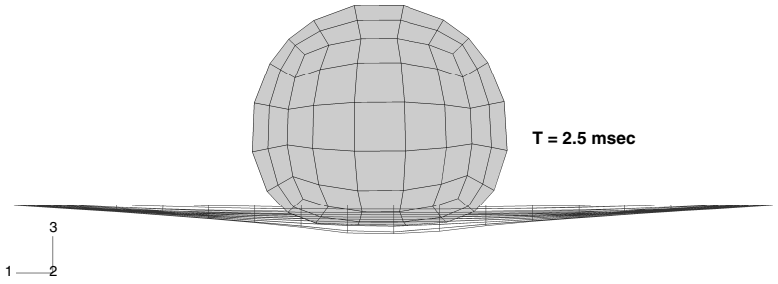


Figure 2.1.5–3 Deformed shape at 2.5 milliseconds.

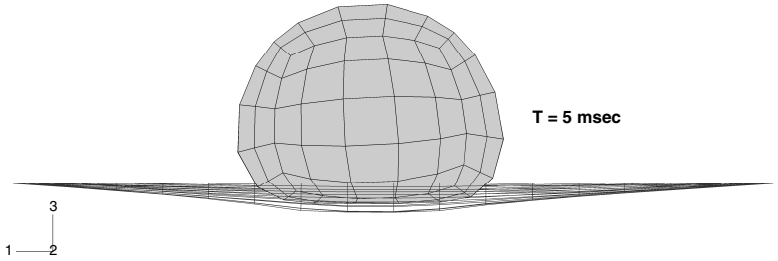


Figure 2.1.5–4 Deformed shape at 5 milliseconds.

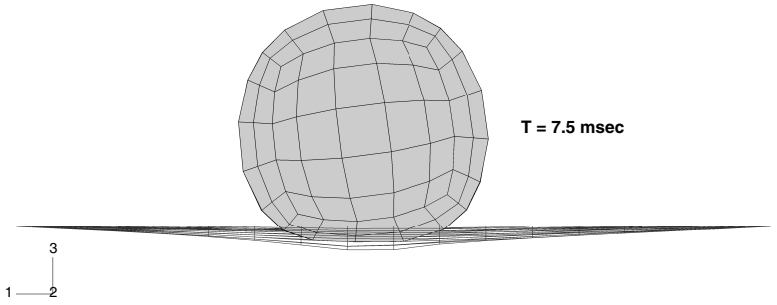


Figure 2.1.5–5 Deformed shape at 7.5 milliseconds.

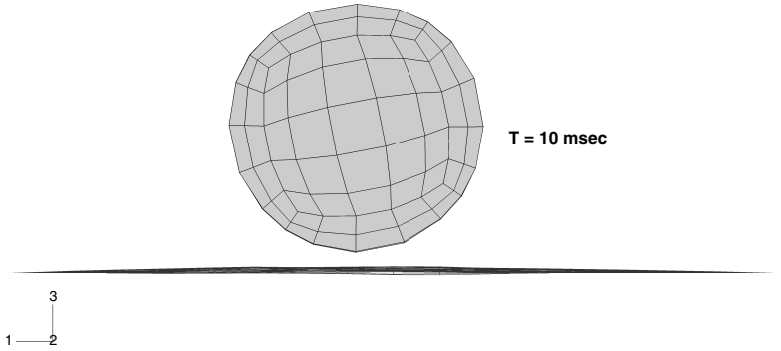


Figure 2.1.5–6 Deformed shape at 10 milliseconds.

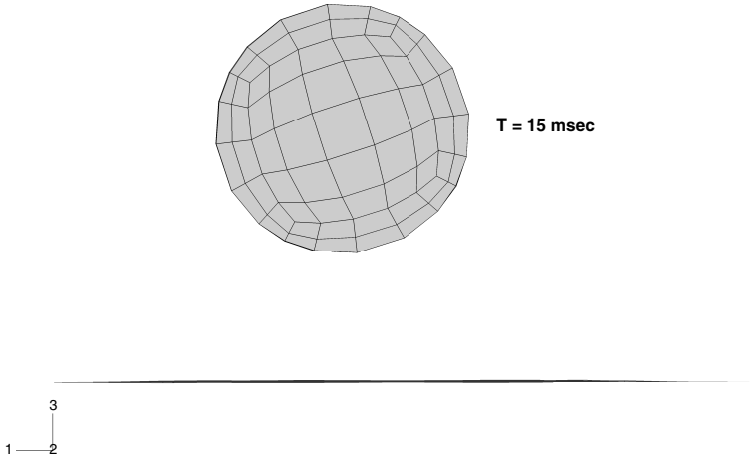


Figure 2.1.5–7 Deformed shape at 15 milliseconds.

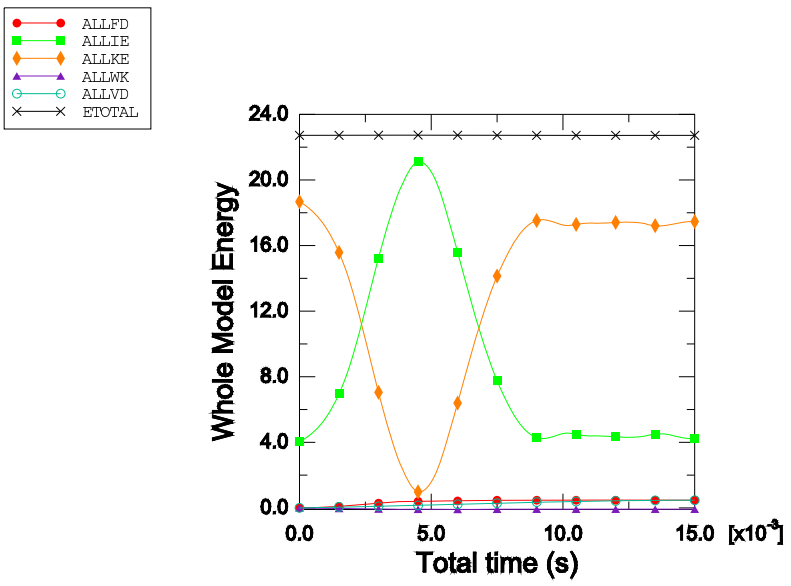


Figure 2.1.5–8 Energy histories.

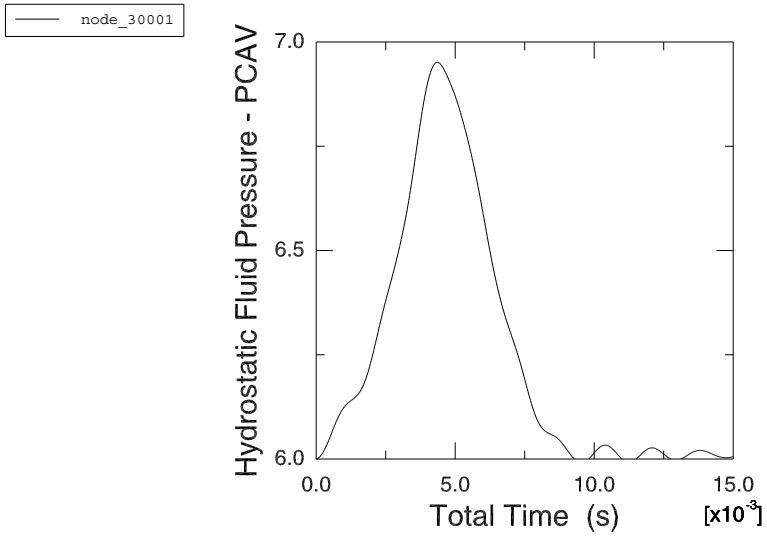


Figure 2.1.5–9 History of the gas pressure inside the tennis ball.

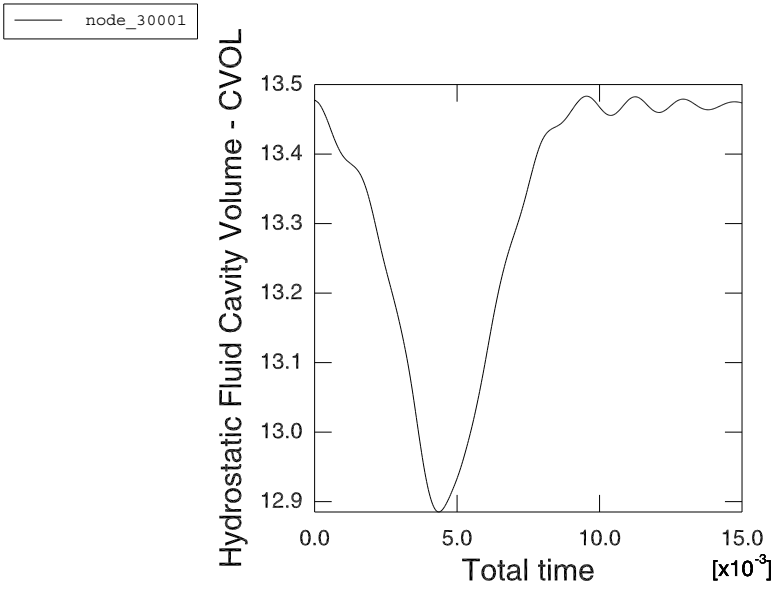


Figure 2.1.5–10 History of the ball volume.

2.1.6 PRESSURIZED FUEL TANK WITH VARIABLE SHELL THICKNESS

Product: Abaqus/Standard

This problem demonstrates the variable shell thickness capability in Abaqus. The example is based on an analysis conducted by SOLVAY RESEARCH & TECHNOLOGY (see reference) of a blow-molded, plastic fuel tank with dimensions similar to those considered here.

Geometry and model

The mesh shown in Figure 2.1.6–1 is used in this example to model a fuel tank and its support straps. The mesh uses 2812 3-node shell elements (S3R), with the support straps modeled with 32 2-node beam elements (B31). Depending on the desired accuracy and detail of the solution, the analyst may identify some regions of the mesh in which additional refinement, or second-order elements, would be appropriate. The fuel tank would fit within a box of dimensions 450 mm × 200 mm × 680 mm. An internal pressure of 7×10^{-3} MPa is applied statically to the tank.

Analyses are conducted for a uniform shell thickness of 5 mm and for a spatially varying shell thickness in the range 1.38 mm to 9.35 mm (see Figure 2.1.6–2), which is a more accurate representation of the tank. The uniform thickness analysis provides a comparison to judge the effects of variable thickness. The overall volume of plastic modeled in the variable thickness analysis is about 93% of that in the uniform thickness analysis. For the variable thickness analysis the shell cross-section indicates that the shell thickness is to be interpolated from nodal values specified with nodal thickness. For elements with more than one integration point, this approach results in a thickness that can vary over the element.

The materials are modeled as isotropic elastic. The plastic fuel tank has a Young's modulus of 0.6 GPa and a Poisson's ratio of 0.3. The steel support straps have a Young's modulus of 206.8 GPa and a Poisson's ratio of 0.29. Geometrically nonlinear effects are significant in this example, so geometric nonlinearity is included in the step.

Results and discussion

Contour plots of the Mises stress at the inner surface of the fuel tank (section point 1 in the shell elements) for the variable shell thickness and uniform shell thickness analyses are shown in Figure 2.1.6–3 and Figure 2.1.6–4. The ratio of the maximum Mises stress found in the variable thickness analysis to that found in the uniform thickness analysis is 1.5. For the variable thickness analysis, the maximum Mises stress occurs at a location where the fuel tank skin is relatively thin (see Figure 2.1.6–2 and Figure 2.1.6–3).

Examination of the y-component of displacement shows that the overall expansion of the tank in the y-direction is about 1.5% greater in the variable thickness analysis.

Input files

pressfueltank_variablethick.inp	Example using variable shell thickness.
pressfueltank_uniformthick.inp	Example using uniform shell thickness.
pressfueltank_node.inp	Nodal coordinate data for both models.

pressfueltank_shellelement.inp	Shell element connectivity data for both models.
pressfueltank_beamelement.inp	Beam element connectivity data for both models.
pressfueltank_shellthickness.inp	Shell thickness data for the variable shell thickness model.

Reference

- SOLVAY RESEARCH & TECHNOLOGY, Plastic Processing Department, Rue de Ransbeek, 310, B-1120 Brussels, Belgium.

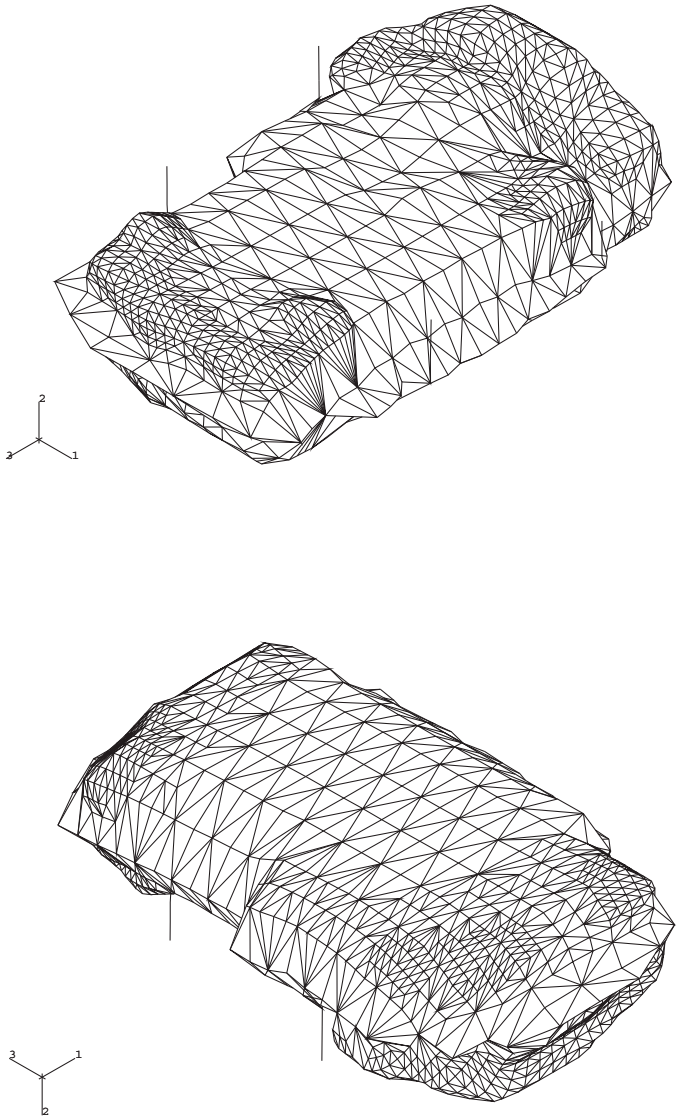


Figure 2.1.6–1 Fuel tank mesh with S3R and B31 elements.

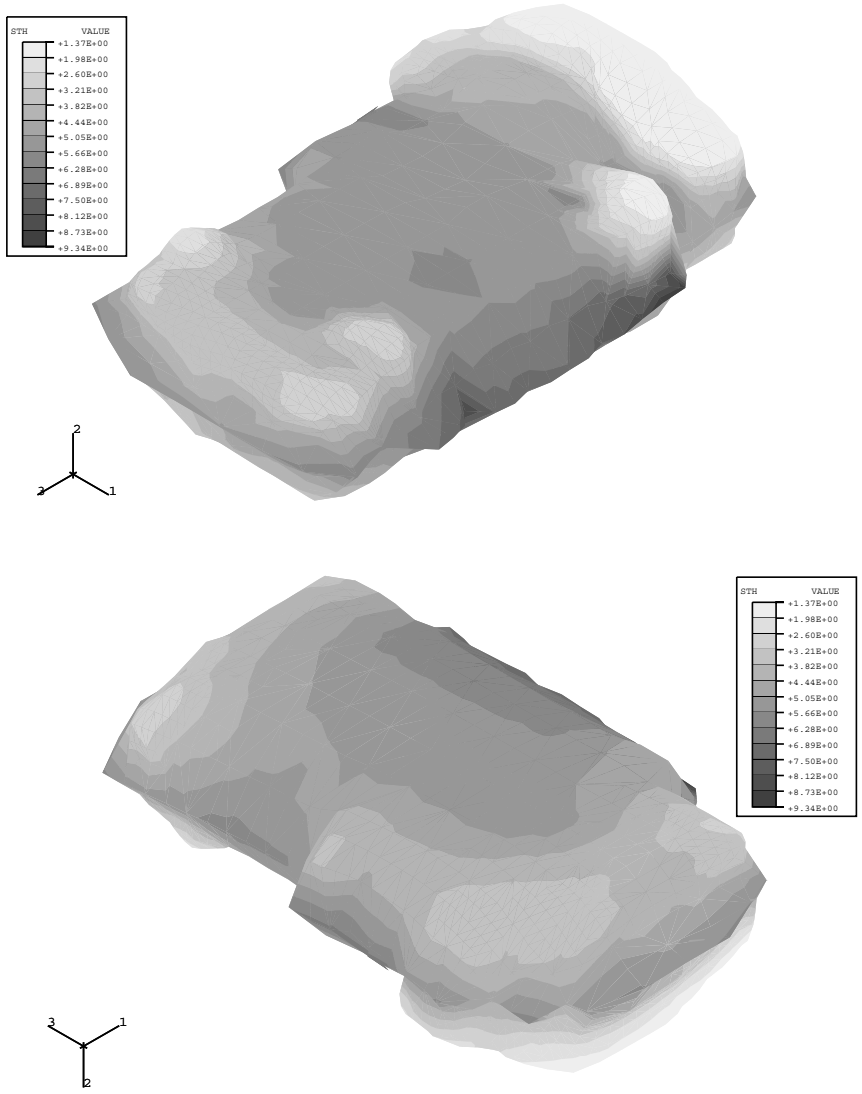


Figure 2.1.6–2 Shell thickness for variable thickness analysis.

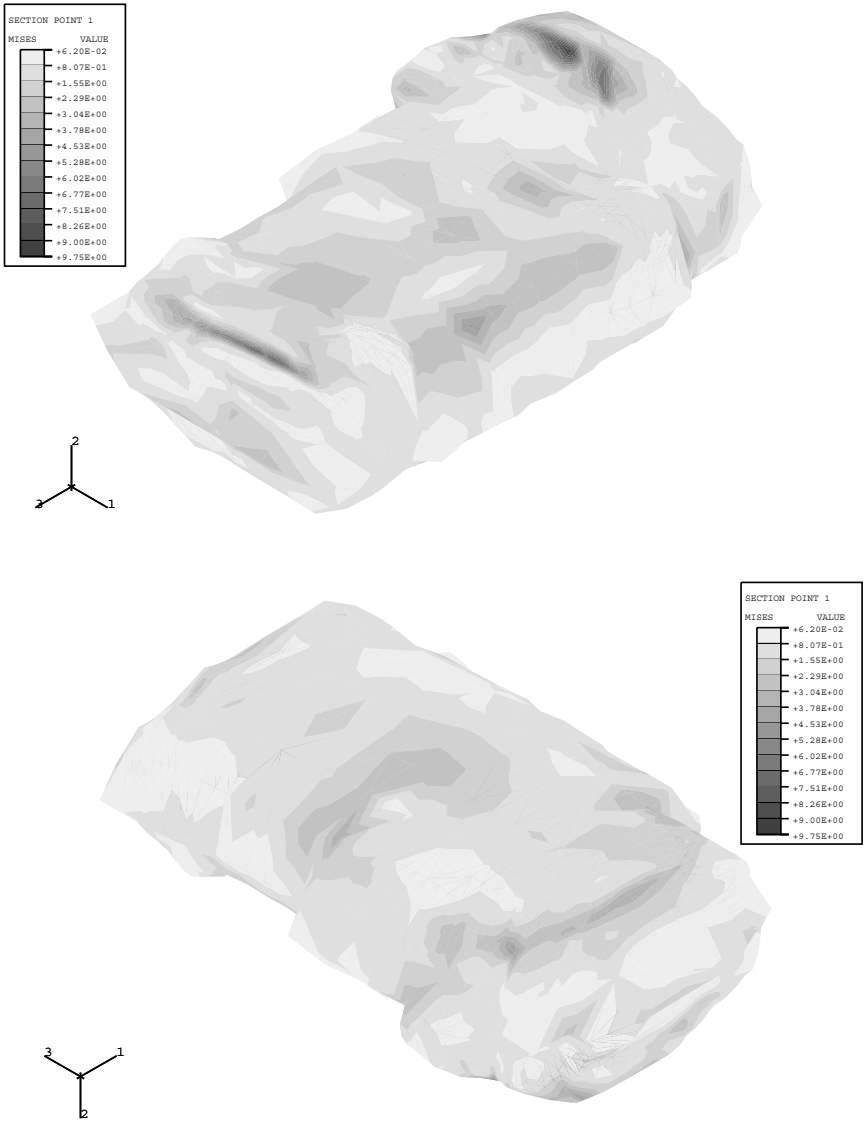


Figure 2.1.6–3 Mises stress solution for variable shell thickness analysis.

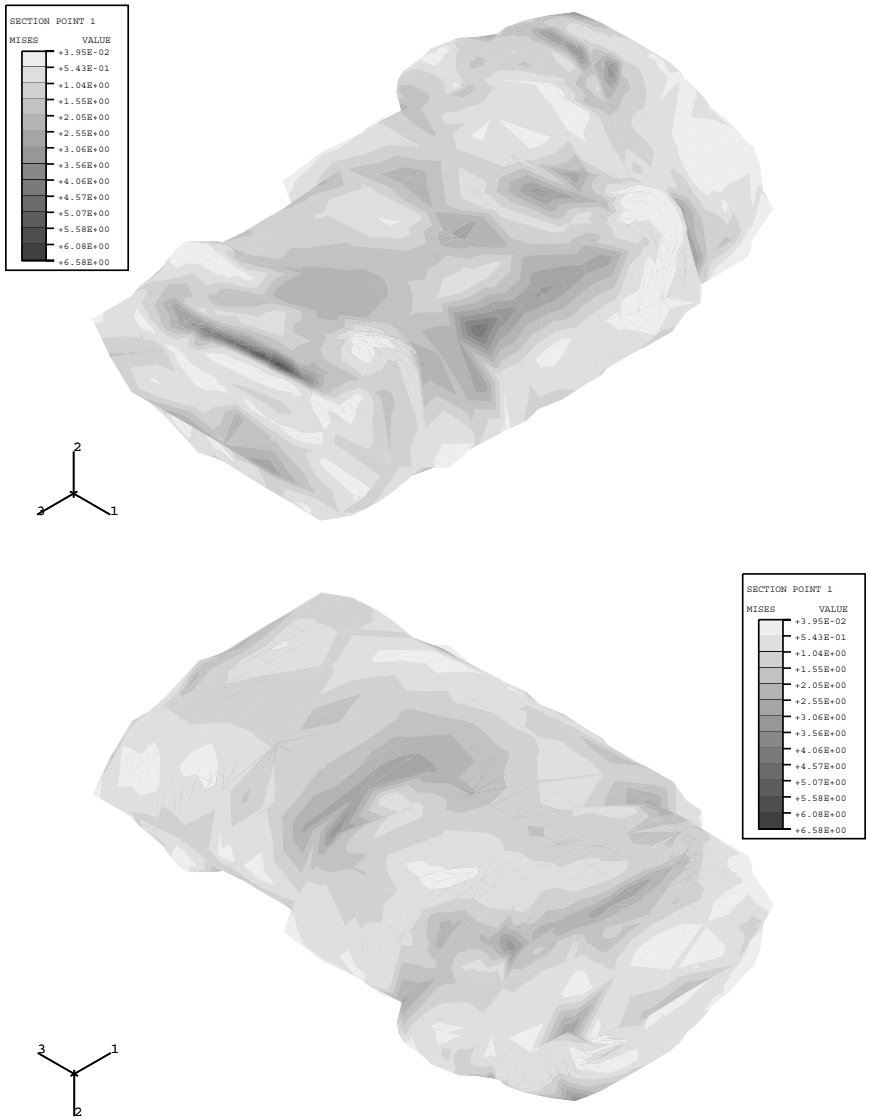


Figure 2.1.6–4 Mises stress solution for uniform shell thickness analysis.

2.1.7 MODELING OF AN AUTOMOBILE SUSPENSION

Product: Abaqus/Standard

This example illustrates the use of JOINTC elements. It is repeated to illustrate the use of connector elements. JOINTC elements ("Flexible joint element," Section 32.3.1 of the Abaqus Analysis User's Guide) can be used to model the interaction between two nodes that are almost coincident geometrically and that represent a joint that has internal stiffness and/or damping. The behavior of the joint is defined in a local coordinate system ("Orientations," Section 2.2.5 of the Abaqus Analysis User's Guide). This system rotates with the motion of the first node of the element and may consist of linear or nonlinear springs and dashpots arranged in parallel, coupling the corresponding components of relative displacement and of relative rotation in the joint. This feature can be used to model, for example, a rubber bearing in a car suspension.

In the connector element model the JOINTC elements are replaced by connector elements (see "Connector elements," Section 31.1.2 of the Abaqus Analysis User's Guide) with connection types CARTESIAN to define the translational behavior and ROTATION to define the rotational behavior. These connection types allow linear or nonlinear spring and dashpot behavior to be defined in a local coordinate system that rotates with the first node on the element. Several different connection types can be used to model the finite rotational response. See "Connection-type library," Section 31.1.5 of the Abaqus Analysis User's Guide, for connection types using different finite rotation parametrizations. In this model the rotation magnitudes are assumed small. Hence, a rotation vector parametrization of the joint using ROTATION is appropriate.

The primary objective of this example is to verify the accuracy of JOINTC and connector elements in a structure undergoing rigid rotation motions. A secondary objective of this example is to demonstrate the use of equivalent rigid body motion output variables in Abaqus/Standard.

Geometry and model

The structure analyzed is an automobile's left front suspension subassembly (see Figure 2.1.7–1). The physical components included in the assemblage are the tire, the wheel, the axle (hub), the A-arm (wishbone), the coil spring, and the frame. The tire is modeled with a JOINTC or connector element; a curved bar element has been attached for visualization. (The JOINTC or connector element is used because it is a convenient way of defining the tire's nonlinear stiffness in a local coordinate system.) The vertical stiffness of the wheel is represented by a beam element. The axle and A-arm are both modeled with beam elements. The axle is connected to the A-arm by a pin-type MPC in the JOINTC model or with connection type JOIN in the connector model. The coil spring is modeled by a SPRINGA element in the JOINTC model or with connection type AXIAL in the connector model. The automobile frame is represented by a MASS element. The top of the coil spring is connected directly to the frame, while the A-arm is connected to the frame by two JOINTC elements or two connector elements with connection types CARTESIAN and ROTATION (representing the A-arm bushings). The initial position represents a fully weighted vehicle, and the tire and coil spring have a corresponding initial preload.

The first step in the analysis allows the suspension system to reach equilibrium. The second step models the tire moving over a bump in the road. The bump is idealized as a triangular shape 100 mm

high by 400 mm long, and the vehicle is assumed to travel at 5 km/hr. A second input file is used to show the effects of large rotation on the suspension response. This file includes an initial rotation step, which rigidly rotates the model by 90° about the vertical axis but is otherwise identical to the first input file. We expect the response from the two analyses to be the same.

In this example we are primarily interested in the equivalent rigid body motion of the A-arm and, in particular, in the average displacement and rotation.

Results and discussion

The results from the connector element models are qualitatively and quantitatively equivalent to the JOINTC models. Hence, only the JOINTC results are discussed further.

The vertical displacement histories in Step 2 are shown in Figure 2.1.7–2 for the contact point of the tire with the ground, the wheel center, and the frame. Figure 2.1.7–3 shows a series of overlaid displaced plots as the tire rolls up the bump.

To verify the behavior of the JOINTC elements with large rotations, the second model rigidly rotates the entire structure by 90° before applying the bump excitation. Figure 2.1.7–4 shows the displacement time histories from the two models overlaid on the same plot. They are nearly identical.

In the analysis of deformable bodies undergoing large motions it is convenient to obtain information about the equivalent rigid body motions: average displacement and rotation, as well as linear and angular momentum about the center of mass. For this purpose Abaqus provides a set of equivalent rigid body output variables. As indicated above, a secondary objective of this example is to demonstrate the use of these output variables, which represent the average motion of the specified element set. Element output requests to the results file and to the data file are included. If no element set is specified, the average motion of the entire model is given. This type of output can only be requested in a direct-integration implicit dynamic analysis, and only elements that have a mass will contribute to the equivalent rigid body motion. For a precise definition of the equivalent rigid body motion of a deformable body, see "Equivalent rigid body dynamic motion," Section 2.4.4 of the Abaqus Theory Guide.

Figure 2.1.7–5 and Figure 2.1.7–6 have been generated using the equivalent rigid body output variables. Figure 2.1.7–5 shows the vertical motion of node 5001 (bearing point "A" of the A-arm), node 5080 (point of A-arm nearest the tire), and the average vertical motion of the A-arm (output variable UC3). As expected, the displacement of the center of mass of the component lies between the displacements of its two ends. Figure 2.1.7–6 shows the average rigid body rotation of the A-arm component about its center of mass. Rigid body rotations are available about the three global axes. Here we are interested in the rotation about the global X-axis (output variable URC1). The standard output for rotations is radians, but the results have been scaled to plot the rotation in degrees.

Input files

jointcautosuspension.inp	Suspension analysis with JOINTC elements.
jointcautosuspension_rotated.inp	Rotated suspension analysis with JOINTC elements. This input file includes one extra (rotation) step but is otherwise identical to jointcautosuspension.inp.
jointcautosuspension_depend.inp	Identical to jointcautosuspension.inp, except that field-variable-dependent linear and nonlinear spring properties are used in the JOINTC elements.
connautosuspension.inp	Suspension analysis with connector elements.
connautosuspension_rotated.inp	Rotated suspension analysis with connector elements. This input file includes one extra (rotation) step but is otherwise identical to connautosuspension.inp.
connautosuspension_depend.inp	Identical to connautosuspension.inp, except that field-variable-dependent linear and nonlinear spring properties are used in the connector elements with connection types CARTESIAN, ROTATION, and AXIAL.

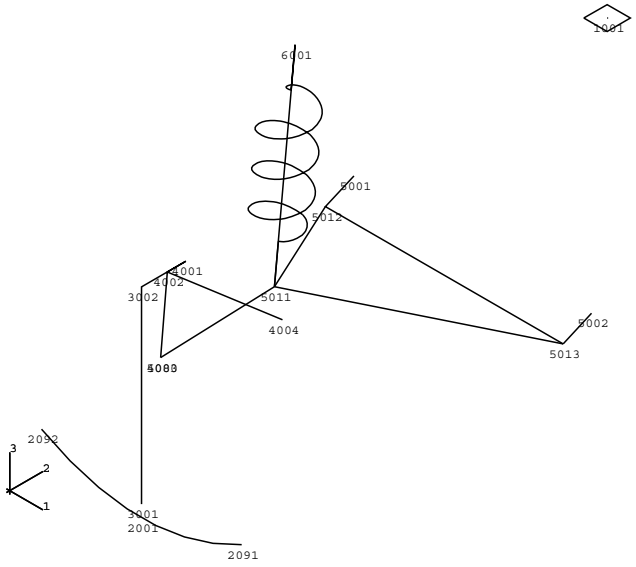


Figure 2.1.7–1 Left front automobile suspension.

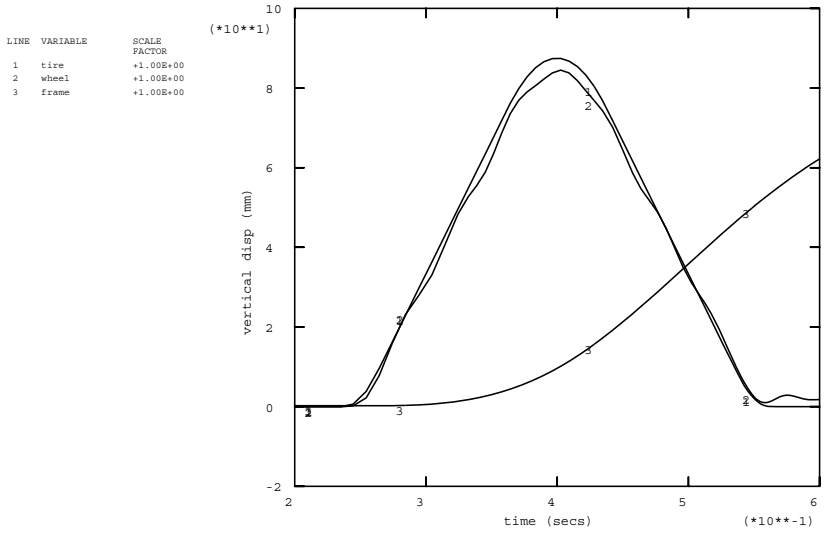


Figure 2.1.7–2 Displacement histories of tire, wheel center, and frame.

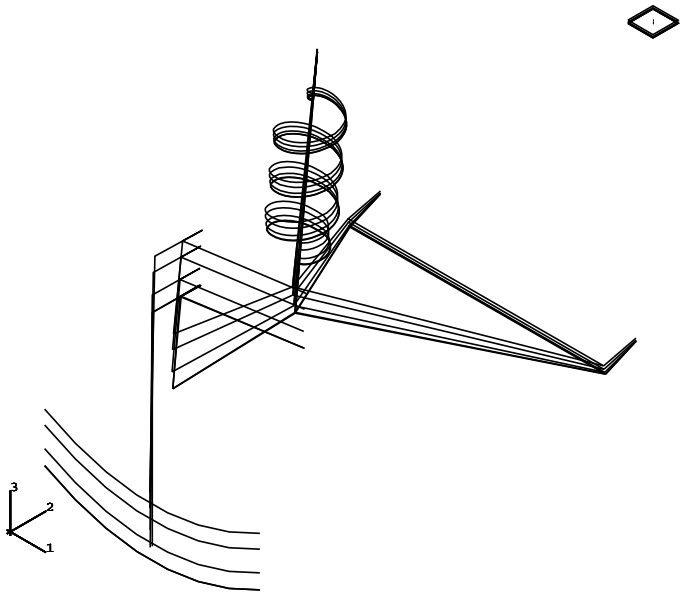


Figure 2.1.7–3 Displaced shapes during positive vertical tire motion.

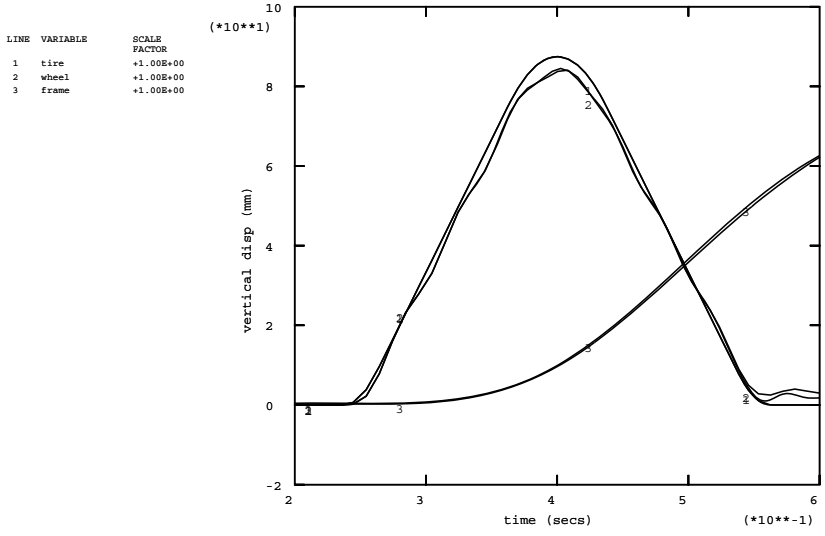


Figure 2.1.7–4 Overlay of unrotated and rotated suspension analyses.

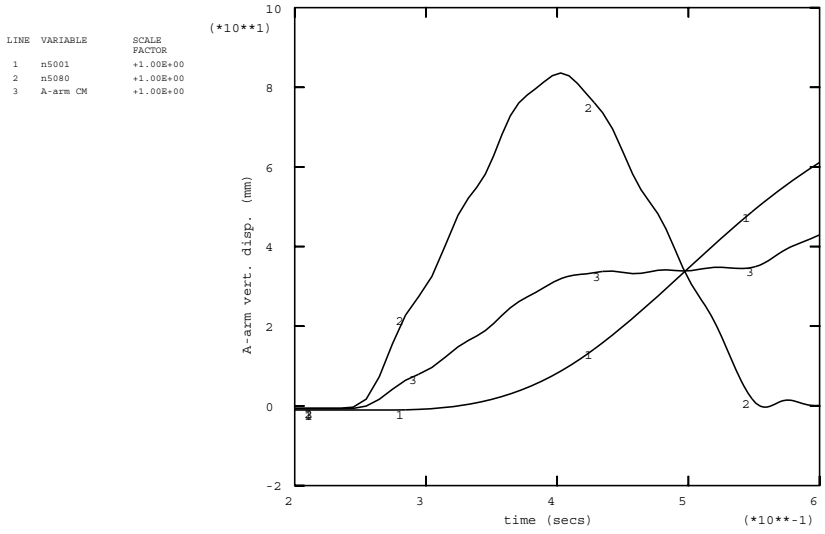


Figure 2.1.7–5 Vertical motion of the A-arm: average and nodal motions.

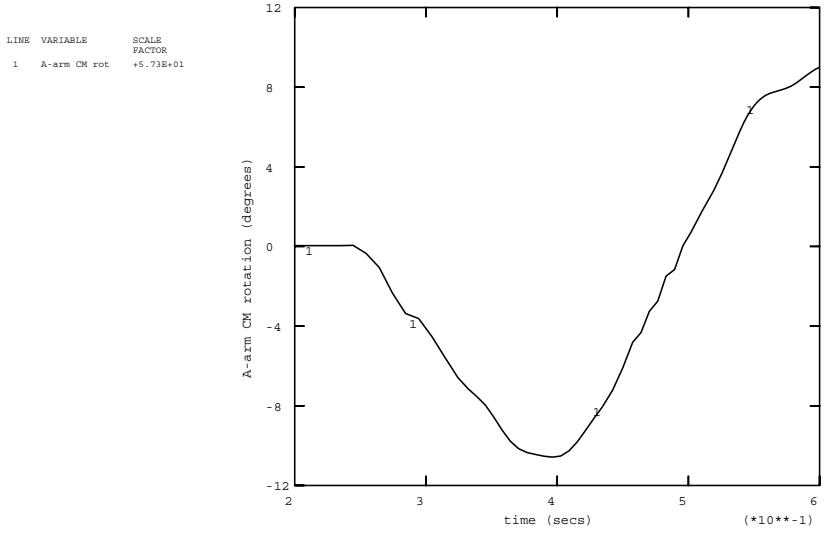


Figure 2.1.7–6 Average rotation of the A-arm about its center of mass.

2.1.8 EXPLOSIVE PIPE CLOSURE

Product: Abaqus/Explicit

This problem illustrates the following concepts: large deformation kinematics, equations of state, elastic-plastic material, transformations, detonation points.

Problem description

The units used in this analysis are referred to as c.g.μsec. Using these units, length is given in centimeters (cm), mass in grams (gm), and time is measured in microseconds (μsec). The stresses have units of mega bars (M bar). These units are commonly used in shock wave physics applications because the pressures tend to have values on the order of unity.

In this example problem two concentric pipes have the annulus between them filled with high explosive (HE). The inside radius of the inner pipe is 10 mm. The inside radius of the outer pipe is 20 mm. Both pipes are steel with a wall thickness of 2 mm. Each pipe is modeled with 6 elements in the radial direction, while the HE is modeled with 24 elements in the radial direction.

The steel pipe is an elastic, perfectly plastic material with Young's modulus of 221.1 GPa (2.211 M bar), Poisson's ratio of 0.279, yield strength of 430 MPa (.0043 M bar), and density of 7846 kg/m^3 (7.846 gm/cm^3).

The explosive material is modeled using the JWL equation of state with detonation wave speed = 7596 m/sec (.7596 cm/microsecond), A = 520.6 GPa (5.206 M bar), B = 5.3 GPa (0.053 M bar), R1 = 4.1, R2 = 1.2, ω = .35, density = 1900 kg/m^3 (1.9 gm/cm^3), and initial specific energy = 3.63 M joule/kg (0.0363 T erg/gm). The tension cutoff pressure is assumed to be zero and is specified using a tensile failure model. Refer to "Equation of state," Section 25.2.1 of the Abaqus Analysis User's Guide, for a description of this material model.

The explosive material is detonated at four points around the circumference of the cylinder. Because of the symmetry in this problem, only one-eighth of the pipe is modeled. Figure 2.1.8–1 shows the original geometry and the location of the detonation point for the model. A transformed coordinate system is used to define the symmetry conditions along the sloping boundary. The interface between the explosive material and the steel is modeled using the no separation relationship for contact that allows for relative slip without separation between the two materials.

This analysis is run in two steps to reduce the amount of output written to the output database file. In the early part of the analysis, the deformations are not of much interest. Hence, the first step has a duration of 6 μsec. After 6 μsec the deformations are becoming significant. The second step has a duration of 1.5 μsec.

This analysis is run as both a two-dimensional case using CPE4R elements and as a three-dimensional case using C3D8R elements. In the three-dimensional case the displacements are constrained to be zero in the out-of-plane direction.

Results and discussion

Figure 2.1.8–2 through Figure 2.1.8–5 show a sequence of the deformed shapes computed by Abaqus/Explicit for the two-dimensional case. The undeformed configuration is shown superimposed on the deformed shapes. Although not shown here, the results of the three-dimensional analysis are indistinguishable from those of the two-dimensional analysis.

This problem tests the features listed, but it does not provide independent verification of them.

Input files

eoscyl2d.inp	Two-dimensional case.
eoscyl3d.inp	Three-dimensional case.

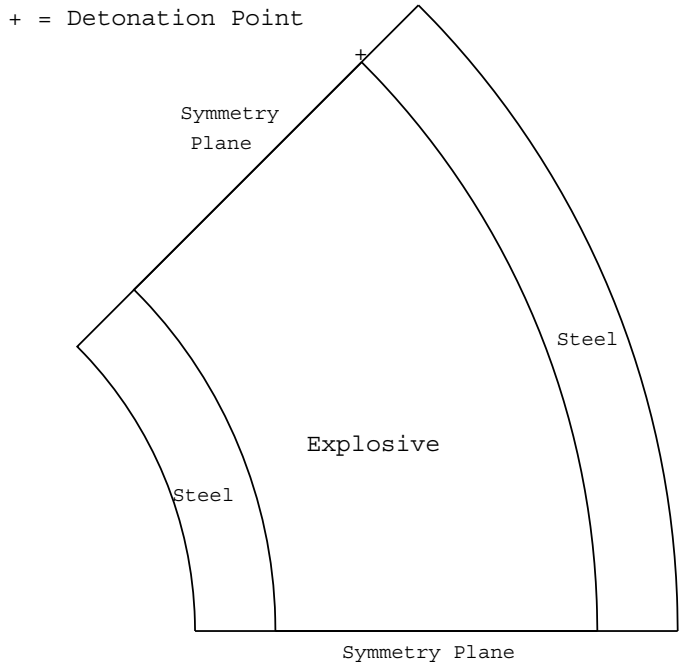


Figure 2.1.8–1 Original geometry.

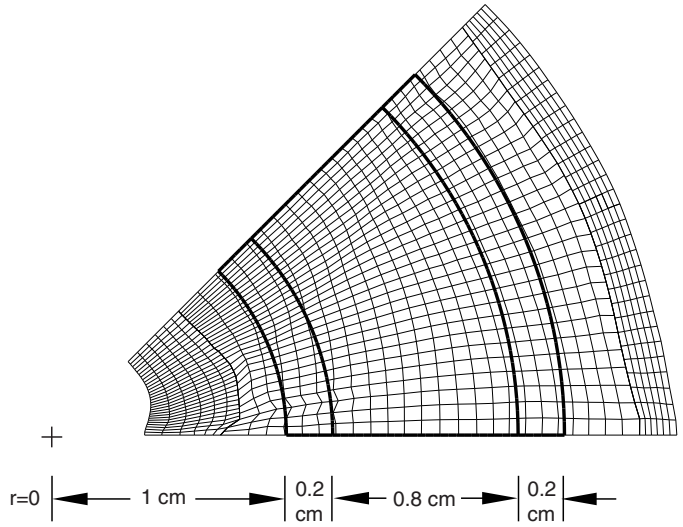


Figure 2.1.8–2 Deformed configuration after 6.0 μsec with undeformed configuration superimposed.

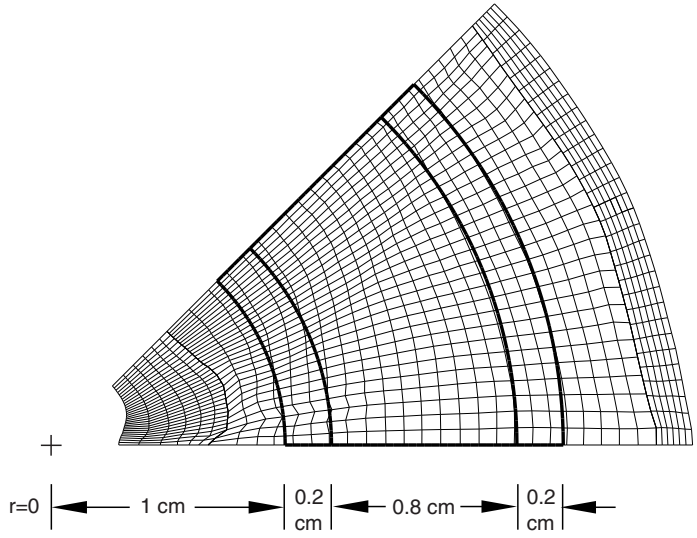


Figure 2.1.8–3 Deformed configuration after 6.5 μsec with undeformed configuration superimposed.

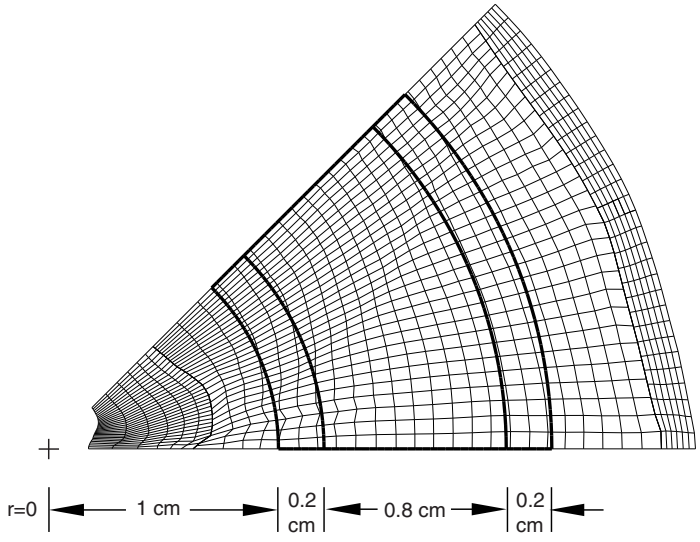


Figure 2.1.8–4 Deformed configuration after 7.0 μsec with undeformed configuration superimposed.

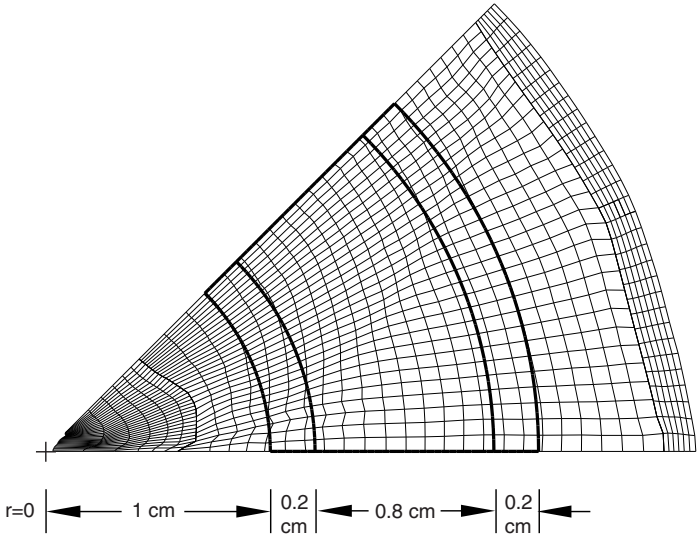


Figure 2.1.8–5 Deformed configuration after 7.5 μsec with undeformed configuration superimposed.

2.1.9 KNEE BOLSTER IMPACT WITH GENERAL CONTACT

Product: Abaqus/Explicit

This example illustrates the use of the general contact capability in a simulation involving large relative motion between potentially contacting surfaces.

Problem description

The model represents an automobile knee bolster assembly—the portion of the instrument panel that the occupant's legs impact in the event of a crash. The assembly consists of a hard plastic cover (the knee bolster) supported by a stiff steel substructure. Proper design of this assembly ensures that the occupant's energy is dissipated with a minimum of injury-causing forces. In this simulation the legs approach the knee bolster at 6 m/s, representing unrestrained motion following a 15 mph to dead stop crash event.

The components of the instrument panel are modeled using S3R and S4R shell elements. The bolster is made up of 2690 shell elements, with the material modeled as a von Mises, elastic strain hardening plastic material with a Young's modulus of 2.346 GPa, a Poisson's ratio of 0.4, a density of 1140 kg/m^3, and a yield stress of 11.7 MPa. The steel substructure is made up of 1648 elements, with the material modeled as a strain hardening steel with a Young's modulus of 207 GPa, a Poisson's ratio of 0.3, a density of 7700 kg/m^3, and a yield stress of 207 MPa. Figure 2.1.9–1 shows the model geometry from the rear of the knee bolster prior to impact, and Figure 2.1.9–2 shows the knee bolster and knee/leg assembly from a position outboard and behind the driver prior to impact.

The legs are represented as structural members with a surrounding rigid surface. The structural members, representing the bones, are modeled with B31 beam elements and T3D2 truss elements, with the material modeled as elastic with a Young's modulus of 207 GPa, a Poisson's ratio of 0.3, and a density of 7.7 kg/m^3. The rigid surfaces, representing the knee and shin, are modeled with R3D4 rigid elements. The body mass is modeled by distributing mass elements at various locations among the nodes of the structural elements.

In the input file knee_bolster_nsm.inp the upper body mass is distributed over the hip bones (B31 beam elements) as a nonstructural mass instead of point masses. The nonstructural mass contribution to an element increases both the element mass and the element rotary inertia, thus resulting in an increased stable time increment. This analysis completes in about 40% fewer increments when the nonstructural mass feature is used since the time increment for the problem is controlled by the beam elements. In the input file knee_bolster_massadjust.inp, the upper body mass is distributed over the hip bones using mass adjustment instead of point masses. The mass distribution and analysis results are similar to those using nonstructural mass. Input files for similar analyses using pipe elements (PIPE31) instead of beam elements are included.

Potential contact among the instrument panel assembly components and between the instrument panel and the legs is modeled using the general contact capability. The general contact inclusions option to automatically define an all-inclusive surface is used and is the simplest way to define contact in the model. In addition, a model that uses the alternative contact pair algorithm is provided; the contact definition is more tedious with the contact pair algorithm.

Initial velocities are defined on the leg components to approximate a 15 mph (6 m/s) crash condition. The hips are constrained to translate in the plane of the seat. The ankles are constrained consistent with fixed planting of the feet on the floor of the car. The dashboard substructure is fixed at locations where it would be welded to the automobile frame; deformations due to this impact are assumed to be confined to the explicitly modeled structure.

Results and discussion

Figure 2.1.9–3 shows the deformed shape of the bolster assembly after 100.0 ms.

Figure 2.1.9–4 shows the energy time history of the whole model: internal energy, kinetic energy, recoverable strain energy, and plastic dissipation. This figure shows that almost all of the body's initial kinetic energy has been transferred by the end of this simulation. Of this transferred amount a small amount has been transferred to elastic deformations in the instrument panel structure and bones, and the balance is lost to plastic dissipation.

Figure 2.1.9–5 shows the total knee and shin contact forces (filtered with the SAE 600 filter) measured against the displacement into the bolster. Consistent with the observations of the energy quantities, it is clear that the crash event is complete. The solution obtained using pipe elements is consistent with that using beam elements.

Acknowledgment

Abaqus would like to thank GE Plastics for supplying the model used in this example.

Input files

knee_bolster.inp	Input data for this analysis using the general contact capability.
knee_bolster_pipe.inp	Input file for similar analysis involving pipe elements and general contact.
knee_bolster_cpair.inp	Input data for this analysis using contact pairs.
knee_bolster_cpair_pipe.inp	Input file for similar analysis involving pipe elements and contact pairs.
knee_bolster_nsm.inp	Input data for this analysis using the general contact capability and the nonstructural mass capability to distribute the upper body mass over the hip bones.
knee_bolster_nsm_pipe.inp	Input file for similar analysis involving pipe elements, general contact, and nonstructural mass.
knee_bolster_massadjust.inp	Input data for this analysis using the general contact capability and mass adjustment to distribute the upper body mass over the hip bones.
knee_bolster_massadjust_pipe.inp	Input file for similar analysis involving pipe elements, general contact, and mass adjustment.
knee_bolster_ef1.inp	External file referenced by this analysis.
knee_bolster_ef2.inp	External file referenced by this analysis.
knee_bolster_ef3.inp	External file referenced by this analysis.

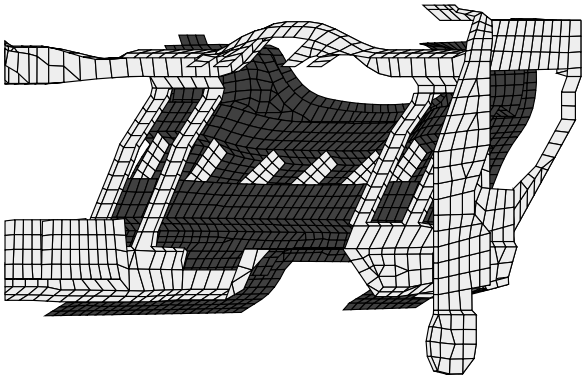


Figure 2.1.9–1 Initial configuration of the knee bolster model (view from behind the bolster).

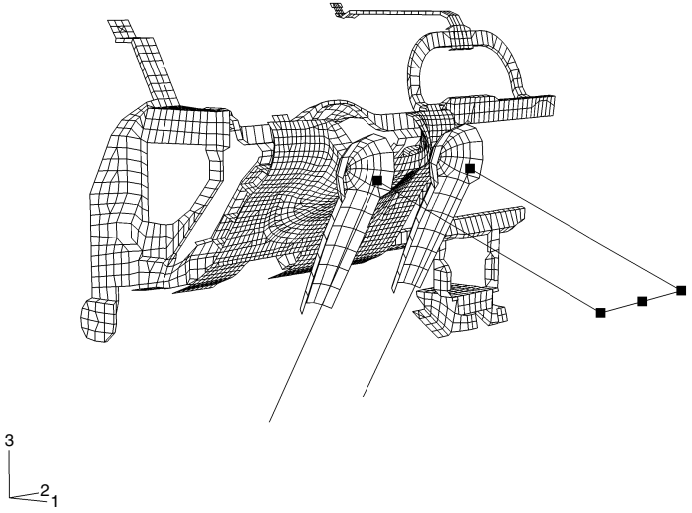


Figure 2.1.9–2 Initial configuration of the knee bolster model (view from outboard and behind the driver).

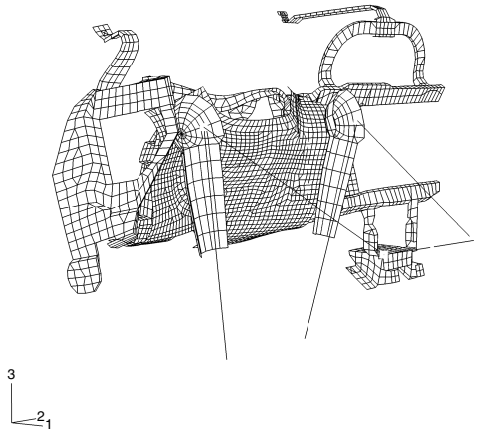


Figure 2.1.9–3 Deformed shape after 100 ms.

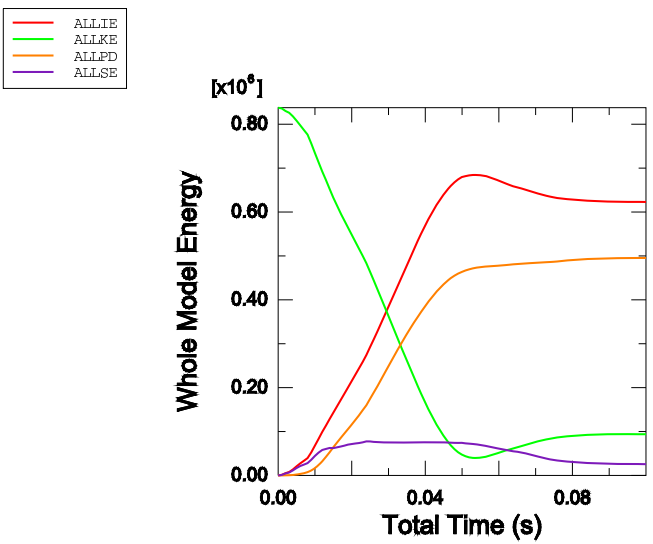


Figure 2.1.9–4 Time histories of the whole model: internal energy, kinetic energy, recoverable strain energy, and plastic dissipation.

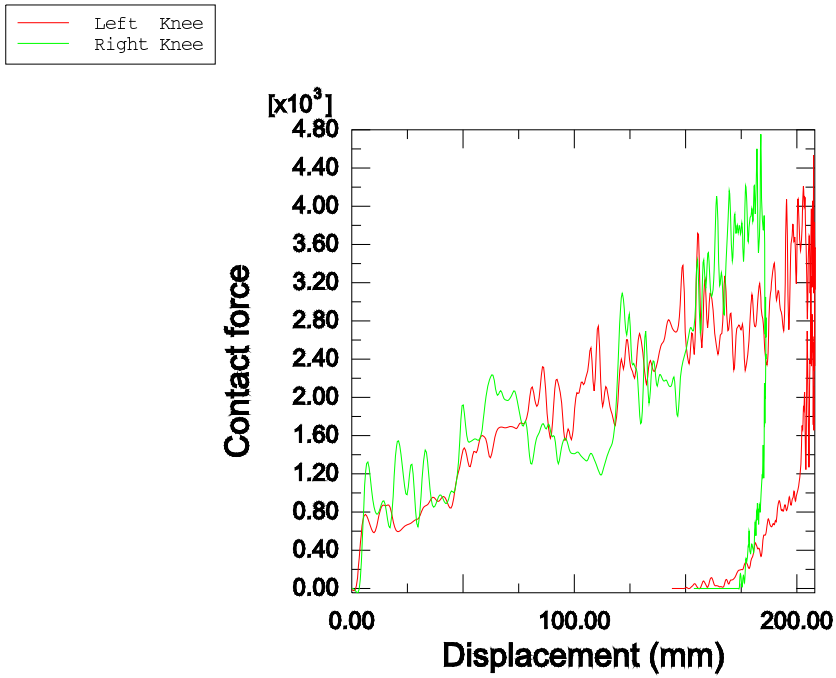


Figure 2.1.9–5 Front leg reaction forces measured against impact displacement.

2.1.10 CRIMP FORMING WITH GENERAL CONTACT

Product: Abaqus/Explicit

This example illustrates the use of the general contact capability in a simulation involving a large number of contacting surfaces. The general contact algorithm allows very simple definitions of contact with very few restrictions on the types of surfaces involved (see "Defining general contact interactions in Abaqus/Explicit," Section 36.4.1 of the Abaqus Analysis User's Guide).

Geometry and model

This model simulates crimp forming. Modern automobiles contain several thousand crimp joints. In a crimp joint a multi-strand wire bundle is mechanically joined to an end terminal to provide electrical continuity across the joint. The portion of the terminal that is folded over and into the wire bundle during crimping is called the grip. Proper design of a crimp joint depends on a number of competing factors including the diameter and number of the wire strands; the thickness, length, and material of the grip; and the geometry and surface finish of the crimp tooling. Out-of-plane extrusion of the wire bundle and grip during crimping is a significant factor in crimp formation.

In this example the grip is 0.25 mm thick and has a 50% coin at the tips. Coining is done to help the grip arms curl over the wire bundle as they are pushed against the roof of the punch during crimping. The grip arm tips are 3.28 mm apart initially (wing tip width). A nineteen-strand wire bundle is used, with each strand having a 0.28 mm diameter. Figure 2.1.10–1 shows the model geometry prior to crimp forming. Figure 2.1.10–2 shows a close-up view of the wire-grip assembly.

The deformable wires and the grip are modeled with C3D8R elements. The punch and the anvil are modeled as rigid parts using R3D4 elements. The grip is made of a half hard copper alloy that is modeled as a von Mises elastic strain hardening plastic material with a Young's modulus of 112 GPa, a Poisson's ratio of 0.34, and a yield stress of 391 MPa. The wires are made of copper that is modeled as a strain hardening plastic material with a Young's modulus of 117 GPa, a Poisson's ratio of 0.35, and a yield stress of 241.5 MPa.

Analysis definition

An explicit dynamic simulation is used because the following aspects would present difficulties for a static analysis with Abaqus/Standard:

- The model has no static stability due to the free rigid body motion of the grip and wires.
- During crimping the grip arms buckle as they are turned by the punch downward into the bundle.
- There is complex multi-body contact in the analysis: between the grip arms and the nineteen wires, between each combination of two wires, and between the two grip arms.

The rigid punch has to undergo a downward stroke of 6.88 mm to complete the crimp forming. The punch is moved downward at varying speeds to conduct the analysis efficiently without having inertia effects significantly influence the solution. Initially the punch is moved at an average speed of 50 mm/sec

to establish contact between the grip arms and the rigid punch. Then the punch is moved at 300 mm/sec until the tip of the grip arms reach the roof of the punch. In the final phase the punch is slowed to about 20 mm/sec while the grip arms buckle and fold over into the wire bundle. The overall analysis time is about 0.12 seconds.

The general contact algorithm in Abaqus/Explicit is used for this analysis. The general contact inclusions option to automatically define an all-inclusive surface is used and is the simplest way to define contact in the model. Since this surface spans all the bodies in the model, self-contact for this surface includes interactions between all of the bodies. The contact pair algorithm cannot use surfaces that span multiple bodies; thus, using the contact pair approach for this model would be very tedious. Since there are 22 contacting parts in the model, 231 contact pairs would need to be defined to account for all the possible two-surface combinations, in addition to one contact pair needed to model self-contact for the grip.

Geometric feature edges of a model can also be considered for edge-to-edge contact by the general contact algorithm if a cutoff feature angle is specified. The feature angle is the angle formed between the normals of the two facets connected to an edge. Most of the interactions in this analysis can be detected by node-to-facet contact and, thus, do not rely on edge-to-edge contact; however, when the grip arms extrude out of the punch and contact the edge of the rigid punch, edge-to-edge contact is necessary to enforce contact accurately. The feature angle criterion of the surface property is used to specify a cutoff feature angle of 20° for this analysis; thus, all edges with feature angles greater than 20° are included in the general contact domain.

Coulomb friction is assumed between the individual wires, between the grip and anvil, between the punch and the grip, and between the two grip arms. The general contact property assignment is used to assign the appropriate friction coefficients to the various types of pairings.

The anvil is held motionless during the analysis. One end of the wire bundle is fully constrained, and the other end has no boundary conditions.

Results and discussion

Figure 2.1.10–3 shows the deformed shape of the crimp assembly after the grip arms have reached the roof of the punch (39 milliseconds). Figure 2.1.10–4 shows the deformed shape of the crimp assembly after the grip arms have curved around the roof of the punch and partially folded over into the wire bundle (76 milliseconds). The grip arms buckle between Figure 2.1.10–3 and Figure 2.1.10–4. Figure 2.1.10–5 shows a cross-sectional view of the grip and wire bundle taken at the midlength of the grip. This figure shows the distortion of the wires after 107 milliseconds, when the punch has made a downward stroke of 6.605 mm. Figure 2.1.10–6 shows the final deformed shape of the model (the rigid punch has been removed from the view for clarity). The grip arms have fully folded over into the wire bundle, and the punch has made its complete downward stroke. This figure also shows the out-of-plane extrusion of the wire bundle after distortion.

Figure 2.1.10–7 shows the final shape of the wire bundle without the surrounding grip. This figure shows that the originally round wires have been distorted during crimp formation. This distortion is essential for the correct formation of the crimp joint. The bare copper wires are actually covered by a thin layer of brittle copper oxide that forms on exposure of the copper to air. The goal of crimp forming

is to break this oxide layer and expose the copper to the surface of the grip by inducing significant surface strains in each wire.

Input files

crimp_gcont.inp	Input data for this analysis.
crimp_assembly.inp	External file referenced by this analysis.

References

- Villeneuve, G., D. Kulkarni, P. Bastnagel, and D. Berry, "Dynamic Finite Element Analysis Simulation of the Terminal Crimping Process," 42nd IEEE Holm Conference, Chicago, IL, October 1996.
- Villeneuve, G., P. Bastnagel, D. Berry, and C. S. Nagaraj, "Determining the Factors Affecting Crimp Formation Using Dynamic Finite Element Analysis," 30th IICIT Connector and Interconnection Symposium, Anaheim, CA, September 1997.
- Berry, D. T., "Development of a Crimp Forming Simulator," ABAQUS User's Conference Proceedings, pp. 125–137, 1998.

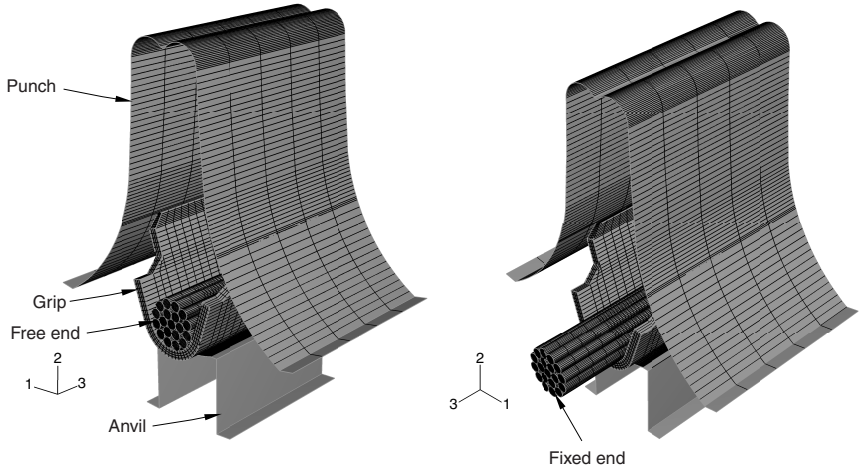


Figure 2.1.10–1 Initial configuration of the crimp forming model (opposite isometric views).

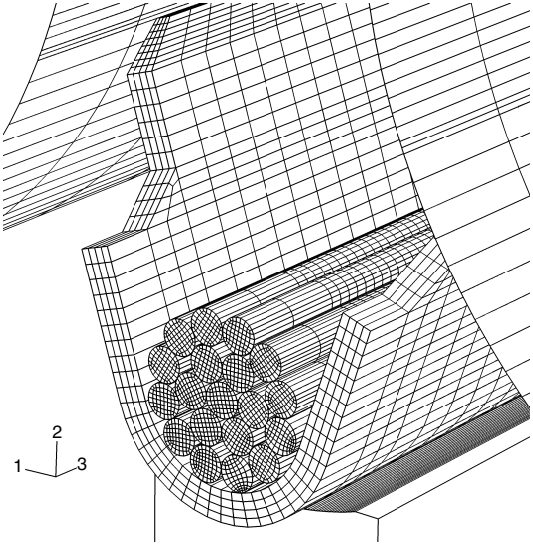


Figure 2.1.10–2 Close-up view of the wire-grip assembly.

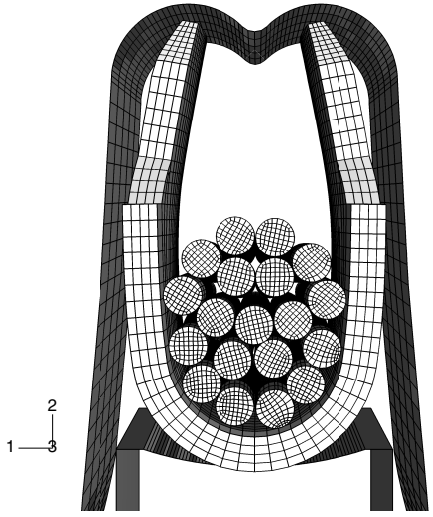


Figure 2.1.10–3 Deformed shape after 39 ms (front view).

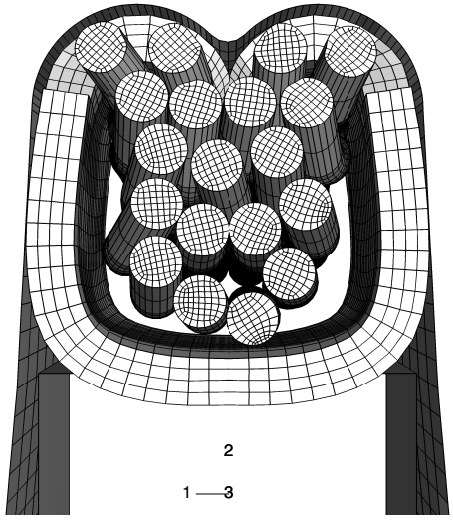


Figure 2.1.10–4 Deformed shape after 76 ms (front view).

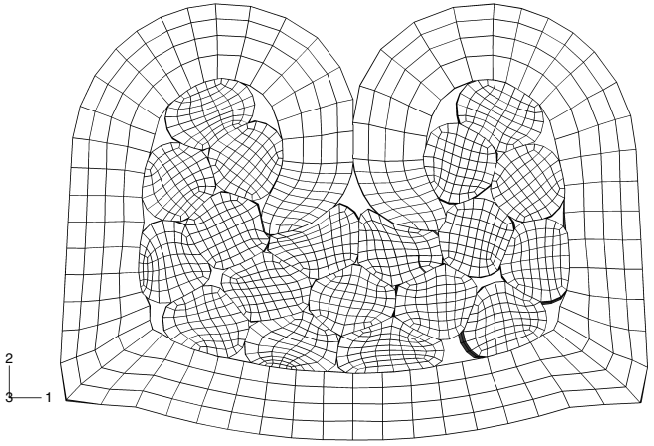


Figure 2.1.10–5 Cross-sectional view of the grip and wire bundle after 107 ms (the punch and anvil are not shown).

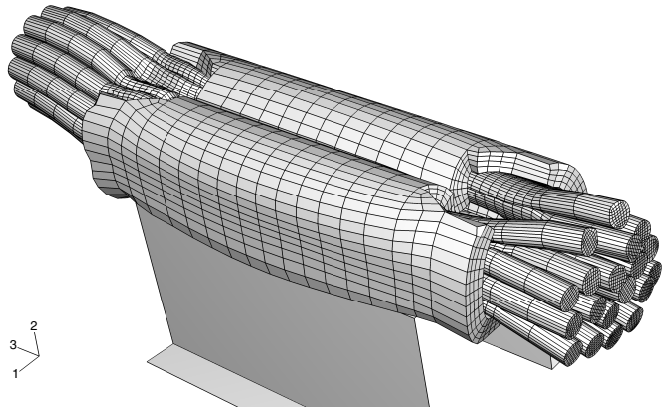


Figure 2.1.10–6 Isometric view of the final deformed configuration for the wire-grip assembly (the rigid punch is not shown).

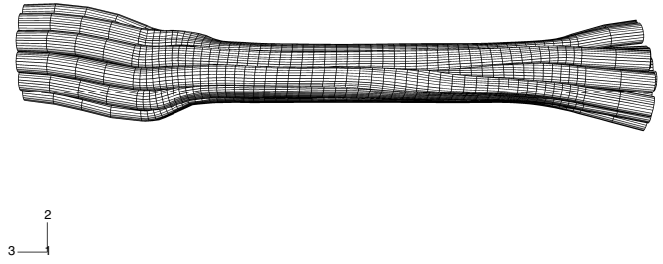


Figure 2.1.10–7 Final deformed shape of the wire bundle (the grip is not shown).

2.1.11 COLLAPSE OF A STACK OF BLOCKS WITH GENERAL CONTACT

Product: Abaqus/Explicit

This example illustrates the use of the general contact capability in a simulation involving a large number of contacting bodies. The general contact algorithm allows very simple definitions of contact with very few restrictions on the types of surfaces involved (see "Defining general contact interactions in Abaqus/Explicit," Section 36.4.1 of the Abaqus Analysis User's Guide).

Problem description

The model simulates the collapse of a stack of blocks. The undeformed configuration of the model is shown in Figure 2.1.11–1. There are 35 blocks, and each block is $12.7 \times 12.7 \times 76.2$ mm ($0.5 \times 0.5 \times 3$ inches) in size. The blocks are stacked on a rigid floor. The stack is subjected to gravity loading. It is assumed that a key block near the bottom of the stack has been removed just before the start of the analysis, initiating the collapse.

Each block is modeled with a single C3D8R element. The use of a coarse mesh highlights the edge-to-edge contact capability of the general contact algorithm, because the majority of the block-to-block interactions do not result in penetrations of nodes into faces.

Two different cases are analyzed. In the first analysis the blocks are rigid. In the second analysis the blocks are deformable. In the latter case, the material of the block is assumed to be linear elastic with a Young's modulus of 12.135 GPa (1.76×10^6 Psi), a Poisson's ratio of 0.3, and a density of 577.098 kg/m^3 (5.4×10^{-5} lb s^2/in^4). Only the density is relevant for the analysis assuming rigid blocks. In addition, ENHANCED hourglass control is used for the deformable analysis. The rigid floor is modeled as a discrete rigid surface using a single R3D4 element.

This model involves a large number of contacting bodies. The general contact capability greatly simplifies the contact definition, since each of the 595 possible block-to-block pairings does not need to be specified individually. The general contact inclusions option to automatically define an all-inclusive surface is used and is the simplest way to define contact in the model. Coulomb friction with a friction coefficient of 0.15 is assumed between the individual blocks and between the blocks and floor. The general contact property assignment is used to assign this nondefault contact property.

By default, the general contact algorithm in Abaqus/Explicit accounts for edge-to-edge contact of perimeter edges on structural elements. Geometric feature edges of a model can also be considered for edge-to-edge contact by the general contact algorithm; including the geometric feature edges is crucial in this analysis. A cutoff feature angle of 20° is specified for the feature angle criterion of the surface property to indicate that all edges with feature angles greater than 20° should be considered for edge-to-edge contact. The feature angle is the angle formed between the normals of the two facets connected to an edge.

The magnitude of the gravity loading is increased by a factor of 10 to facilitate demonstration of the edge-to-edge contact capability with a short analysis time. The analysis is performed for a period of 0.15 seconds. For the analysis with rigid blocks there is no deformable element available in the model to control the stable time increment. A fixed time increment of 1×10^{-6} seconds is specified for this

purpose, which is similar to the time increment used by the analysis with deformable blocks. The time increment chosen for the analysis with rigid blocks will affect the penalty stiffness used by the contact algorithm since the penalty stiffness is inversely proportional to the time increment squared.

Results and discussion

Results are shown for the rigid body case. Results for the deformable case are very similar to the rigid model results.

Figure 2.1.11–2 shows the displaced shape of the block assembly after 0.0375 seconds. The stack of blocks has started to collapse under gravity loading. Figure 2.1.11–3 shows a close-up view of the collapsing blocks after 0.1125 seconds. This figure clearly shows that the geometric feature edges of individual blocks contact each other during collapse. Figure 2.1.11–4 shows the final configuration of the blocks. The stack has collapsed completely on the rigid surface.

Input files

blocks_rigid_gcont.inp	Input file for the rigid body analysis.
blocks_rigid_assembly.inp	External file referenced by the rigid body analysis.
blocks_deform_gcont.inp	Input file for the deformable analysis.
blocks_deform_assembly.inp	External file referenced by the deformable analysis.

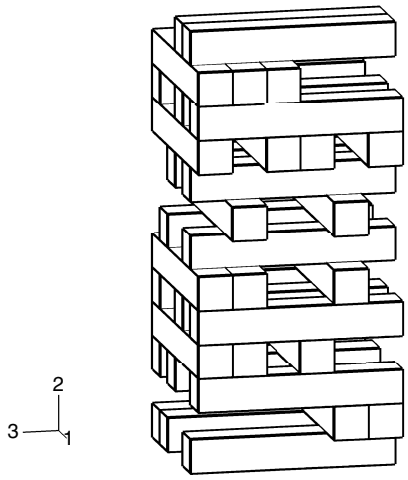


Figure 2.1.11–1 Initial configuration of the stack of blocks.

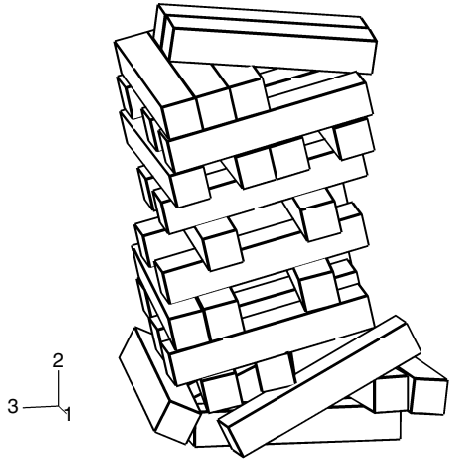


Figure 2.1.11–2 Displaced shape after 0.0375 s.

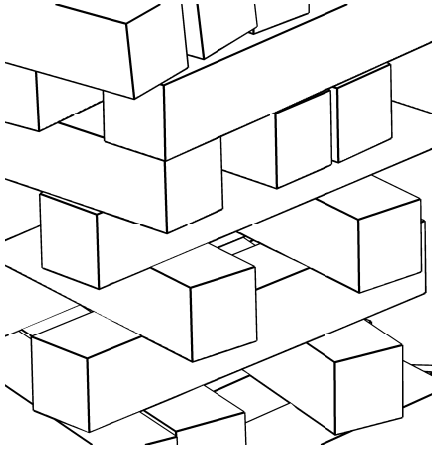


Figure 2.1.11–3 Close-up view of the collapsing blocks after 0.1125 s.

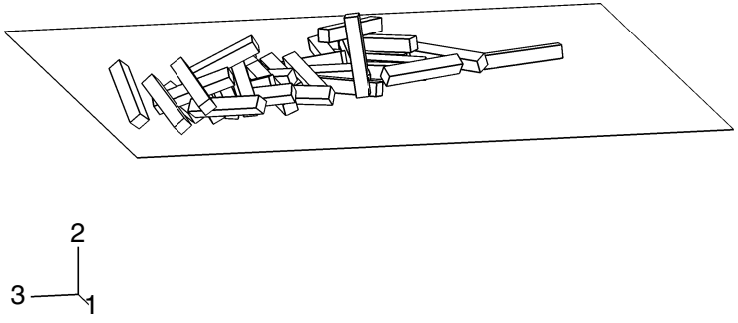


Figure 2.1.11–4 Final configuration of the model.

2.1.12 CASK DROP WITH FOAM IMPACT LIMITER

Product: Abaqus/Explicit

A containment cask is partially filled with fluid and a foam impact limiter. The complete package is dropped a distance of 9.09 m (30 ft) onto a rigid surface, which results in an impact speed of 13.35 m/sec (525.3 in/sec). The problem illustrates the use of an initial velocity condition and the analysis of a structure containing liquid and incorporating crushable foam to absorb impact energy. Experimental and numerical results for this problem have been reported by Sauvé et al. (1993). The numerical results given in the reference were obtained using a relatively coarse finite element mesh. In this example results are presented for the same coarse mesh as used in the reference and also for a more refined mesh. Both continuum meshes and particle methods are illustrated.

Model description

The containment cask shown in Figure 2.1.12–2 consists of two compartments. The upper compartment surrounds the fluid and is made of stainless steel (304L). It has a height of 580 mm (22.8 in), a diameter of 300 mm (11.8 in), and a wall thickness of 4.76 mm (0.187 in). The top mild steel cover has a thickness of 9.52 mm (0.375 in). The water is filled to a depth of 522 mm (20.55 in), which is 90% of the container's capacity. Figure 2.1.12–3 shows the original, coarse mesh of C3D8R elements used to model the fluid. Contact conditions are defined between the fluid and the inside of the upper compartment.

An impact limiter made of polyurethane foam is contained within the bottom mild steel compartment of the cask. The height of the foam impact limiter is 127.3 mm (5.01 in). Figure 2.1.12–4 shows the coarse mesh used to model the foam. Contact conditions are defined between the foam and the inside of the bottom compartment of the cask. The foam impact limiter and the fluid/stainless steel liner are separated by a mild steel bulkhead with a thickness of 12.7 mm (0.5 in). A 12.7 mm (0.5 in) air gap exists between the top of the foam surface and this bulkhead.

In the experiment a pressure transducer is located in the polyurethane foam on the centerline of the cask at the top of the impact limiter. This result is compared with vertical stress-time histories taken from the element at the top of the foam model on the centerline.

Both axisymmetric and three-dimensional models are analyzed. Figure 2.1.12–5 shows the three-dimensional model formed by assembling the parts shown in Figure 2.1.12–2 through Figure 2.1.12–4. The equivalent axisymmetric model is shown in Figure 2.1.12–6. The corresponding smoothed particle hydrodynamic model incorporates PC3D elements for both the water and the foam parts of the model.

Contact pairs are defined between the solids and the shells. Element-based surfaces are defined on the shells, and node-based surfaces are defined containing the nodes on the outer surfaces of the solid or particle elements. Input files that use the alternative general contact algorithm are also provided. The shell thickness was not taken into account when the original meshes were designed, and the outer surface of the solids usually coincides with the midsurface of the enclosing shell. This would lead to an initial overclosure of one-half the shell thickness, but contact at the midsurface of the shell is enforced, as if the shell had zero thickness. The use of a node-based surface implies a pure master-slave relationship for

the contact pair. This is important in this problem because the default in Abaqus/Explicit when contact is defined between shells and solids is to define a pure master-slave relationship with the solids as the master and the shells as the slave. In this case the shell structures are much stiffer than the fluid and foam structures, so the master-slave roles must be reversed.

For the axisymmetric model two cases using different section controls for the foam and fluid elements are analyzed. The first case uses a linear combination of stiffness and viscous hourglass control; the second case uses the default section controls (the integral viscoelastic form for hourglass control). The three-dimensional model also has two cases with different section controls for the foam and fluid elements. The first case uses the orthogonal kinematic formulation and combined (viscous-stiffness form) hourglass control; the second three-dimensional case uses the default section controls (the average strain kinematic formulation and the integral viscoelastic form for hourglass control). The section controls used are summarized in Table 2.1.12–1. Coarse and refined meshes are used for all analysis cases.

Material description

The general material properties are listed in Table 2.1.12–2. The material models for the water and foam are further described below.

Water:

The water is treated as a simple hydrodynamic material model. This provides zero shear strength and a bulk response given by

$$p = K\varepsilon_{vol},$$

where K is the bulk modulus with a value of 2068 MPa (300000 psi). This model is defined using the linear $U_s - U_p$ equation of state model provided in Abaqus/Explicit. The linear $U_s - U_p$ Hugoniot form, $p = f(\rho) + g(\rho)E_m$, is

$$p = \frac{\rho_0 c_0^2 \eta}{(1 - s\eta)^2}\left(1 - \frac{\Gamma_0 \eta}{2}\right) + \Gamma_0 \rho_0 E_m,$$

where $\eta = 1 - \rho_0/\rho$ is the same as the nominal volumetric strain measure, ε_{vol}. Since $K = \rho_0 c_0^2$, setting the parameters $s = 0.0$ and $\Gamma_0 = 0.0$ gives the simple hydrostatic bulk response defined earlier. In this analysis $c_0 = 1450.6$ m/sec (57100 in/sec) and $\rho_0 = 983.2$ kg/m^3 (0.92×10^{-4} lb sec^{2}in^{-4}). The tension cutoff pressure is assumed to be zero and is specified using a tensile failure model. Refer to "Equation of state," Section 25.2.1 of the Abaqus Analysis User's Guide, for a description of this material model.

Foam:

The crushable foam model is used for the polyurethane foam. In this model the flow potential, h, is chosen as

$$h = \sqrt{\frac{9}{2}p^2 + q^2},$$

where q is the Mises equivalent stress and p is the hydrostatic pressure. The yield surface is defined as

$$\left[\left(\frac{p_t - p_c}{2} + p\right)^2 + \left(\frac{q}{M}\right)^2\right]^{\frac{1}{2}} = \frac{p_c + p_t}{2}.$$

Sauvé et al. use the "soils and crushable foams" model, which was originally defined in an unpublished report by Krieg (1978) and is based upon a Mises plasticity model in which the yield stress depends upon the mean volumetric pressure. The volumetric deformation allows for plastic behavior, defined by tabular data defining pressure versus volume strain. This model is easy to implement in an explicit dynamics algorithm and useful because the deviatoric and volumetric terms are only loosely coupled. However, it requires an experienced analyst to ensure that meaningful results are obtained, mainly because the model does not match physical behavior well under deviatoric straining.

To define the initial shape of the yield surface, the Abaqus/Explicit crushable foam model with volumetric hardening requires the initial yield stress in uniaxial compression, σ_0; the magnitude of the strength in hydrostatic tension, p_t; and the initial yield stress in hydrostatic compression, p_{c0}. Sauvé et al. define the pressure-dependent yield surface for the foam model as

$$\sigma_y^2 = 3.18 + 2.06p,$$

where the units of stress are MPa and pressure is positive in compression. To calibrate the Abaqus/Explicit crushable foam model to this pressure-dependent data, we observe that $p = \frac{1}{3}\sigma_y$ for the uniaxial compression case. Substituting this value for p in the above equation and solving for σ_y gives σ_0 = 2.16 MPa (313.3 psi). The value of p_t is obtained by solving the above equation for $\sigma_y = 0$, giving p_t = 1.54 MPa (223.8 psi). The value of p_{c0} is given in the reference as p_{c0} = 5.52 MPa (800.0 psi).

The pressure-volumetric strain data in the reference are given in Table 2.1.12–3. Table 2.1.12–4 shows the uniaxial stress-plastic strain data converted to the form required for the Abaqus/Explicit volumetric hardening model. Each form of the data is plotted in Figure 2.1.12–1.

Results and discussion

The deformed geometries for the three-dimensional and the axisymmetric models at 5 msec are shown in Figure 2.1.12–7 and Figure 2.1.12–8. The axisymmetric model is analyzed using combined hourglass control. The three-dimensional model uses orthogonal kinematic and combined hourglass control. Figure 2.1.12–9 shows plots of the vertical stress versus time for the element located at the pressure transducer in the foam; results from the models with the previous section control options, as well as results from analyses using the default section control options, are reported for comparison

(see Table 2.1.12–1). Axisymmetric and three-dimensional results are compared to the experimental pressure trace. The time origin of the experimental curve is not defined in the reference; therefore, the experimental curve is shifted so that the time when pressure in the transducer changes to a positive value is assumed to be the time at which impact occurs. The numerical pressure results show significant oscillations about the experimental results during the first 2 msec of the response. This is partly because the meshes are quite coarse and partly because pressure transducers in experiments exhibit inertia in their response and will not report sharp gradients in time. During the next 3 msec the numerical results correspond more closely with the experimental results. The analyses run with different section control options compare very well.

A more refined three-dimensional mesh is shown in Figure 2.1.12–10. The refined axisymmetric model is the same model used in the r–z plane. The deformed geometries for these models are shown in Figure 2.1.12–11 (using orthogonal kinematic and combined hourglass control) and Figure 2.1.12–12 (using combined hourglass control). The vertical stress histories for the refined models are shown in Figure 2.1.12–13 for the same section controls used for the coarse meshes (see Table 2.1.12–1). The numerical results show less oscillation about the experimental results than those obtained with the coarse mesh. They compare well with the experimental results during the following 3 msec of the response. In addition, Figure 2.1.12–11 shows that the refined mesh eliminates much of the fluid's hourglass-like response due to its zero shear strength. Figure 2.1.12–14 shows a comparison of the vertical stress results for the three-dimensional refined model and the smoothed particle hydrodynamic model.

Input files

cask_drop_axi_cs.inp	Coarse axisymmetric model using COMBINED hourglass control.
cask_drop_3d_ocs.inp	Coarse three-dimensional model using ORTHOGONAL kinematic and COMBINED hourglass control.
cask_drop_3d_ocs_gcont.inp	Coarse three-dimensional model using ORTHOGONAL kinematic and COMBINED hourglass control and the general contact capability.
cask_drop_axi.inp	Coarse axisymmetric mesh using the default section controls.
cask_drop_3d.inp	Coarse three-dimensional mesh using the default section controls.
cask_drop_3d_gcont.inp	Coarse three-dimensional mesh using the default section controls and the general contact capability.
cask_drop_axi_r_cs.inp	Refined axisymmetric model using COMBINED hourglass control.
cask_drop_3d_r_ocs.inp	Refined three-dimensional model using ORTHOGONAL kinematic and COMBINED hourglass control.
cask_drop_3d_r_ocs_gcont.inp	Refined three-dimensional model using ORTHOGONAL kinematic and COMBINED hourglass control and the general contact capability.

cask_drop_axi_r.inp	Refined axisymmetric mesh using the default section controls.
cask_drop_3d_r.inp	Refined three-dimensional mesh using the default section controls.
cask_drop_3d_r_gcont.inp	Refined three-dimensional mesh using the default section controls and the general contact capability.
cask_drop_3d_sph.inp	Three-dimensional model using the smoothed particle hydrodynamic method.

References

- Krieg, R. D., "A Simple Constitutive Description for Soils and Crushable Foams," SC-DR-72-0883, Sandia National Laboratories, Albuquerque, NM, 1978.
- Sauvé, R. G., G. D. Morandin, and E. Nadeau, "Impact Simulation of Liquid-Filled Containers Including Fluid-Structure Interaction," Journal of Pressure Vessel Technology, vol. 115, pp. 68–79, 1993.

Table 2.1.12–1 Analysis section controls tested.

Analysis Label	Section Controls	
	Kinematic Formulation	Hourglass Control
AXI	n/a	integral viscoelastic
AXI CS	n/a	combined
3D	average strain	integral viscoelastic
3D OCS	orthogonal	combined

Table 2.1.12–2 Material properties.

Properties	A36	304L	Liquid	Foam
Density, ρ (kg/m^3)	8032	8032	983	305
Young's modulus, E (GPa)	193.1	193.1		.129
Poisson's ratio, ν	0.28	0.28		0
Yield stress, σ_{y0} (MPa)	206.8	305.4		
Bulk modulus, K (GPa)			2.07	
Hardening modulus, E_t (GPa)	0	1.52		

Table 2.1.12–3 Pressure-volumetric strain data.

ϵ_v	0	0.01	0.02	0.03	0.04	0.05	0.06	0.385	0.48	0.53	0.55
p (MPa)	0	2.76	4.14	5.17	5.52	5.86	6.21	10.34	19.31	39.30	82.74

Table 2.1.12–4 Uniaxial stress-plastic strain data.

ϵ^p	0.00	0.01	0.02	0.345	0.44	0.49	0.51	2.00
σ_c (MPa)	2.16	2.24	2.33	3.23	4.91	8.20	14.67	758.89

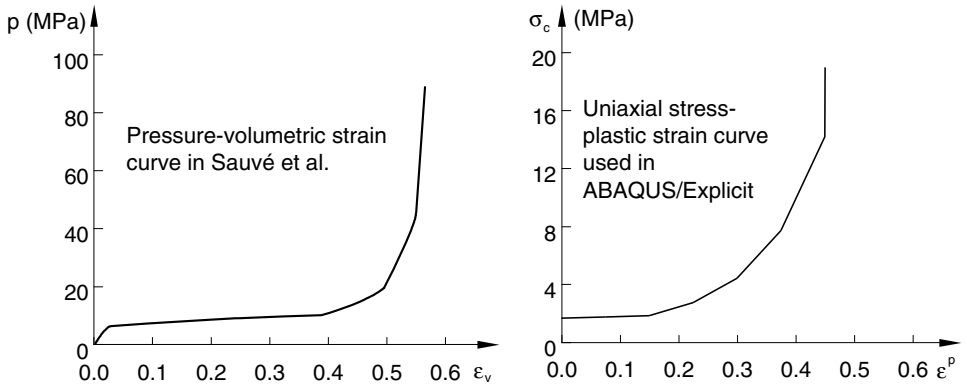


Figure 2.1.12–1 Foam hardening curves.

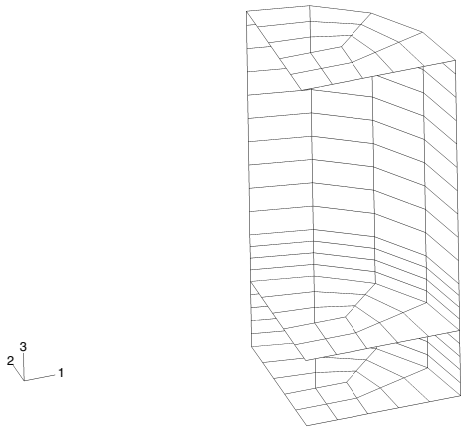


Figure 2.1.12–2 Containment structure mesh in the three-dimensional model (coarse mesh).

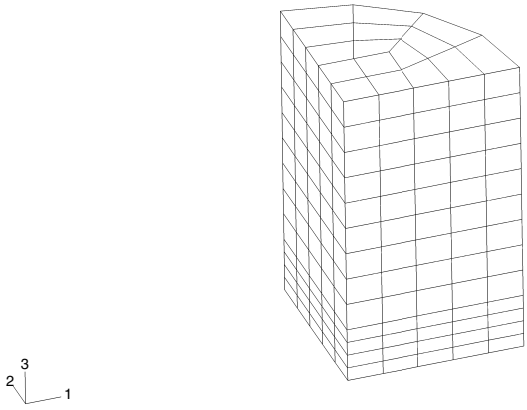


Figure 2.1.12–3 Fluid mesh in the three-dimensional model (coarse mesh).

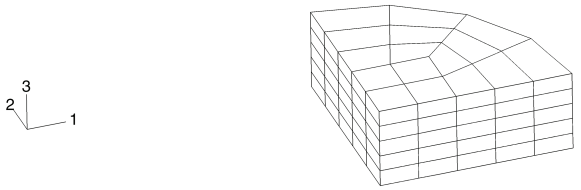


Figure 2.1.12–4 Foam mesh in the three-dimensional model (coarse mesh).

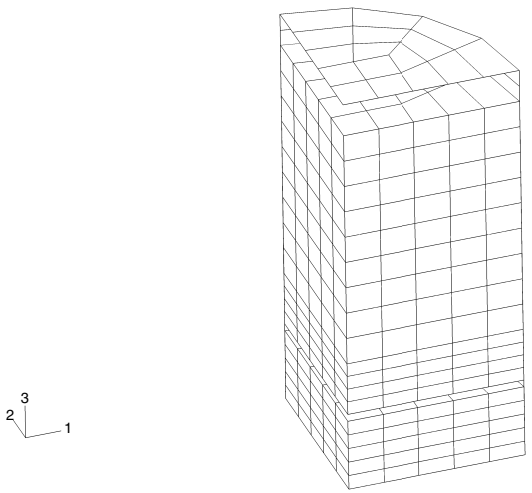


Figure 2.1.12–5 The complete three-dimensional model (coarse mesh).

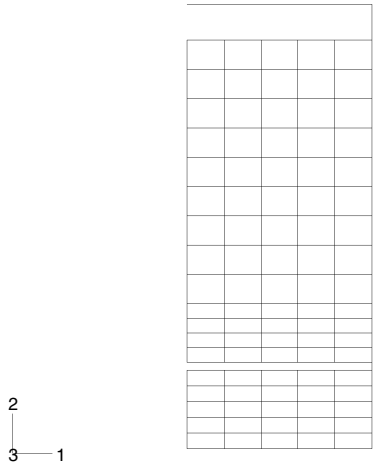


Figure 2.1.12–6 Axisymmetric model (coarse mesh).

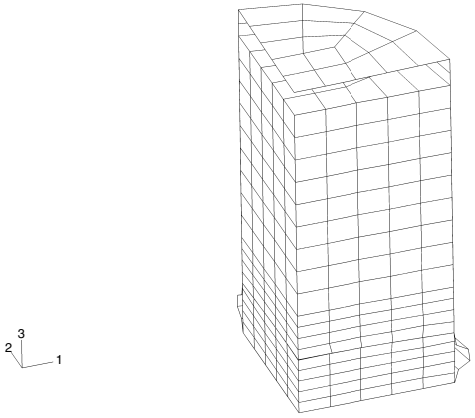


Figure 2.1.12–7 Three-dimensional deformed geometry using orthogonal element kinematics and combined hourglass control (coarse mesh).

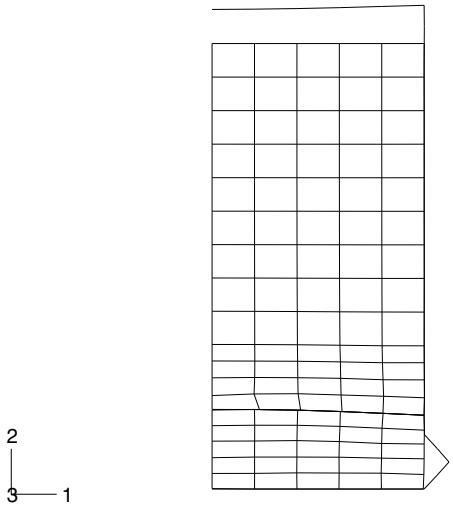


Figure 2.1.12–8 Axisymmetric deformed geometry using combined hourglass control (coarse mesh).

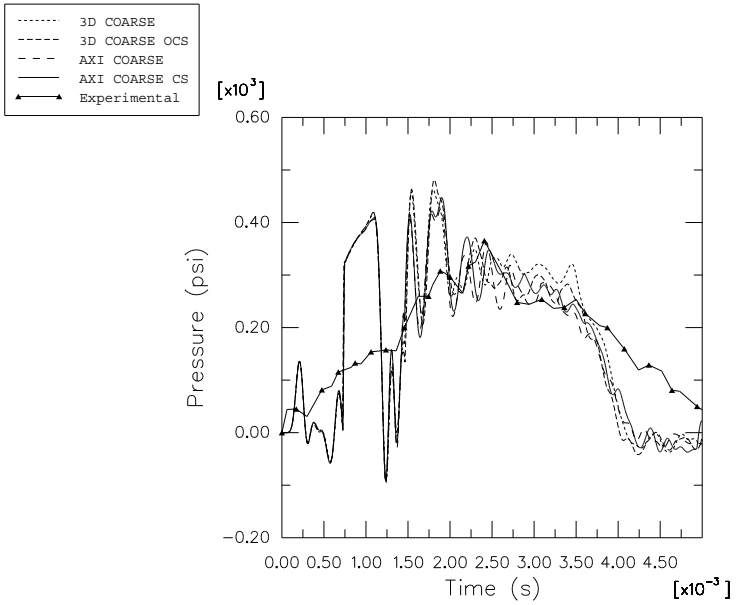


Figure 2.1.12–9 Vertical stress history in the foam (coarse mesh).

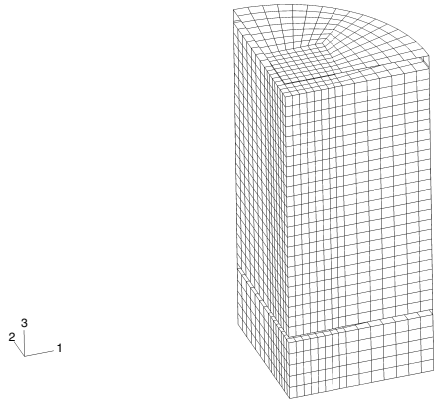


Figure 2.1.12–10 Refined mesh for the three-dimensional model.

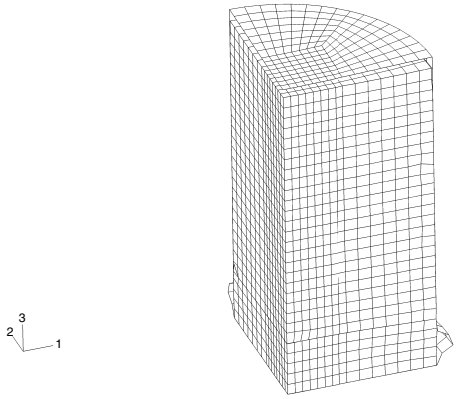


Figure 2.1.12–11 Three-dimensional deformed geometry using orthogonal element kinematics and combined hourglass control (refined mesh).

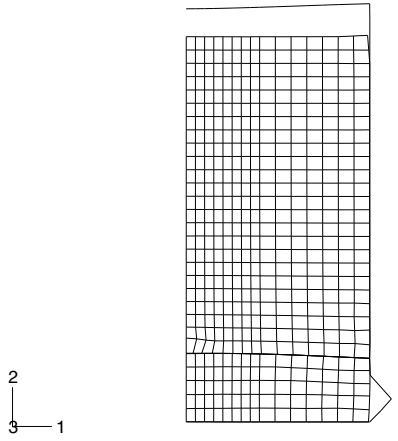


Figure 2.1.12–12 Axisymmetric deformed geometry using combined hourglass control (refined mesh).

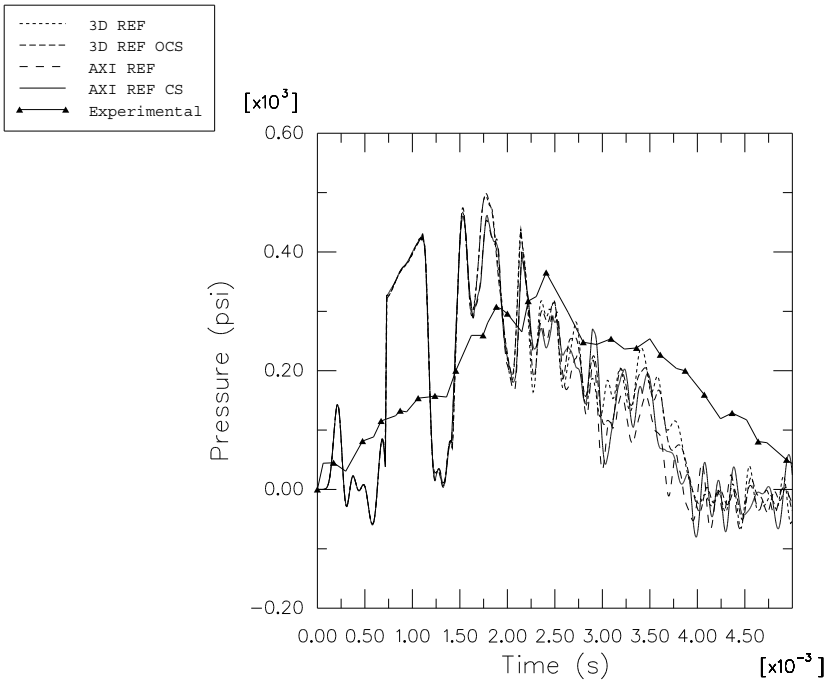


Figure 2.1.12–13 Vertical stress history in the foam (refined mesh).

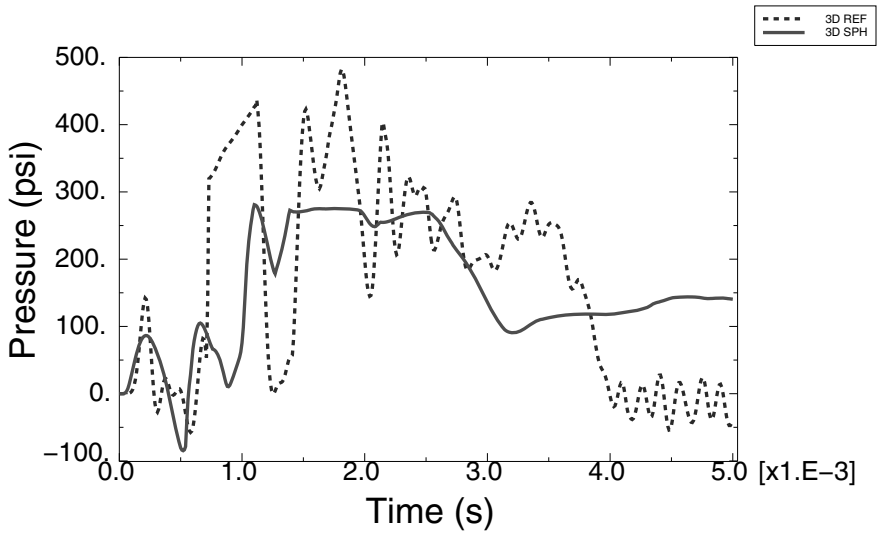


Figure 2.1.12–14 Vertical stress history comparison for the three-dimensional refined model and the smoothed particle hydrodynamic model.

2.1.13 OBLIQUE IMPACT OF A COPPER ROD

Product: Abaqus/Explicit

This example simulates a high velocity, oblique impact of a copper rod into a rigid wall. Extremely high plastic strains develop at the crushed end of the rod, resulting in severe local mesh distortion. Adaptive meshing is used to reduce element distortion and to obtain an accurate and economical solution to the problem.

Problem description

The model geometry is depicted in Figure 2.1.13–1. A cylindrical rod, measuring 32.4 × 3.2 mm, impacts a rigid wall with an initial velocity of v_y=340 m/sec. The wall is perpendicular to the *x*–*z* plane and makes an angle of 30° with the *x*–*y* plane. The half-symmetric finite element model is shown in Figure 2.1.13–2. Symmetry boundary conditions are applied at the *y*=0 plane. The rod is meshed with CAX4R elements, and the wall is modeled as an analytical rigid surface using a three-dimensional cylindrical surface in conjunction with a rigid body constraint. Coulomb friction is assumed between the rod and the wall, with a friction coefficient of 0.2. The analysis is performed for a period of 120 microseconds.

The rod is modeled as a Johnson-Cook, elastic-plastic material with a Young's modulus of 124 GPa, a Poisson's ratio of 0.34, and a density of 8960 kg/m^3. The Johnson-Cook model is appropriate for modeling high-rate impacts involving metals. The Johnson-Cook material parameters are taken from Johnson and Cook (1985) in which the following constants are used: $A = 90$ MPa, $n = 0.31$, $m = 1.09$, $C = 0.025$, and $\dot{\varepsilon}_o = 1$ s^{-1}. Furthermore, the melting temperature is 1058°C, and the transition temperature is 25°C. Adiabatic conditions are assumed with a heat fraction of 50%. The specific heat of the material is 383 J/Kg°C, and the thermal expansion coefficient is 0.00005°C^{-1}.

Adaptive meshing

A single adaptive mesh domain that incorporates the entire rod is defined. Symmetry boundary conditions are defined as Lagrangian surfaces (the default), and contact surfaces are defined as sliding contact surfaces (the default). Because the impact phenomenon modeled in this example is an extremely dynamic event with large changes in geometry occurring over a relatively small number of increments, it is necessary to increase the frequency and intensity of adaptive meshing. The frequency value is reduced to 1 increment from a default value of 10, and the number of mesh sweeps used to smooth the mesh is increased to 3 from the default value of 1. The default values are used for all other adaptive mesh controls.

Results and discussion

Deformed shape plots at 40, 80, and 120 microseconds are shown in Figure 2.1.13–3, Figure 2.1.13–4, and Figure 2.1.13–5, respectively. The rod rebounds from the wall near the end of the analysis. High-speed collisions such as these result in significant amounts of material flow in the impact zone. A pure Lagrangian analysis of this finite element model fails as a result of excessive distortions. Continuous adaptive meshing allows the analysis to run to completion while retaining a high-quality mesh. The

kinetic and internal energy histories are plotted in Figure 2.1.13–6. Most of the initial kinetic energy is converted to internal energy as the rod is plastically deformed. Both energy curves plateau as the rod rebounds from the wall.

Input files

ale_rodimpac_inclined.inp	Analysis using adaptive meshing.
ale_rodimpac_inclined_nodelem.inp	External file referenced by this analysis.

Reference

- Johnson, G. R., and W. H. Cook, "Fracture Characteristics of Three Metals Subjected to Various Strains, Strain Rates, Temperatures and Pressures," Engineering Fracture Mechanics, 21, pp. 31–48, 1985.

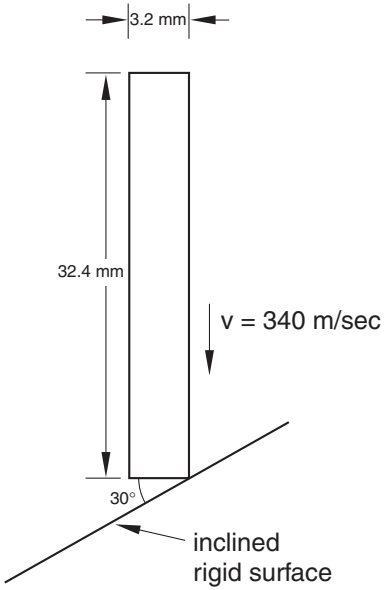


Figure 2.1.13–1 Model geometry.

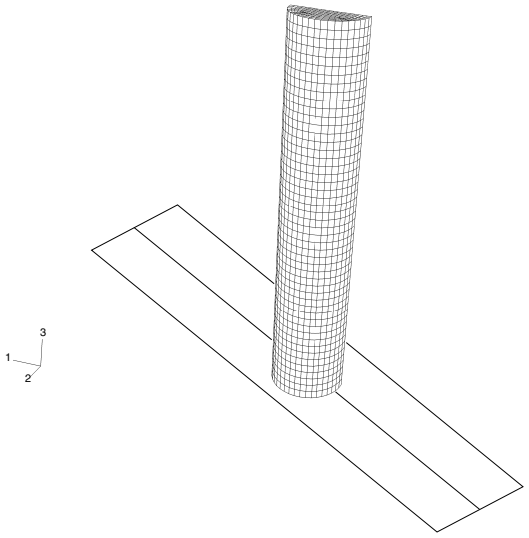


Figure 2.1.13–2 Initial configuration.

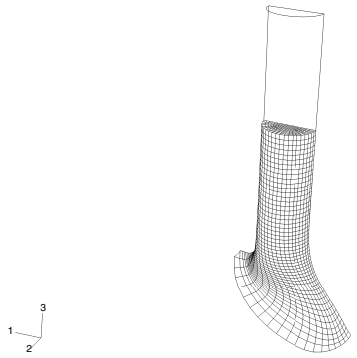


Figure 2.1.13–3 Deformed configuration at 40 microseconds.

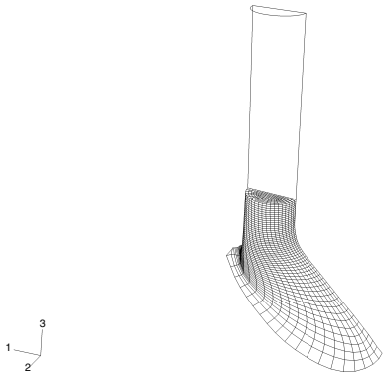


Figure 2.1.13–4 Deformed configuration at 80 microseconds.

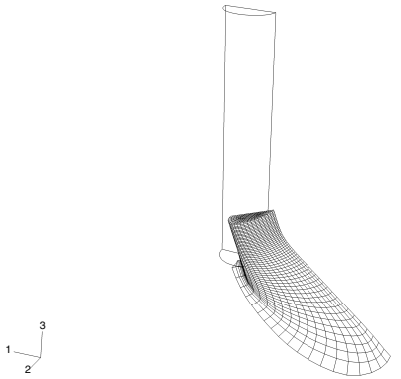


Figure 2.1.13–5 Deformed configuration at 120 microseconds.

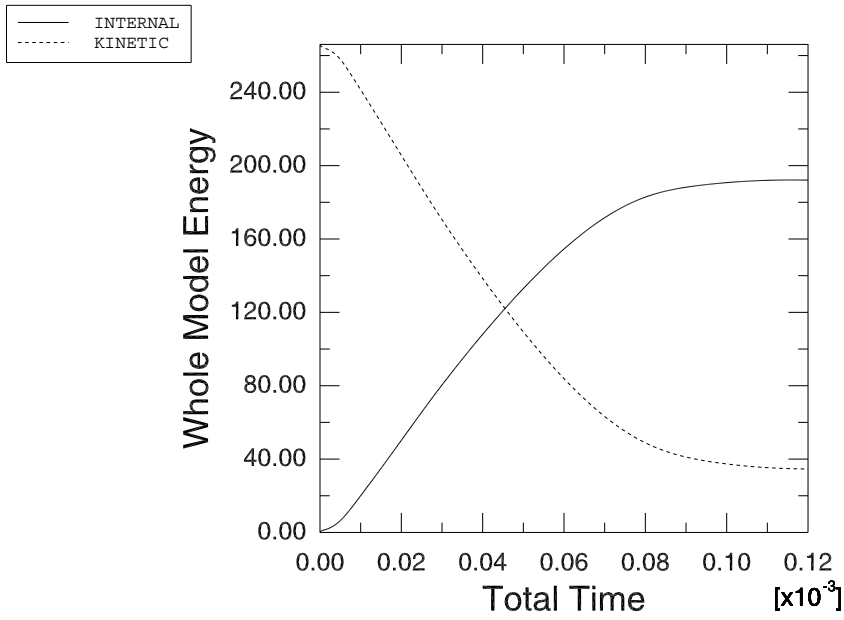


Figure 2.1.13–6 Time history of kinetic and internal energies of the rod.

2.1.14 WATER SLOSHING IN A BAFFLED TANK

Product: Abaqus/Explicit

This example illustrates the use of adaptive meshing to model an inviscid fluid sloshing inside a baffled tank. The overall structural response resulting from the coupling between the water and tank, rather than a detailed solution in the fluid, is sought. Adaptive meshing permits the investigation of this response over longer time periods than a pure Lagrangian approach would because the mesh entanglement that occurs in the latter case is prevented.

Problem description

The geometry for the problem is shown in Figure 2.1.14–1. The model consists of a baffled tank filled with water. The baffle, which is attached to the sides and top of the tank, does not penetrate the entire depth of the water. The tank measures 508 × 152.4 × 152.4 mm (20 × 6 × 6 inches), and the baffle measures 3.048 × 152.4 × 121.92 mm (0.12 × 6 × 4.8 inches). The tank is filled with 101.6 mm (4 inches) of water.

A cutaway view of the finite element model that displays the baffle and the water is shown in Figure 2.1.14–2. The top of the tank is not modeled because the water is not expected to come into contact with it. The tank is modeled as a rigid body and is meshed with R3D4 elements. The baffle is modeled as a deformable body and is meshed with S4R elements. A graded mesh of C3D8R elements is used for the water, with more refinement adjacent to the baffle where significant deformations are expected.

In sloshing problems water can be considered an incompressible and inviscid material. An effective method for modeling water in Abaqus/Explicit is to use a simple Newtonian viscous shear model and a linear $U_s - U_p$ equation of state for the bulk response. The bulk modulus functions as a penalty parameter for the incompressible constraint. Since sloshing problems are unconfined, the bulk modulus chosen can be two or three orders of magnitude less than the actual bulk modulus and the water will still behave as an incompressible medium. The shear viscosity also acts as a penalty parameter to suppress shear modes that could tangle the mesh. The shear viscosity chosen should be small because water is inviscid; a high shear viscosity will result in an overly stiff response. An appropriate value for the shear viscosity can be calculated based on the bulk modulus. To avoid an overly stiff response, the internal forces arising due to the deviatoric response of the material should be kept several orders of magnitude below the forces arising due to the volumetric response. This can be done by choosing an elastic shear modulus that is several orders of magnitude lower than the bulk modulus. If the Newtonian viscous deviatoric model is used, the shear viscosity specified should be on the order of an equivalent shear modulus, calculated as mentioned earlier, scaled by the stable time increment. The expected stable time increment can be obtained from a datacheck analysis of the model. This method is a convenient way to approximate a shear strength that will not introduce excessive viscosity in the material.

In addition, if a shear model is defined, the hourglass control forces are calculated based on the shear stiffness of the material. Thus, in materials with extremely low or zero shear strengths such as inviscid fluids, the hourglass forces calculated based on the default parameters are insufficient to prevent

spurious hourglass modes. Therefore, a sufficiently high hourglass scaling factor is used to increase the resistance to such modes. This analysis methodology is verified in "Water sloshing in a pitching tank," Section 1.12.7 of the Abaqus Benchmarks Guide.

For this example the linear $U_s - U_p$ equation of state is used with a wave speed of 45.85 m/sec (1805 in/sec) and a density of 983.204 kg/m^3 (0.92×10^{-4} lb sec^2/in^4). The wave speed corresponds to a bulk modulus of 2.07 MPa (300 psi), three orders of magnitude less than the actual bulk modulus of water, 2.07 GPa (3.0×10^5 psi). The shear viscosity is chosen as 1.5×10^{-8} psi sec. The baffle is modeled as a Mooney-Rivlin elastomeric material with hyperelastic constants C_{10} = 689480 Pa (100 psi) and C_{01} = 172370 Pa (25 psi) and a density of 10900.74 kg/m^3 (1.02×10^{-3} lb sec^2/in^4).

Pure master-slave contact is defined between the tank and the water; balanced master-slave contact is defined between the baffle and the water. The bottom edge of the baffle has nodes in common with the underlying water surface. This prevents relative slip between the bottom edge of the baffle and the water immediately below it. The motion of the other edges of the baffle coincides with that of the tank.

The water is subjected to gravity loading. Consequently, an initial geostatic stress field is defined to equilibrate the stresses caused by the self-weight of the water. A velocity pulse in the form of a sine wave with an amplitude of 63.5 mm (2.5 inches) and a period of 2 seconds is prescribed for the tank in both the *x*- and *y*-directions simultaneously. All remaining degrees of freedom for the tank are fully constrained. The sloshing analysis is performed for two seconds.

Adaptive meshing

A single adaptive mesh domain that incorporates the water is defined. Sliding boundary regions are used for all contact surface definitions on the water (the default). Because the sloshing phenomenon modeled in this example results in large mesh motions, it is necessary to increase the frequency and intensity of adaptive meshing. The frequency value is reduced to 5 increments from a default value of 10, and the number of mesh sweeps used to smooth the mesh is increased to 3 from a default value of 1. The adaptive meshing algorithm is configured in this example so that the initial mesh gradation of the water is preserved while continuous adaptive meshing is performed. The default values are used for all other parameters and controls.

Results and discussion

Figure 2.1.14–3, Figure 2.1.14–4, and Figure 2.1.14–5 show the deformed mesh configuration at $t = 1.2$ s, $t = 1.6$ s, and $t = 2.0$ s, respectively. Four time histories of the vertical displacement of the water level are shown in Figure 2.1.14–6; these correspond to the water level at the baffle in the front and back of the left and right bays. The locations at which the time histories are measured are denoted A, B, C, and D in Figure 2.1.14–1. An analysis such as this could be used to design a baffle that attenuates sloshing at certain frequencies. Using adaptivity in Abaqus/Explicit is appropriate for sloshing problems in which the structural response is of primary interest. It is generally not possible to model such flow behaviors as splashing or complex free surface interactions. Furthermore, surface tension is not modeled.

Input files

ale_water_sloshing.inp	Input data for this analysis.
ale_water_sloshingel.inp	Element data.

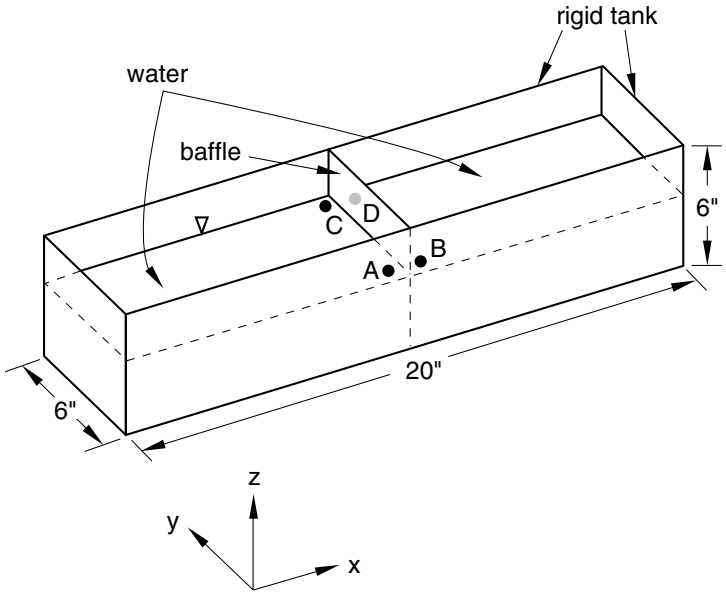


Figure 2.1.14–1 Model geometry.

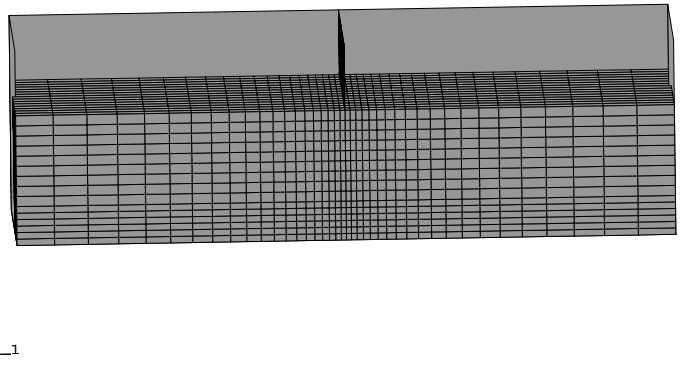


Figure 2.1.14–2 Initial configuration (front of rigid tank is not shown).

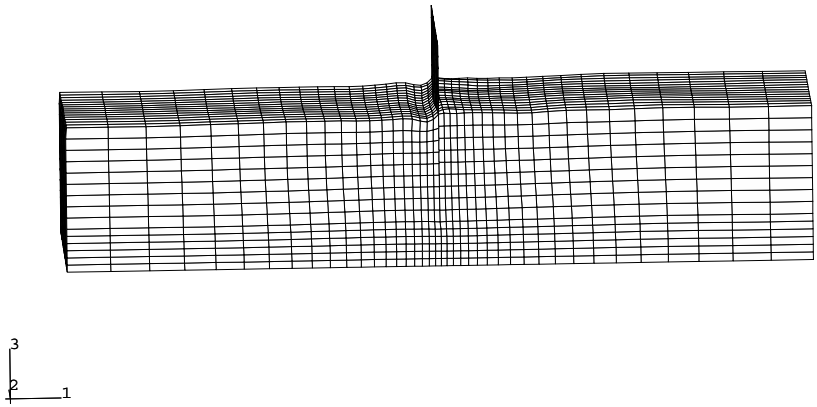


Figure 2.1.14–3 Deformed configuration of the water and the baffle at 1.2 seconds.

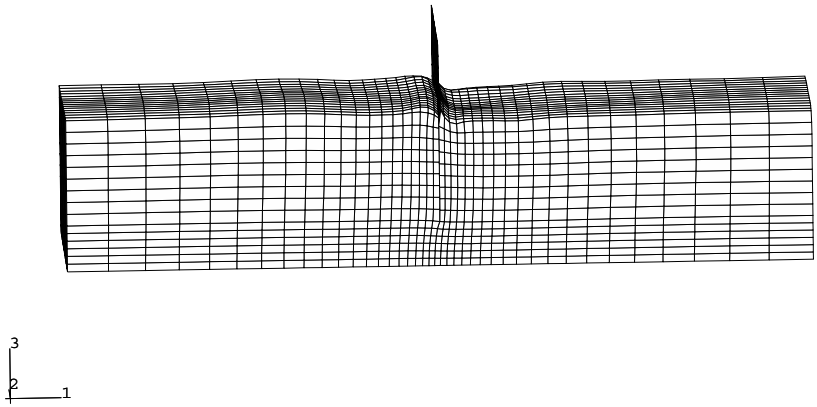


Figure 2.1.14–4 Deformed configuration of the water and the baffle at 1.6 seconds.

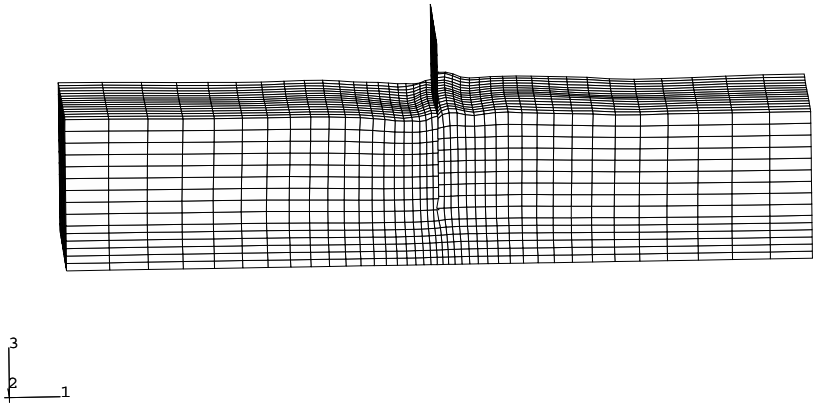


Figure 2.1.14–5 Deformed configuration of the water and the baffle at 2.0 seconds.

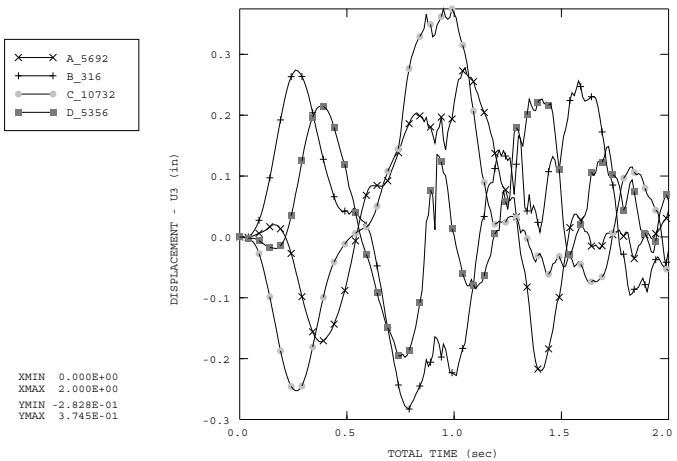


Figure 2.1.14–6 Time histories of the vertical displacement of the water at the baffle at both the front and back of the left and right bays.

2.1.15 SEISMIC ANALYSIS OF A CONCRETE GRAVITY DAM

Products: Abaqus/Standard Abaqus/Explicit

In this example we consider an analysis of the Koyna dam, which was subjected to an earthquake of magnitude 6.5 on the Richter scale on December 11, 1967. The example illustrates a typical application of the concrete damaged plasticity material model for the assessment of the structural stability and damage of concrete structures subjected to arbitrary loading. This problem is chosen because it has been extensively analyzed by a number of investigators, including Chopra and Chakrabarti (1973), Bhattacharjee and Léger (1993), Ghrib and Tinawi (1995), Cervera et al. (1996), and Lee and Fenves (1998).

Problem description

The geometry of a typical non-overflow monolith of the Koyna dam is illustrated in Figure 2.1.15–1. The monolith is 103 m high and 71 m wide at its base. The upstream wall of the monolith is assumed to be straight and vertical, which is slightly different from the real configuration. The depth of the reservoir at the time of the earthquake is h_w = 91.75 m. Following the work of other investigators, we consider a two-dimensional analysis of the non-overflow monolith assuming plane stress conditions. The finite element mesh used for the analysis is shown in Figure 2.1.15–2. It consists of 760 first-order, reduced-integration, plane stress elements (CPS4R). Nodal definitions are referred to a global rectangular coordinate system centered at the lower left corner of the dam, with the vertical y-axis pointing in the upward direction and the horizontal x-axis pointing in the downstream direction. The transverse and vertical components of the ground accelerations recorded during the Koyna earthquake are shown in Figure 2.1.15–3 (units of g = 9.81 m sec^{-2}). Prior to the earthquake excitation, the dam is subjected to gravity loading due to its self-weight and to the hydrostatic pressure of the reservoir on the upstream wall.

For the purpose of this example we neglect the dam–foundation interactions by assuming that the foundation is rigid. The dam–reservoir dynamic interactions resulting from the transverse component of ground motion can be modeled in a simple form using the Westergaard added mass technique. According to Westergaard (1933), the hydrodynamic pressures that the water exerts on the dam during an earthquake are the same as if a certain body of water moves back and forth with the dam while the remainder of the reservoir is left inactive. The added mass per unit area of the upstream wall is given in approximate form by the expression $\frac{7}{8}\rho_w\sqrt{h_w(h_w - y)}$, with $y \leq h_w$, where ρ_w = 1000 kg/m^3 is the density of water. In the Abaqus/Standard analysis the added mass approach is implemented using a simple 2-node user element that has been coded in user subroutine **UEL**. In the Abaqus/Explicit analysis the dynamic interactions between the dam and the reservoir are ignored.

The hydrodynamic pressures resulting from the vertical component of ground motion are assumed to be small and are neglected in all the simulations.

Material properties

The mechanical behavior of the concrete material is modeled using the concrete damaged plasticity constitutive model described in "Concrete damaged plasticity," Section 23.6.3 of the Abaqus Analysis User's Guide, and "Damaged plasticity model for concrete and other quasi-brittle materials," Section 4.5.2 of the Abaqus Theory Guide. The material properties used for the simulations are given in Table 2.1.15–1 and Figure 2.1.15–4. These properties are assumed to be representative of the concrete material in the Koyna dam and are based on the properties used by previous investigators. In obtaining some of these material properties, a number of assumptions are made. Of particular interest is the calibration of the concrete tensile behavior. The tensile strength is estimated to be 10% of the ultimate compressive strength (σ_{cu} = 24.1 MPa), multiplied by a dynamic amplification factor of 1.2 to account for rate effects; thus, σ_{t0} = 2.9 MPa. To avoid unreasonable mesh-sensitive results due to the lack of reinforcement in the structure, the tensile postfailure behavior is given in terms of a fracture energy cracking criterion by specifying a stress/displacement curve instead of a stress-strain curve, as shown in Figure 2.1.15–4(a). This is accomplished with the postcracking stress/displacement curve. Similarly, tensile damage, d_t, is specified in tabular form as a function of cracking displacement by using the postcracking damage displacement curve. This curve is shown in Figure 2.1.15–4(b). The stiffness degradation damage caused by compressive failure (crushing) of the concrete, d_c, is assumed to be zero.

Damping

It is generally accepted that dams have damping ratios of about 2–5%. In this example we tune the material damping properties to provide approximately 3% fraction of critical damping for the first mode of vibration of the dam. Assuming Rayleigh stiffness proportional damping, the factor β required to provide a fraction ξ_1 of critical damping for the first mode is given as $\beta = 2\xi_1/\omega_1$. From a natural frequency extraction analysis of the dam the first eigenfrequency is found to be ω_1 = 18.61 rad sec^{-1} (see Table 2.1.15–2). Based on this, β is chosen to be 3.23 × 10^{-3} sec.

Loading and solution control

Loading conditions and solution controls are discussed for each analysis.

Abaqus/Standard analysis

Prior to the dynamic simulation of the earthquake, the dam is subjected to gravity loading and hydrostatic pressure. In the Abaqus/Standard analysis these loads are specified in two consecutive static steps, using a distributed load with the load type labels GRAV (for the gravity load) in the first step and HP (for the hydrostatic pressure) in the second step. For the dynamic analysis in the third step the transverse and vertical components of the ground accelerations shown in Figure 2.1.15–3 are applied to all nodes at the base of the dam.

Since considerable nonlinearity is expected in the response, including the possibility of unstable regimes as the concrete cracks, the overall convergence of the solution in the Abaqus/Standard analysis is expected to be non-monotonic. In such cases automatically setting the time incrementation parameters is generally recommended to prevent premature termination of the equilibrium iteration process because

the solution may appear to be diverging. The unsymmetric matrix storage and solution scheme is activated by specifying an unsymmetric equation solver for the step. This is essential for obtaining an acceptable rate of convergence with the concrete damaged plasticity model since plastic flow is nonassociated. Automatic time incrementation is used for the dynamic analysis of the earthquake, with the half-increment residual tolerance set to 10^7 and a maximum time increment of 0.02 sec.

Abaqus/Explicit analysis

While it is possible to perform the analysis of the pre-seismic state in Abaqus/Explicit, Abaqus/Standard is much more efficient at solving quasi-static analyses. Therefore, we apply the gravity and hydrostatic loads in an Abaqus/Standard analysis. These results are then imported into Abaqus/Explicit to continue with the seismic analysis of the dam subjected to the earthquake accelerogram. We still need to continue to apply the gravity and hydrostatic pressure loads during the explicit dynamic step. In Abaqus/Explicit gravity loading is specified in exactly the same way as in Abaqus/Standard. The specification of the hydrostatic pressure, however, requires some extra consideration because this load type is not currently supported by Abaqus/Explicit. Here we apply the hydrostatic pressure using user subroutine **VDLOAD**.

The Abaqus/Explicit simulation requires a very large number of increments since the stable time increment (6×10^{-6} sec) is much smaller than the total duration of the earthquake (10 sec). The analysis is run in double precision to prevent the accumulation of round-off errors. The stability limit could be increased by using mass scaling; however, this may affect the dynamic response of the structure.

For this particular problem Abaqus/Standard is computationally more effective than Abaqus/Explicit because the earthquake is a relatively long event that requires a very large number of increments in Abaqus/Explicit. In addition, the size of the finite element model is small, and the cost of each solution of the global equilibrium equations in Abaqus/Standard is quite inexpensive.

Results and discussion

The results for each analysis are discussed in the following sections.

Abaqus/Standard results

The results from a frequency extraction analysis of the dam without the reservoir are summarized in Table 2.1.15–2. The first four natural frequencies of the finite element model are in good agreement with the values reported by Chopra and Chakrabarti (1973). As discussed above, the frequency extraction analysis is useful for the calibration of the material damping to be used during the dynamic simulation of the earthquake.

Figure 2.1.15–5 shows the horizontal displacement at the left corner of the crest of the dam relative to the ground motion. In this figure positive values represent displacement in the downstream direction. The crest displacement remains less than 30 mm during the first 4 seconds of the earthquake. After 4 seconds, the amplitude of the oscillations of the crest increases substantially. As discussed below, severe damage to the structure develops during these oscillations.

The concrete material remains elastic with no damage at the end of the second step, after the dam has been subjected to the gravity and hydrostatic pressure loads. Damage to the dam initiates during the seismic analysis in the third step. The evolution of damage in the concrete dam at six different times

during the earthquake is illustrated in Figure 2.1.15–6, Figure 2.1.15–7, and Figure 2.1.15–8. Times $t_1 = 3.96$ sec, $t_3 = 4.315$ sec, and $t_5 = 4.687$ sec correspond to the first three large excursions of the crest in the upstream direction, as shown in Figure 2.1.15–5. Times $t_2 = 4.163$ sec and $t_4 = 4.526$ sec correspond to the first two large excursions of the crest in the downstream direction. Time $t_6 = 10$ sec corresponds to the end of the earthquake. The figures show the contour plots of the tensile damage variable, DAMAGET (or d_t), on the left, and the stiffness degradation variable, SDEG (or d), on the right. The tensile damage variable is a nondecreasing quantity associated with tensile failure of the material. On the other hand, the stiffness degradation variable can increase or decrease, reflecting the stiffness recovery effects associated with the opening/closing of cracks. Thus, assuming that there is no compressive damage ($d_c = 0$), the combination $d_t > 0$ and $d > 0$ at a given material point represents an open crack, whereas $d_t > 0$ and $d = 0$ represents a closed crack.

At time t_1, damage has initiated at two locations: at the base of the dam on the upstream face and in the region near the stress concentration where the slope on the downstream face changes.

When the dam displaces toward the downstream direction at time t_2, the damage at the base leads to the formation of a localized crack-like band of damaged elements. This crack propagates into the dam along the dam–foundation boundary. The nucleation of this crack is induced by the stress concentration in this area due to the infinitely rigid foundation. At this time, some partial tensile damage is also observed on several elements along the upstream face.

During the next large excursion in the upstream direction, at time t_3, a localized band of damaged elements forms near the downstream change of slope. As this downstream crack propagates toward the upstream direction, it curves down due to the rocking motion of the top block of the dam. The crack at the base of the dam is closed at time t_3 by the compressive stresses in this region. This is easily verified by looking at the contour plot of SDEG at time t_3, which clearly shows that the stiffness is recovered on this region, indicating that the crack is closed.

When the load is reversed, corresponding to the next excursion in the downstream direction at time t_4, the downstream crack closes and the stiffness is recovered on that region. At this time tensile damage localizes on several elements along the upstream face, leading to the formation of a horizontal crack that propagates toward the downstream crack.

As the upper block of the dam oscillates back and forth during the remainder of the earthquake, the upstream and downstream cracks close and open in an alternate fashion. The dam retains its overall structural stability since both cracks are never under tensile stress during the earthquake. The distribution of tensile damage at the end of the earthquake is shown in Figure 2.1.15–8, at time t_6. The contour plot of the stiffness degradation variable indicates that, except at the vicinity of the crack tips, all cracks are closed under compressive stresses and most of the stiffness is recovered. No compressive failure is observed during the simulation. The damage patterns predicted by Abaqus are consistent with those reported by other investigators.

Abaqus/Explicit results

Figure 2.1.15–9 shows the distribution of tensile damage at the end of the Abaqus/Explicit simulation. Two major cracks develop during the earthquake, one at the base of the dam and the other at the downstream change of slope. If we compare these results with those from the analysis in Abaqus/Standard (see Figure 2.1.15–8 at time t_6), we find that Abaqus/Standard predicted additional

damage localization zones on the upstream face of the dam. The differences between the results are due to the effect of the dam–reservoir hydrodynamic interactions, which are included in the Abaqus/Standard simulation via an added-mass user element and are ignored in Abaqus/Explicit. This is easily verified by running an Abaqus/Standard analysis without the added-mass user element. The results from this analysis, shown in Figure 2.1.15–10, are consistent with the Abaqus/Explicit results in Figure 2.1.15–9 and confirm that additional damage to the upstream wall occurs when the hydrodynamic interactions are taken into account.

Input files

Abaqus/Standard input files

koyna_freq.inp	Frequency analysis of the Koyna dam.
koyna_std.inp	Seismic analysis of the Koyna dam, including hydrodynamic interactions.
koyna2_std.inp	Seismic analysis of the Koyna dam, not including hydrodynamic interactions.
koyna_haccel.inp	Transverse ground acceleration record.
koyna_vaccel.inp	Vertical ground acceleration record.
addedmass_uel.f	User subroutine **UEL** used by koyna_std.inp to model hydrodynamic interactions via the added mass technique.
koyna_std_to_xpl.inp	Analysis of the pre-seismic state of the Koyna dam. These results are imported by koyna_xpl.inp.

Abaqus/Explicit input files

koyna_xpl.inp	Seismic analysis of the Koyna dam, not including hydrodynamic interactions; requires import of the results from koyna_std_to_xpl.inp.
koyna_hp_vdload.f	User subroutine **VDLOAD** used by koyna_xpl.inp to specify hydrostatic pressure.
koyna2_xpl_std.inp	Analysis of the post-seismic state of the Koyna Dam; requires import of the results from koyna_xpl.inp.

References

- Bhattacharjee, S. S., and P. Léger, "Seismic Cracking and Energy Dissipation in Concrete Gravity Dams," Earthquake Engineering and Structural Dynamics, vol. 22, pp. 991–1007, 1993.
- Cervera, M., J. Oliver, and O. Manzoli, "A Rate-Dependent Isotropic Damage Model for the Seismic Analysis of Concrete Dams," Earthquake Engineering and Structural Dynamics, vol. 25, pp. 987–1010, 1996.
- Chopra, A. K., and P. Chakrabarti, "The Koyna Earthquake and the Damage to Koyna Dam," Bulletin of the Seismological Society of America, vol. 63, no. 2, pp. 381–397, 1973.

- Ghrib, F., and R. Tinawi, "An Application of Damage Mechanics for Seismic Analysis of Concrete Gravity Dams," Earthquake Engineering and Structural Dynamics, vol. 24, pp. 157–173, 1995.
- Lee, J., and G. L. Fenves, "A Plastic-Damage Concrete Model for Earthquake Analysis of Dams," Earthquake Engineering and Structural Dynamics, vol. 27, pp. 937–956, 1998.
- Westergaard, H. M., "Water Pressures on Dams during Earthquakes," Transactions of the American Society of Civil Engineers, vol. 98, pp. 418–433, 1933.

Table 2.1.15–1 Material properties for the Koyna dam concrete.

Young's modulus:	E = 31027 MPa
Poisson's ratio:	ν = 0.15
Density:	ρ = 2643 kg/m^3
Dilation angle:	ψ = 36.31^o
Compressive initial yield stress:	σ_{c0} = 13.0 MPa
Compressive ultimate stress:	σ_{cu} = 24.1 MPa
Tensile failure stress:	σ_{t0} = 2.9 MPa

Table 2.1.15–2 Natural frequencies of the Koyna dam.

Mode	Natural Frequency (rad sec^{-1})	
	Abaqus	Chopra and Chakrabarti (1973)
1	18.86	19.27
2	49.97	51.50
3	68.16	67.56
4	98.27	99.73

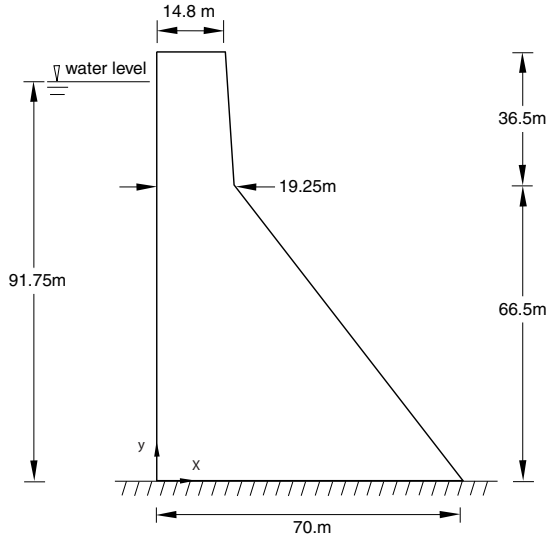


Figure 2.1.15–1 Geometry of the Koyna dam.

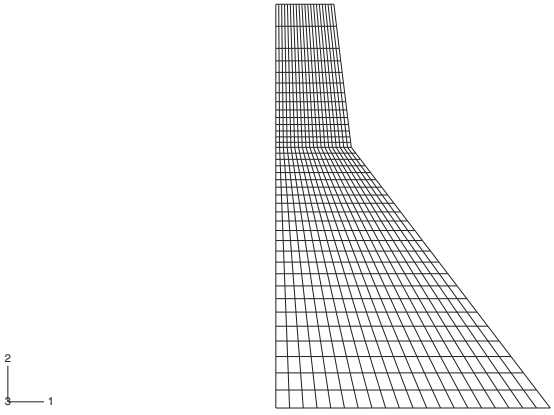


Figure 2.1.15–2 Finite element mesh.

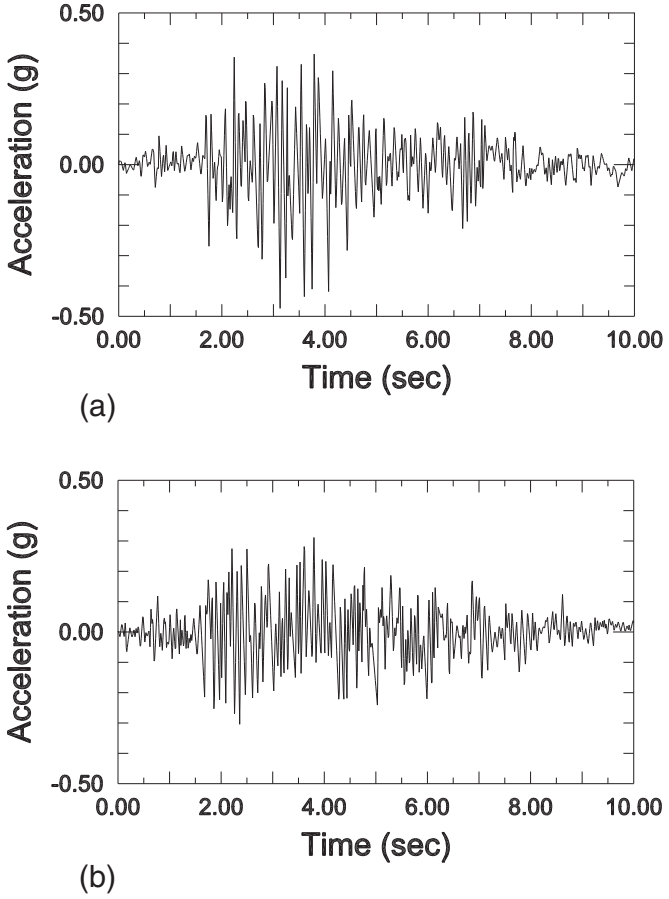


Figure 2.1.15–3 Koyna earthquake: (a) transverse and (b) vertical ground accelerations.

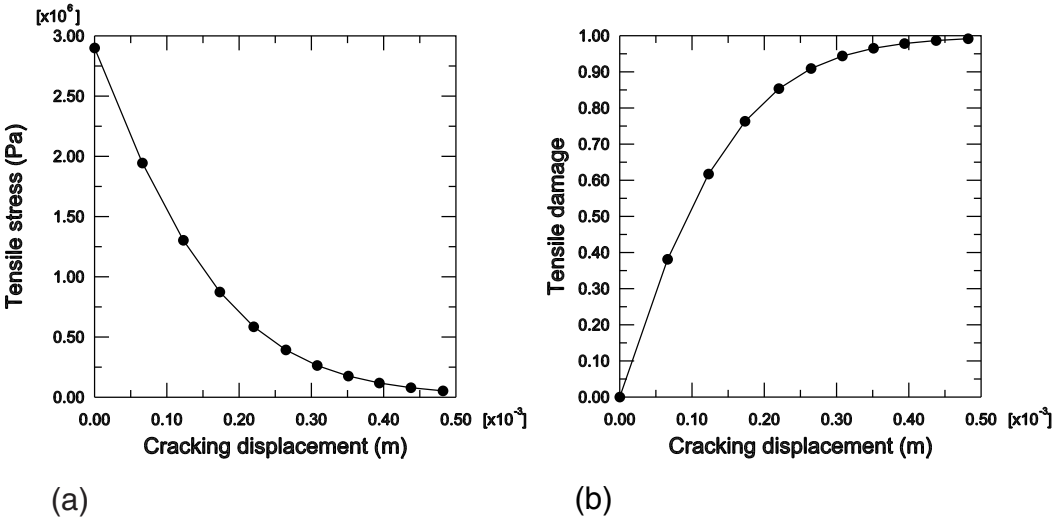


Figure 2.1.15–4 Concrete tensile properties: (a) tension stiffening and (b) tension damage.

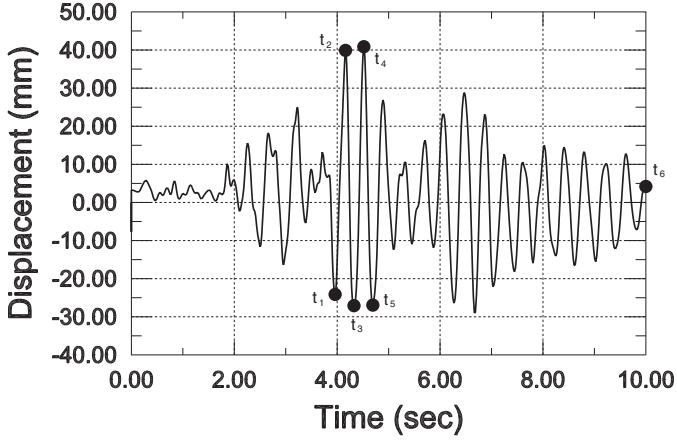


Figure 2.1.15–5 Horizontal crest displacement (relative to ground displacement).

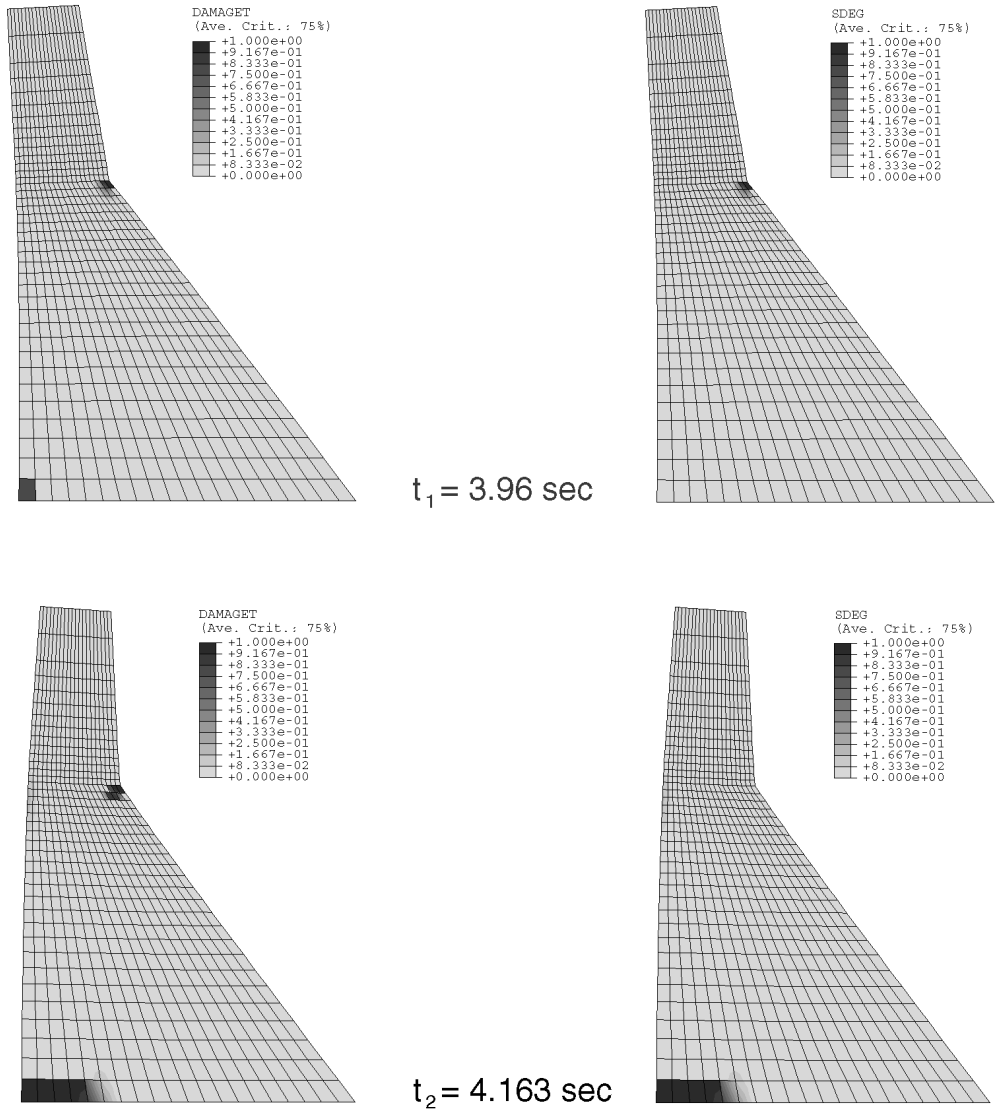


Figure 2.1.15–6 Evolution of tensile damage (Abaqus/Standard); deformation scale factor = 100.

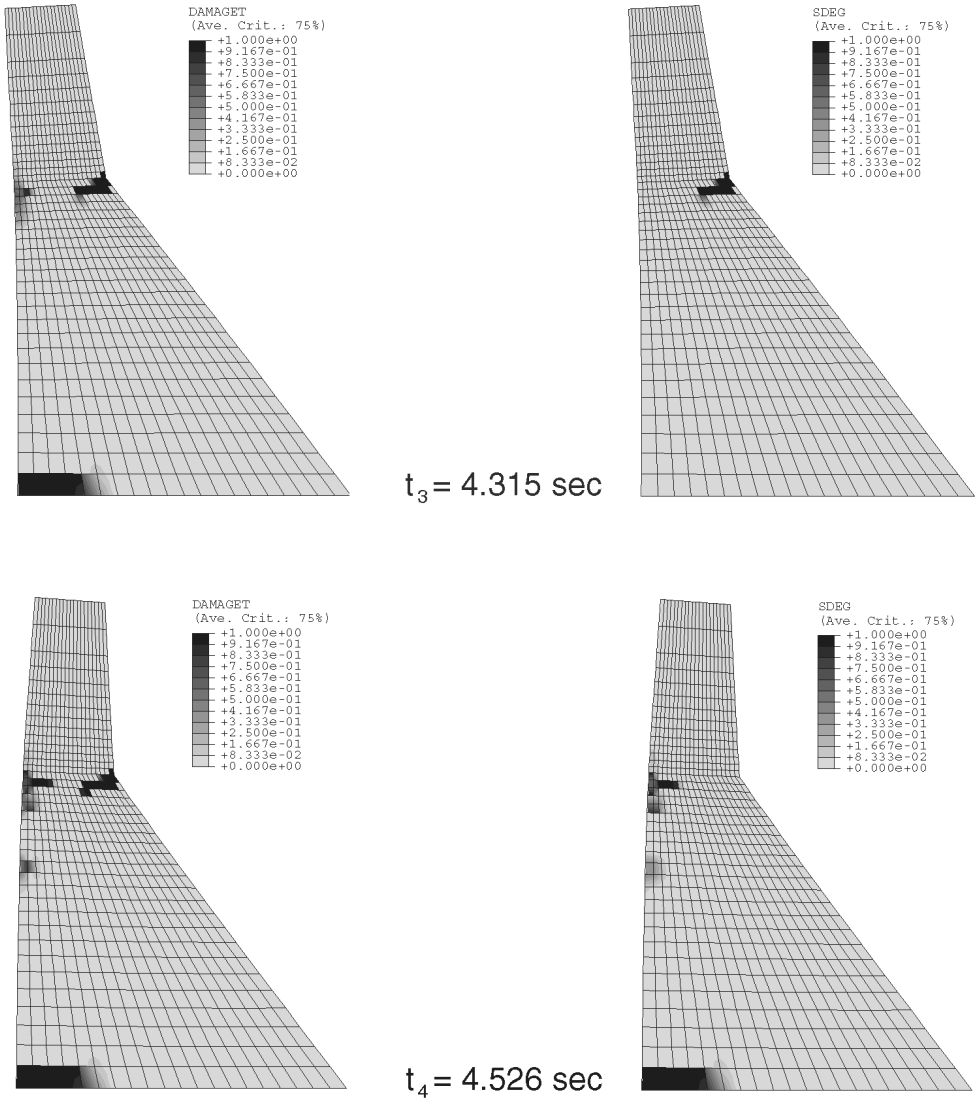


Figure 2.1.15–7 Evolution of tensile damage (Abaqus/Standard); deformation scale factor = 100.

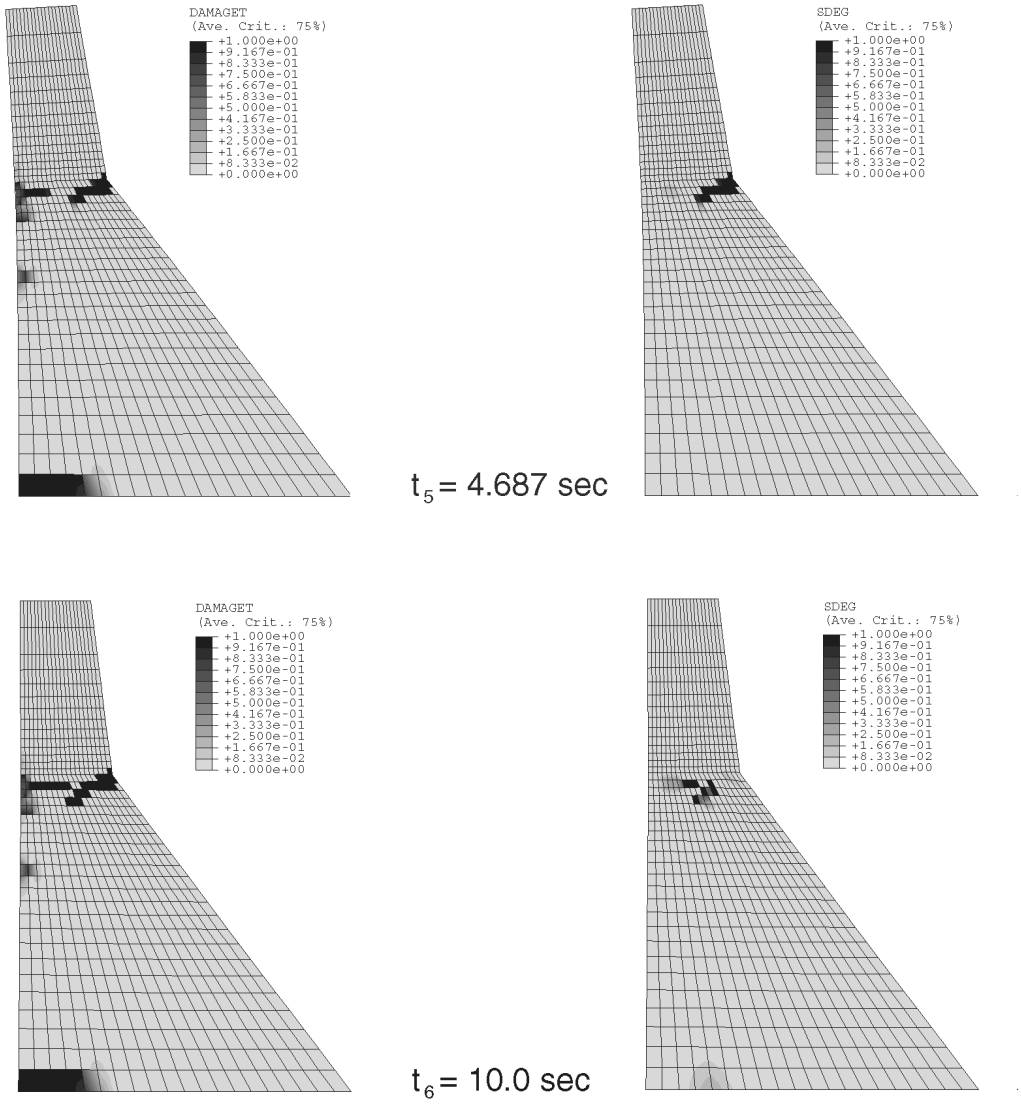


Figure 2.1.15–8 Evolution of tensile damage (Abaqus/Standard); deformation scale factor = 100.

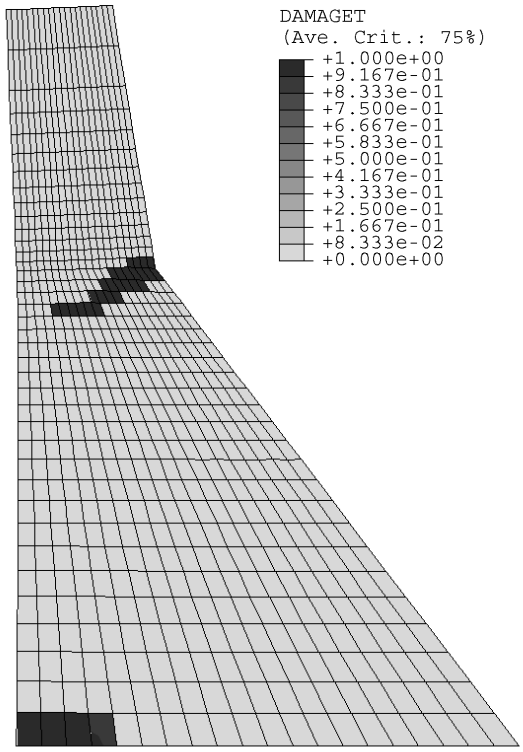


Figure 2.1.15–9 Tensile damage at the end of the Abaqus/Explicit simulation without dam–reservoir hydrodynamic interactions; deformation scale factor = 100.

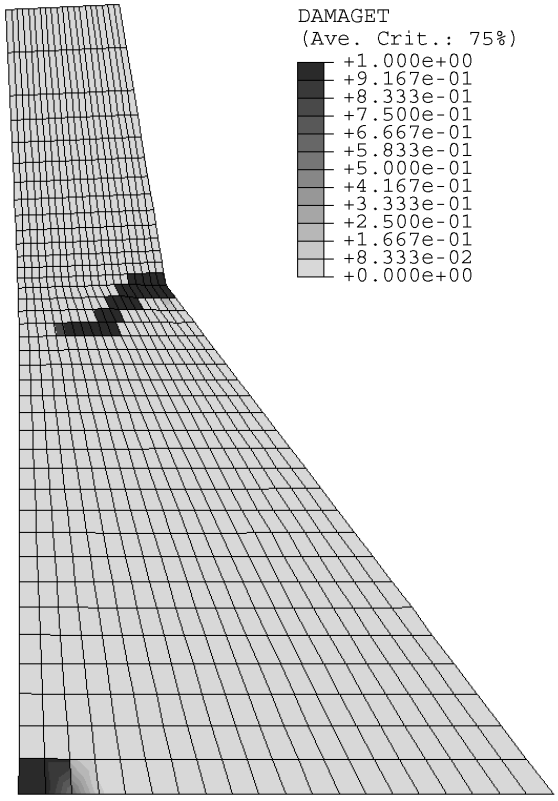


Figure 2.1.15–10 Tensile damage at the end of the Abaqus/Standard simulation without dam–reservoir hydrodynamic interactions; deformation scale factor = 100.

2.1.16 PROGRESSIVE FAILURE ANALYSIS OF THIN-WALL ALUMINUM EXTRUSION UNDER QUASI-STATIC AND DYNAMIC LOADS

Product: Abaqus/Explicit

Objectives

This example problem demonstrates the following Abaqus features and techniques:

- using ductile, shear, and Müschenborn-Sonne forming limit diagram (MSFLD) damage initiation criteria to study the initiation of failure due to three different mechanisms: ductile fracture, shear band formation, and necking instability, respectively; and
- modeling progressive failure of components using damage evolution and element removal.

Application description

New materials such as aluminum and magnesium alloys and high-strength steels are being introduced increasingly in automotive components to reduce weight and, hence, to increase overall vehicle performance. These materials typically have low ductility at fracture compared to traditional steels and may suffer damage and failure under crash loading conditions. A typical component made of sheet metal may undergo damage due to a number of mechanisms including void nucleation and coalescence, shear band formation, and necking instability. Thus, to obtain reliable predictions from crashworthiness simulations, it is essential to model damage initiation and progressive failure due to various failure mechanisms as well as modeling accurate plastic deformation behavior. In this example problem we consider the overall deformation and failure behavior of a thin-wall, double-chambered aluminum extrusion under quasi-static three-point bending and dynamic axial loading conditions. The overall load-displacement response and the fracture patterns are compared with the experimental results given by Hooputra et al. (2004).

Geometry

The three-point bending and the axial crushing configurations are shown in Figure 2.1.16–1 and Figure 2.1.16–6, respectively. The overall dimensions of the aluminum extrusion are L=500 mm, W=95 mm, and H=68 mm for the three-point bending case and L≈396.5 mm, W=95 mm, and H=68 mm for the axial crushing case. The thickness of the sheet is 2.5 mm for both cases.

Materials

The material used in this study is an extruded aluminum alloy EN AW-7108 T6. This material behaves in an elastic-plastic manner and can undergo damage due to either one or a combination of the following damage mechanisms: nucleation and coalescence of voids, shear band formation, and necking instability.

Boundary conditions and loading

The three-point bending configuration consists of the aluminum extrusion supported on two rigid cylinders and loaded in the transverse direction by another rigid cylinder (Figure 2.1.16–1). In the axial crushing simulation, one end of the aluminum extrusion is supported by a fixed rigid base and the other end is subjected to an instantaneous velocity by a planar rigid impactor (Figure 2.1.16–6).

Abaqus modeling approaches and simulation techniques

Two loading cases are considered. The first case consists of a quasi-static three-point bending configuration where the part is loaded transversely to the extrusion direction. In the second case the part is subjected to a dynamic loading in the axial (extrusion) direction.

Summary of analysis cases

Case 1 Quasi-static three-point bending simulation.

Case 2 Dynamic axial crushing simulation.

The sections that follow discuss the analysis considerations that are applicable to both cases.

Mesh design

In both cases the mesh is similar to that used by Hooputra et al. (2004). The aluminum extrusion is meshed with a uniform mesh consisting primarily of 4-node shell elements (S4R). In the axial crushing case some 3-node shell elements (S3R) are also used. The planar dimensions of the elements are an order of magnitude larger than the shell thickness. The simulations with this mesh yield results in agreement with the experimental observations. No mesh refinement studies were conducted.

Material model

The details of the Abaqus models used for constitutive behavior and progressive damage analysis are discussed below. Guidelines for obtaining the material parameters from experimental data are also provided.

Elastoplasticity

Hooputra et al. (2004) have shown that the extruded aluminum alloy EN AW-7108 T6 displays plastic orthotropy due to the nature of the extrusion processing and have used the Barlat symmetric yield locus (Barlat et al., 1991) to fit the experimental data. In this example we neglect the orthotropy and assume both the elastic and the plastic behavior to be isotropic with the yield surface described by the Mises yield function (see "Inelastic behavior," Section 23.1.1 of the Abaqus Analysis User's Guide). The assumption of isotropic plasticity may appear to be too restrictive for the accurate prediction of failure in extruded alloys. However, in crashworthiness simulations the assumption of isotropy usually yields acceptable results when compared with experimental observations, as shown in the results obtained in this example. Nevertheless, you should compare your simulation results with experimental data to check the validity of the isotropic plasticity assumption.

Damage initiation

Metal sheets and thin-walled extrusions made of aluminum alloys may fail due to one or a combination of the following failure mechanisms (Hooputra et al. 2004): ductile failure due to nucleation, growth, and coalescence of voids; shear failure due to fracture within shear bands; and failure due to necking instabilities. If the model consists of shell elements, a criterion for the last failure mechanism is necessary because the size of the localized neck is of the order of the sheet thickness and, hence, cannot be resolved with shell elements of dimensions one order of magnitude larger than the thickness.

Abaqus/Explicit offers a number of damage initiation criteria to model the onset of necking instabilities in sheet metals. These include the Forming Limit Diagram (FLD), Forming Limit Stress Diagram (FLSD), Müschenborn-Sonne Forming Limit Diagram (MSFLD), and Marciniak-Kuczynski (M-K) criteria. The first three criteria utilize the experimentally measured forming limit curves in the appropriate strain or stress spaces. The last criterion introduces virtual thickness imperfections in the sheet metal and analyzes the deformation in the imperfection zone to determine the onset of the instability (see "Damage initiation for ductile metals," Section 24.2.2 of the Abaqus Analysis User's Guide).

The strain-based FLD criterion is limited to applications where the strain path is linear. On the other hand, the stress-based FLSD criterion is relatively insensitive to changes in the strain path. However, this apparent independence of the stress-based limit curve due to the strain path may simply reflect the small sensitivity of the yield stress to changes in the plastic deformation. The M-K criterion can capture the effects of nonlinear strain paths accurately; however, it is computationally expensive, especially if large numbers of imperfection orientations are introduced. It has been verified that the results obtained using the MSFLD criterion are similar to those obtained using the M-K criterion but with a much reduced computational expense (see "Progressive damage and failure of ductile metals," Section 2.2.21 of the Abaqus Verification Guide). Therefore, in this example we choose the MSFLD damage initiation criterion for necking instability.

For specifying the MSFLD damage initiation criterion, the forming limit curve of the material is required. In Abaqus this criterion can be specified by converting the forming limit curve from the space of major versus minor strains to the space of equivalent plastic strain versus ratio of principal strain rates. Abaqus also allows direct specification of the forming limit curve for the MSFLD criterion (see "Müschenborn-Sonne forming limit diagram (MSFLD) criterion" in "Damage initiation for ductile metals," Section 24.2.2 of the Abaqus Analysis User's Guide). We use the forming limit curve based on the experimental work of Hooputra (2005). This curve is assumed to be valid at both the quasi-static and the dynamic strain rates. The parameter OMEGA used in conjunction with the MSFLD criterion to provide filtering of numerical noise in the evaluation of the ratio of principal strain rates is set to 0.001 in both cases (see "Damage initiation for ductile metals," Section 24.2.2 of the Abaqus Analysis User's Guide); this value is recommended for crashworthiness simulations.

Damage due to initiation, growth, and coalescence of voids leads to ductile failure in metals; the formation of cracks within shear bands leads to shear failure. Abaqus offers phenomenological damage initiation criteria for both of these mechanisms. The ductile criterion is specified by providing the equivalent plastic strain at the onset of ductile damage as a function of stress triaxiality and strain rate. Similarly, the shear criterion is specified by providing the equivalent plastic strain at the onset of shear

damage as a function of shear stress ratio and strain rate (see "Damage initiation for ductile metals," Section 24.2.2 of the Abaqus Analysis User's Guide). The data required for both of these criteria may be difficult to obtain through direct experimentation since it would require experiments spanning a range of stress triaxiality and shear stress ratio that may be difficult to achieve. Hooputra et al. (2004) have given simplified analytical expressions for the ductile and the shear failure criteria that require only a limited number of experiments. In this example we adopt those expressions; however, we ignore the orthotropy of the ductile fracture to be consistent with the assumption of isotropic plasticity made earlier.

For the ductile damage initiation criterion the equivalent plastic strain is given by the following function of the stress triaxiality, η (Hooputra et al, 2004):

$$\bar{\varepsilon}_D^{pl}(\eta, \dot{\bar{\varepsilon}}^{pl}) = \frac{\varepsilon_T^+ \sinh[k_0(\eta^- - \eta)] + \varepsilon_T^- \sinh[k_0(\eta - \eta^+)]}{\sinh[k_0(\eta^- - \eta^+)]},$$

where ε_T^+ and ε_T^- correspond to the equivalent plastic strain at ductile damage initiation for equibiaxial tensile and equibiaxial compressive deformation, respectively. For isotropic materials the stress triaxiality in equibiaxial tensile deformation state, η^+, is $2/3$, and in equibiaxial compressive deformation state, η^-, is $-2/3$. The definition of η in Abaqus, as a ratio of the equivalent mean stress to the Mises equivalent stress, differs from that used by Hooputra et al. (2004) by a factor of $1/3$. Consequently, the value of k_0 used in the above expression is three times the value used in Hooputra et al. (2004). The above expression has three parameters that must be obtained experimentally: ε_T^+, ε_T^-, and k_0. These parameters depend on the material, strain rate, and possibly the temperature. For each strain rate of interest, three experiments are needed at different values of stress triaxiality to obtain the three material parameters. ε_T^+ can be obtained directly from the Erichsen test ($\eta = \eta^+$). Three-point bending of sheet coupons (with width/thickness > 4) under plane strain tension ($\eta = 1/\sqrt{3}$) and fracture at the notch root of waisted tensile coupons in uniaxial tension ($\eta = 1/3$) may provide two additional experiments to determine ε_T^- and k_0. In the Erichsen and three-point bending experiments the local fracture strain can be derived by placing a grid on the specimen's surface; in the waisted tensile experiment the fracture strain can be obtained from the sheet thickness in the fracture plane (Hooputra et al., 2004). For the aluminum alloy used in this example the experimentally obtained ductile failure parameters at quasi-static and dynamic strain rates (250 s^{-1}) are listed in Table 2.1.16–1.

For the shear damage initiation criterion the equivalent plastic strain at the onset of damage is given by the following function of the shear stress ratio, θ_s (Hooputra et al, 2004):

$$\bar{\varepsilon}_S^{pl}(\theta_s, \dot{\bar{\varepsilon}}^{pl}) = \frac{\varepsilon_S^+ \sinh[f(\theta_s - \theta_s^-)] + \varepsilon_S^- \sinh[f(\theta_s^+ - \theta_s)]}{\sinh[f(\theta_s^+ - \theta_s^-)]},$$

where $\theta_s = (1 - k_s\eta)/\phi$ with $\phi = \tau_{max}/\sigma_{eq}$, and ε_S^+ and ε_S^- correspond to the equivalent plastic strain at shear damage initiation for equibiaxial tensile and equibiaxial compressive deformation, respectively. The parameters θ_s^+ and θ_s^- correspond to the values of θ_s at $\eta = \eta^+$ and $\eta = \eta^-$, respectively. This expression has four parameters that must be determined experimentally: k_s, ε_S^+, ε_S^-, and f. These parameters depend on the material and strain rate. Hooputra et al. (2004) have used tensile specimens with a groove (rectangular cross-section and groove depth=half the sheet thickness) at $45°$ to the loading direction ($\theta_s = 1.469$), specially designed tensile specimens with a groove parallel to the

loading direction (pure shear, $\theta_s = 1.732$), and Erichsen tests ($\theta_s = 1.6$) in conjunction with the above expression to determine ε_S^+, ε_S^-, and f. The value of the material parameter k_s is taken as 0.3. For the aluminum alloy used in this example the experimentally obtained shear failure parameters at quasi-static and dynamic strain rates (250 s^{-1}) are listed in Table 2.1.16–2.

Using the aforementioned expressions and the material parameters listed in Table 2.1.16–1 and Table 2.1.16–2, tabular data for ductile and shear damage initiation criteria can be generated as a function of stress triaxiality and shear stress ratio, respectively. This tabular data is provided in the Abaqus input files. The above expressions may give very high values of the equivalent plastic strain at damage initiation when the stress triaxiality or the shear stress ratio is very small. A cutoff value of the equivalent plastic strain can be provided in such cases.

Damage evolution and element removal

Damage evolution occurs once the damage initiation criteria are satisfied. Plastic displacement-based linear damage evolution law is used for each of the three damage initiation criterion. The value of the plastic displacement at which the damage variable reaches 1 is taken as 0.1. The default maximum degradation rule is used, and the elements are removed from the mesh once the maximum degradation has occurred (see "Maximum degradation and choice of element removal" in "Damage evolution and element removal for ductile metals," Section 24.2.3 of the Abaqus Analysis User's Guide).

Initial conditions

For the axial crushing simulation a velocity initial condition is specified at the reference node of the planar rigid impactor in the global 1-direction.

Boundary conditions

For the three-point bending simulation all the degrees of freedom at the reference node of the rigid supports are constrained. A velocity boundary condition in the global 2-direction is specified at the reference node of the rigid punch with all the remaining degrees of freedom constrained.

For the axial crushing simulation all the degrees of freedom at the reference node associated with the rigid support are constrained. Furthermore, all of the degrees of freedom except that associated with the global 1-direction are constrained at the reference node of the planar rigid impactor.

Loading

The velocity boundary condition at the rigid punch applies the load in the three-point bending simulation. In the case of the axial crushing simulation the initial velocity of the planar rigid impactor loads the component.

Constraints

Rigid body constraints are specified in both cases to form element-based rigid bodies. These rigid bodies form the support and apply loads to the aluminum extrusion.

Interactions

For the three-point bending simulation a contact pair interaction is defined between the rigid punch and the node-based surface of the aluminum extrusion component. A general contact interaction is defined between the rigid cylinders forming the support and the element-based surface of the aluminum extrusion component. In addition, self-contact is defined between the element-based surfaces of the extruded component. A friction coefficient of 0.05 is used for the contact between the rigid cylinders and the extruded component, and a value of 0.15 is used for the self-contact.

For the axial crushing simulation a contact pair interaction is defined between the extruded component and the rigid support as well as between the component and the rigid impactor. A general contact interaction is used for self-contact between the surfaces of the extruded component. A friction coefficient of 0.15 is used for all of the contact interactions in this case.

For both the three-point bending and axial crushing cases, a penalty-type mechanical constraint is used for all of the contact pair definitions.

Analysis steps

Both the three-point bending and the axial crushing analyses consist of one explicit dynamic step. The total simulation times in the three-point bending and the axial crushing cases are 0.0501 s and 0.072 s, respectively. Both the analyses consider geometric nonlinearity and use automatic time incrementation using element-by-element stable time increment estimates.

Output requests

For both cases the field output request includes the following quantities: displacement, stress, strain, element status, and damage initiation criteria variables. The history output request consists of displacement, velocity, acceleration, and reaction force at the reference point of the top rigid cylinder (for the three-point bending simulation) and at the reference point of the rigid impactor and the supporting base (for the axial crushing simulation). Energy output variables are requested for the entire model.

Discussion of results and comparison of cases

The overall deformed shape of the aluminum extrusion obtained from the three-point bending simulation is shown in Figure 2.1.16–2, and the experimentally observed deformed shape (Hooputra et al. 2004) is shown in Figure 2.1.16–3. The elements that have failed at the end of the simulation are shown in Figure 2.1.16–4, mapped into the undeformed configuration. Good qualitative agreement is seen between the simulation results and experimental observations. The load-displacement history of the punch obtained from the simulation is compared with three different experimental results in Figure 2.1.16–5. Again, a very good match is observed, indicating the reliability of the simulation results. In Figure 2.1.16–5 the simulation results are plotted after applying the Butterworth filter with a cutoff frequency of 1000 (see "Applying Butterworth filtering to an *X*–*Y* data object," Section 47.4.26 of the Abaqus/CAE User's Guide).

The overall deformed shape including the failure patterns obtained from the axial crushing simulation is shown in Figure 2.1.16–7. The deformed shape and the failure patterns are qualitatively similar to those observed experimentally (Figure 2.1.16–8). The overall force-displacement response from the simulation (filtered using the Butterworth filter with a cutoff frequency of 1500) is compared with the results from three different experiments (Hooputra, 2005) in Figure 2.1.16–9. Again, a good qualitative match is seen, and the numerical results are within the experimentally observed scatter.

In conclusion, the results from both the quasi-static three-point bending and the dynamic axial crushing simulations match the experimental data very well. It is also concluded that the use of progressive damage and failure is essential to capture the overall deformation and failure behavior of thin-wall aluminum extrusion.

Input files

Case 1: Three-point bending

threepointbending_alextrusion.inp	Input file to create and analyze the model.

Case 2: Axial crushing

axialcrushing_alextrusion.inp	Input file to create and analyze the model.

References

Abaqus Analysis User's Guide

- "Progressive damage and failure," Section 24.1.1 of the Abaqus Analysis User's Guide

Abaqus Keywords Reference Guide

- *DAMAGE INITIATION
- *DAMAGE EVOLUTION

Abaqus Verification Guide

- "Progressive damage and failure of ductile metals," Section 2.2.21 of the Abaqus Verification Guide

Other

- Barlat, F., D. J. Lege, and J. C. Brem, "A Six-Component Yield Function for Anisotropic Materials," International Journal of Plasticity, vol. 7, pp. 693–712, 1991.
- Hooputra, H., H. Gese, H. Dell, and H. Werner, "A Comprehensive Failure Model for Crashworthiness Simulation of Aluminium Extrusions," International Journal of Crashworthiness, vol. 9, pp. 449–463, 2004.
- Hooputra, H., Private Communication, 2005.

Table 2.1.16–1 Experimentally determined ductile failure parameters (Hooputra et al., 2004).

Parameter	Quasi-static	Dynamic ($\dot{\varepsilon} = 250\,s^{-1}$)
ε_T^+	0.26	0.44
ε_T^-	193.0	1494.0
k_0	5.277	8.6304

Table 2.1.16–2 Experimentally determined shear failure parameters (Hooputra et al., 2004).

Parameter	Quasi-static	Dynamic ($\dot{\varepsilon} = 250\,s^{-1}$)
ε_S^+	0.26	0.35
ε_S^-	4.16	1.2
f	4.04	2.05
k_s	0.3	0.3

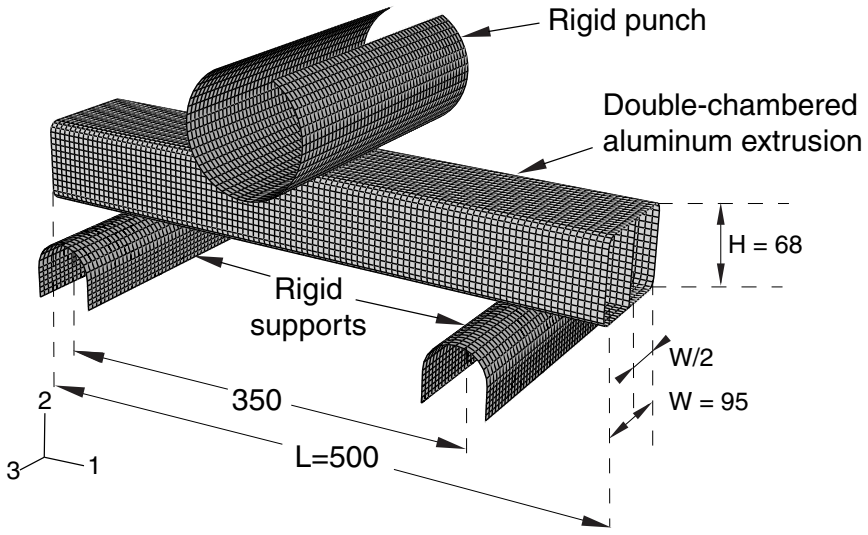


Figure 2.1.16–1 Three-point bending configuration: geometry and finite element mesh.

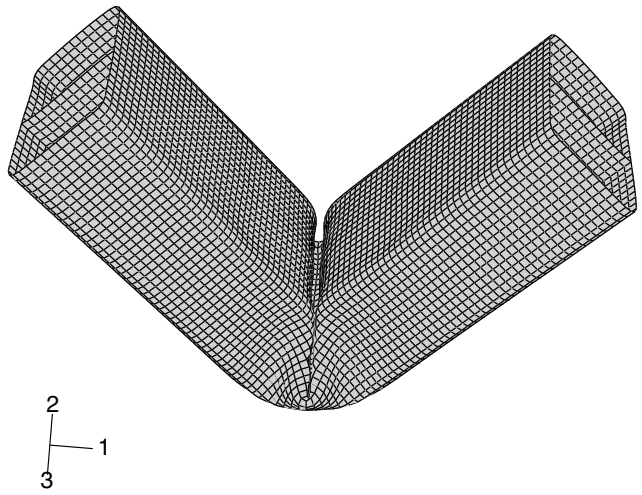


Figure 2.1.16–2 Final deformed shape of the aluminum extrusion in the quasi-static three-point bending simulation.

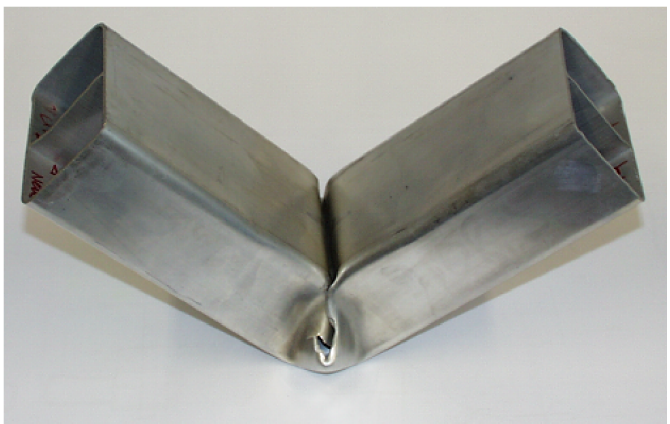


Figure 2.1.16–3 Deformed shape of the aluminum extrusion in the quasi-static three-point bending experiment (Hooputra et al., 2004).

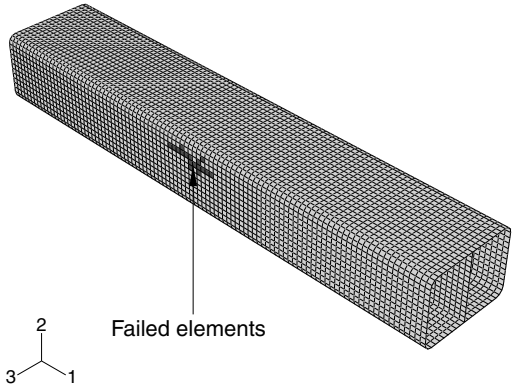


Figure 2.1.16–4 Completely failed elements at the end of the three-point bending simulation mapped into the undeformed configuration.

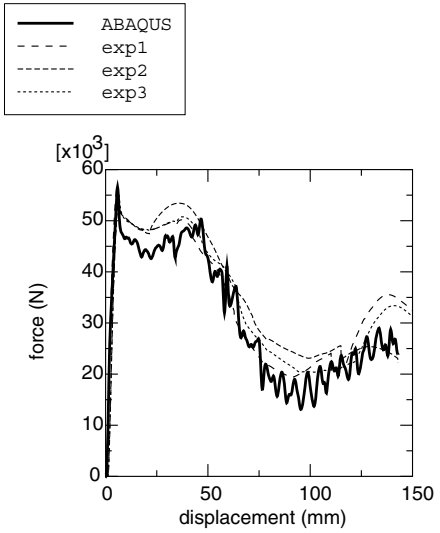


Figure 2.1.16–5 Comparison of the force-displacement response obtained from the three-point bending simulation with the experimental results of Hooputra et al. (2004).

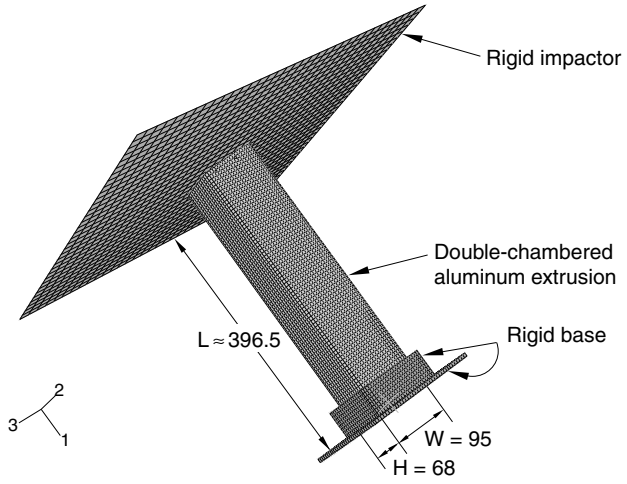


Figure 2.1.16–6 Axial crushing configuration: geometry and finite element mesh.

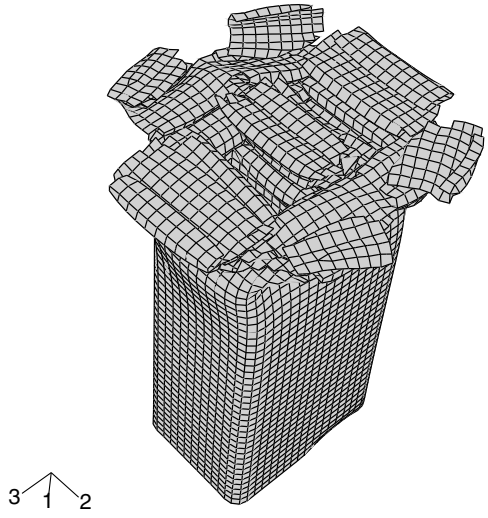


Figure 2.1.16–7 Final deformed shape of the aluminum extrusion in the dynamic axial crushing simulation.

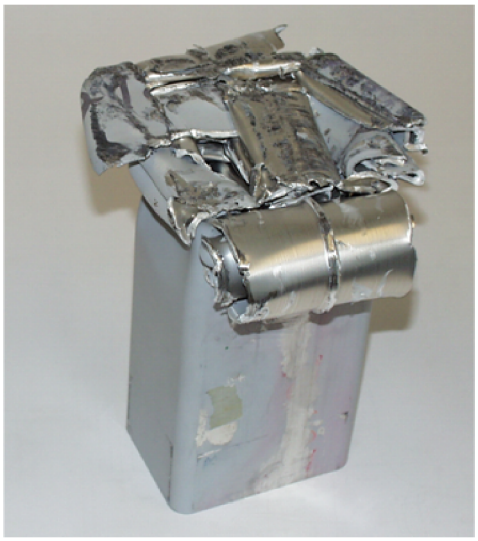


Figure 2.1.16–8 Deformed shape of the aluminum extrusion in the dynamic axial crushing experiment (Hooputra et al., 2004).

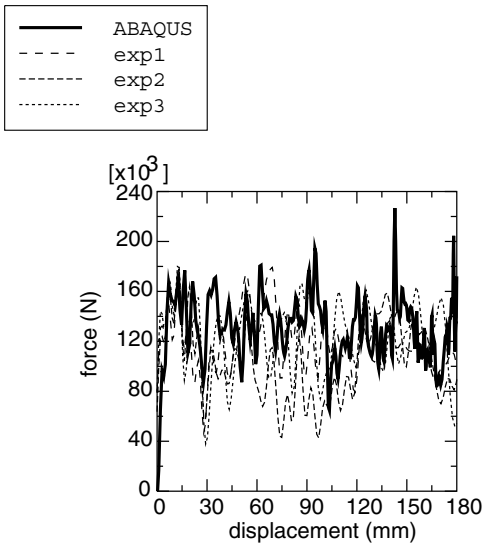


Figure 2.1.16–9 Comparison of the force-displacement response obtained from the axial crushing simulation with the experimental results of Hooputra et al. (2004).

2.1.17 IMPACT ANALYSIS OF A PAWL-RATCHET DEVICE

Product: Abaqus/Standard

Objectives

This example illustrates the use of the direct-integration implicit dynamic procedure with general contact for an example involving impact and significant sliding between contacting surfaces.

Application description

Pawl-ratchet mechanisms exist in a variety of applications such as ratchet lever hoists (also known as come-along winches; see Figure 2.1.17–1) and parking brakes. Some care should be taken when using these devices to minimize dynamic impact between the pawl and ratchet teeth—for example, the lever of a ratchet lever hoist should be controlled such as to gradually engage the antireverse mechanism, and a car should be stopped before putting its transmission into "park." This analysis provides a preliminary assessment of the integrity of a pawl-ratchet mechanism under misuse conditions in which dynamic effects are significant when the pawl engages a ratchet tooth.

Geometry

The model is shown in Figure 2.1.17–2, which somewhat resembles the ratchet mechanism within the tool shown in Figure 2.1.17–1; however, this model is not representative of design details of any specific product. The ratchet mechanism is intended to restrict clockwise motion of the ratchet wheel. The pawl and ratchet wheel are each attached to a stiff support using cylindrical bolts. The ratchet-bolt and pawl-bolt interfaces do not restrict axial rotation of the ratchet and pawl. A torsional spring, which is not shown in the figure, tends to keep the pawl in contact with the ratchet wheel. The ratchet wheel is attached to a hollow shaft. A cable and a lever would also be attached to the shaft in a ratchet lever hoist; however, these aspects are not modeled explicitly.

The model uses a millimeter-tonne-second unit system. The hollow shaft has a length of 40 mm and a radius of 20 mm. The radius of the shaft hole is 12 mm. The width of both the pawl and the ratchet wheel is 8 mm. The radius of the wheel hole is 8 mm, and the radius of the pawl hole is 4 mm. The wheel outer diameter is 100 mm.

Materials

The materials considered in this example are assumed to have linear elastic, perfectly plastic behavior except for the support, which is considered rigid with fixed position. The deformable bodies have an elastic modulus of 21×10^4 MPa, Poisson's ratio of 0.3, and density of 7.9×10^{-9} tonne/mm^3. The yield strength of the pawl is 600 MPa, and the yield strength of the other deformable bodies is 800 MPa.

Interactions

The model employs the general contact capabilities available in Abaqus/Standard to simulate contact interactions between the various parts.

Abaqus modeling approaches and simulation techniques

The analysis is performed using implicit dynamic steps. The first step establishes contact between the pawl and the ratchet wheel, thereby establishing appropriate initial conditions for the second step. Inertia effects are considered during the first step only to avoid issues associated with initially unconstrained rigid body modes that could be troublesome for a purely static procedure; therefore, the quasi-static implicit dynamic application option is used in the first step.

The remainder of the analysis considers the effect of a suddenly applied, sustained net torque acting on the ratchet shaft that leads to impact between the pawl and a ratchet tooth. This applied net torque is meant to approximate the effect of suddenly removing an applied torque that had been counteracting a working load. In a ratchet lever hoist (see Figure 2.1.17–1) this applied net torque approximates the effect of suddenly removing the load on the lever before the pawl is properly engaged with the ratchet (while the hoist is carrying a load). The response due to the applied torque is modeled with a single (second) step in a preliminary analysis and alternatively with three separate steps for the pre-impact, impact, and post-impact phases to facilitate use of smaller time increments for a more accurate analysis of the highly nonlinear impact phase. The moderate dissipation implicit dynamic application option is used for steps modeling the response to the applied torque, which is appropriate for most dynamic contact simulations.

This analysis is meant to approximate a situation in which the cable loading is due to gravity acting on a mass attached to the cable. Since the cable and this mass are not modeled directly, the inertial effect of the mass is approximated with additional rotary inertia on the shaft. In general, if the cable were included in the model, one could expect that the solution may be affected by the cable dynamic response and the details of cable-shaft interactions and self-contact of the cable where it is wrapped onto the shaft. However, in this preliminary study, the model simplifications are assumed to provide reasonably representative results with respect to the severity of the impact loading.

Mesh design

The shaft mesh uses fully integrated first-order continuum (C3D8) elements. All other parts are meshed with reduced integration first-order continuum (C3D8R) elements.

Material model

The deformable materials use the Mises plasticity model without hardening. The material parameters were discussed previously.

Initial conditions

All of the bodies are initially at rest. The positioning of the pawl and ratchet wheel at the beginning of the analysis is shown in Figure 2.1.17–2. A small initial gap exists between the pawl and the ratchet.

Initially, the pawl is not in static equilibrium due to initial loading by a torsional spring, as discussed in "Constraints" below.

Boundary conditions

The reference node of the support (which is considered rigid) has all components of translation and rotation fixed. The annular face of the shaft away from the ratchet wheel is constrained to a reference node that is positioned on the shaft axis, at the center of the annular face. Zero displacement along the global X- and global Y-axes is prescribed for this reference node.

Loading

After the initialization step, force and torque act on the ratchet shaft due to cable loading associated with a 0.2 tonne mass under gravitational loading (as discussed previously, this mass and the cable are not modeled explicitly). A force of 1960 N and torque of 42,140 Nmm act on the ratchet shaft via a distributing coupling reference node. These concentrated loads are applied instantaneously at the beginning of the second step and remain constant in time.

The applied torque leads to impact loading (modeled with general contact), which is of primary interest in this analysis.

Constraints

All translations and rotations for the rigid support are fixed. The circumferential surface of the bolts is partitioned in two halves (with respect to a plane perpendicular to their axis) with one half being tied to the corresponding hole in the support part, using a tie constraint. A distributing coupling is employed to apply concentrated force and torque to the outer cylindrical surface of the hollow shaft. One of the annular faces of the shaft is tied to the ratchet wheel. A kinematic coupling is employed to constrain the other annular face of the shaft (i.e., the face away from the ratchet wheel) to a reference node at the center of this face.

A torsional spring—modeled using a connector element—is used for maintaining contact between the pawl and the ratchet wheel. One end of the connector is coupled using a distributing coupling to the outer cylindrical surface of the pawl.

Interactions

Contact is defined using the default general contact inclusions option. This is the easiest way to ensure that Abaqus/Standard will enforce any mechanical interactions between various parts of the model. A friction coefficient of 0.15 is specified for all interactions. Circumferential smoothing is applied to all cylindrical surfaces (associated with bolts and bolt holes), which was facilitated by using Abaqus/CAE to create the model (Abaqus/CAE assigns circumferential smoothing by default to cylindrical surfaces).

A connector element (CONN3D2) using a revolute connection type with a prescribed reference angle is employed for representing the torsional spring, which maintains contact between the pawl and ratchet. The rotational stiffness of the connector element is 2000 Nmm/rad. A reference angle of 45° is defined for the connector behavior. When in static equilibrium, the pawl presses the ratchet with a contact force of about 31.5 N.

Results and discussion

The evolution of the angular velocity of the ratchet wheel is shown in Figure 2.1.17–3 for the preliminary analysis that uses two load steps. This angular velocity remains zero during the first step because no net torque acts on the ratchet wheel during that step. Once the torque loading is applied after the first step, an approximately linear increase in the absolute value of this angular velocity occurs while the pawl slides along the ratchet; during this time the net torque acting on the ratchet wheel (affected by the applied torque and frictional forces) is approximately constant, so the angular acceleration is also nearly constant. The sign of the angular acceleration changes upon impact. Large contact forces transmitted between the pawl and ratchet tooth eventually lead to a rebound. This simulation completes before any secondary rebounds occur (the analysis duration could be extended if secondary rebounds were of interest).

In a subsequent 4-step analysis, the initialization step remains the same but the response to torque loading is modeled with three steps instead of one step to facilitate specification of smaller time increments during impact. These three steps correspond to the pre-impact, impact, and post-impact phases that are apparent in Figure 2.1.17–3. Usage of smaller time increments during impact enables more accurate modeling of the highly nonlinear response during this phase. Figure 2.1.17–4 shows contour plots of equivalent plastic strain at the end of the 2-step and 4-step analyses in the vicinity of the pawl-to-ratchet impact for the imagined mechanism. The maximum equivalent plastic strain, which occurs at the tip of the pawl, is approximately 8% in the 2-step analysis and approximately 10% in the 4-step analysis. The 4-step analysis uses a time increment during impact about one-fifth of that in the 2-step analysis. The maximum equivalent plastic strain is quite insensitive to further reductions in the time increment during impact. The difference between the equivelent plastic strain plots shown in Figure 2.1.17–4 demonstrates the importance of using an adequately small time increment to model portions of an analysis with sudden changes in acceleration. Less severe plastic deformation is predicted in the ratchet tooth and, as shown on the right side of Figure 2.1.17–5, in the vicinity of the pawl-bolt interface.

The evolution of kinetic energy (ALLKE) and energy dissipation due to yielding (ALLPD) is plotted for the 2-step and 4-step analyses in Figure 2.1.17–6. Plastic dissipation occurs while the ratchet wheel decelerates after impact; differences in plastic dissipation for the two analyses are again apparent in Figure 2.1.17–6. The various phases of the simulation can be identified in the kinetic energy plot, including the pre-impact, impact, rebound, and post-rebound phases.

Input files

pawlratchet_impact.inp	Implicit dynamic 2-step analysis.
pawlratchet_impact_4step.inp	Implicit dynamic 4-step analysis.

References

Abaqus Analysis User's Guide

- "Implicit dynamic analysis using direct integration," Section 6.3.2 of the Abaqus Analysis User's Guide
- "Defining general contact in Abaqus/Standard," Section 36.2 of the Abaqus Analysis User's Guide

Abaqus Keywords Reference Guide

- *CONTACT
- *DYNAMIC

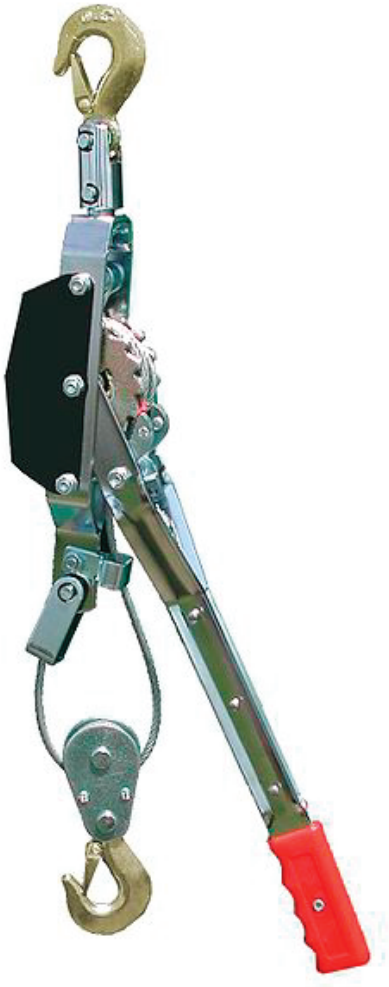


Figure 2.1.17–1 Ratchet lever hoist (come-along winch).

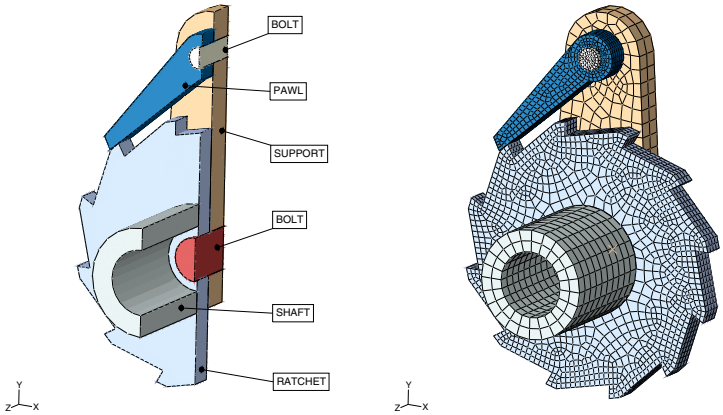


Figure 2.1.17–2 Geometry and mesh of the pawl-ratchet model.

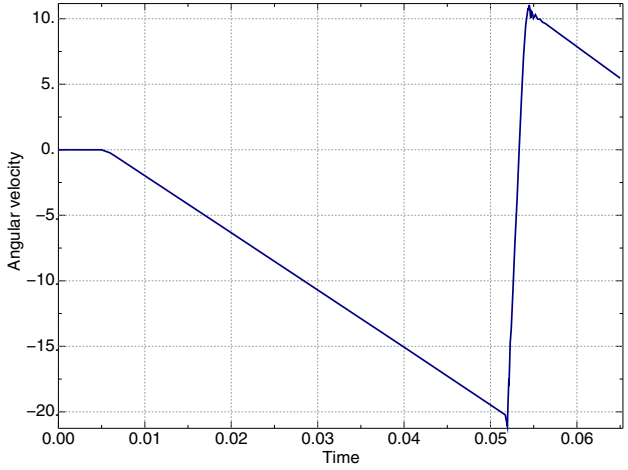


Figure 2.1.17–3 Evolution of angular velocity of the ratchet wheel.

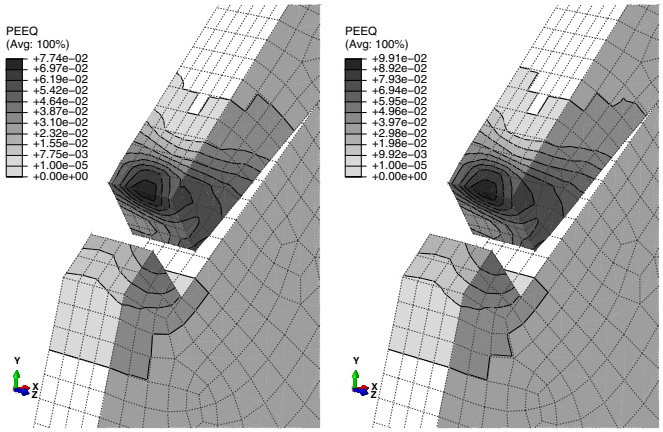


Figure 2.1.17–4 Distribution of equivalent plastic strain (PEEQ) near pawl tip for 2-step analysis (left) and 4-step analysis (right).

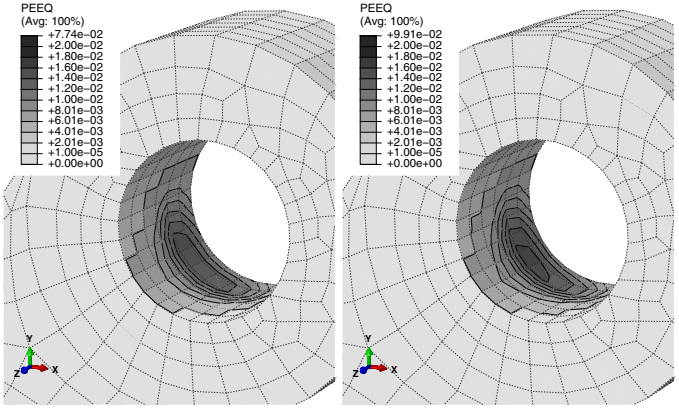


Figure 2.1.17–5 Distribution of equivalent plastic strain (PEEQ) near pawl hole for 2-step analysis (left) and 4-step analysis (right).

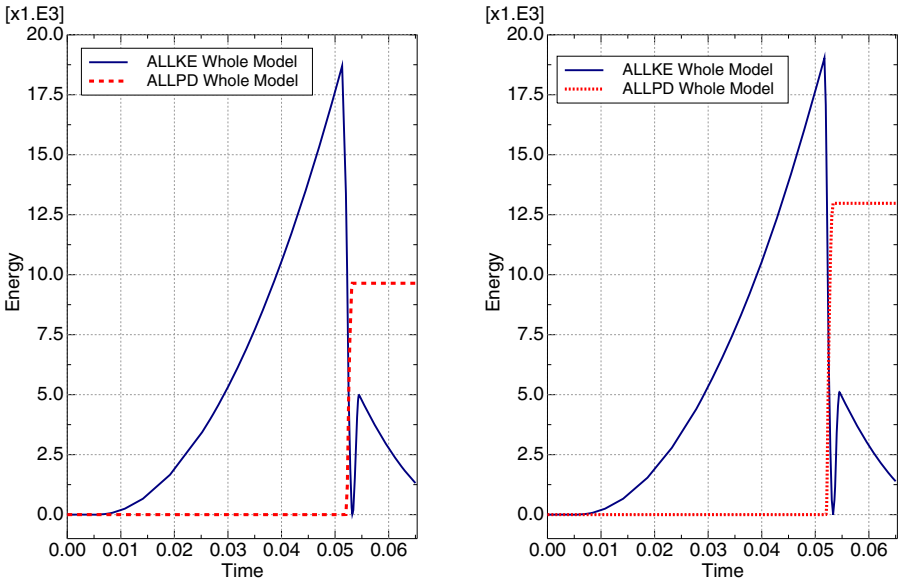


Figure 2.1.17–6 Evolution of the kinetic and plastically dissipated energy for 2-step analysis (left) and 4-step analysis (right).

2.1.18 HIGH-VELOCITY IMPACT OF A CERAMIC TARGET

Product: Abaqus/Explicit

Objectives

This example problem illustrates the following Abaqus features and techniques:

- using the Johnson-Holmquist-Beissel (JHB) and the Johnson-Holmquist (JH-2) ceramic material models to study the high-velocity impact of a silicon carbide target. The JHB and JH-2 models are available in Abaqus/Explicit as built-in user materials;
- achieving a similar material response for ceramics with proper calibration of the Drucker-Prager plasticity and the equation of state functionality in Abaqus/Explicit; and
- comparing numerical results with published results.

Application description

Ceramic materials are commonly used in armor protection applications. In recent years Johnson, Holmquist, and their coworkers have developed a series of constitutive relations to simulate the response of ceramic materials under large strain, high-strain rate, and high-pressure impacting conditions. In this example the JHB and JH-2 material models are explored to investigate the penetration velocity of a gold projectile impacting on a silicon carbide target. The computed results are compared with published results given by Holmquist and Johnson (2005).

Geometry

The initial configuration is shown in Figure 2.1.18–1. Both target and projectile are of cylindrical shape. The silicon carbide target has a radius of 7.5 mm and a length of 40 mm. The gold projectile has a radius of 0.375 mm and a length of 30 mm.

Materials

The target material is silicon carbide. This material is very hard and mainly used under compressive load conditions and can only sustain very little tension. Typical applications include bulletproof vests and car brakes due to its high endurance. The strength has a dependence on pressure. In high-speed impact applications, damage to the material plays an important role in the evolution of the strength. The totally failed silicon carbide will not sustain any load. The projectile is gold, which is soft compared to the target material.

Initial conditions

An initial velocity of 4000 m/s is prescribed for the projectile.

Interactions

The projectile will penetrate into the target due to the high-speed impact.

Abaqus modeling approaches and simulation techniques

Three cases are investigated, each using a different approach to model the silicon carbide material: the first case uses the JHB material model, the second uses the JH-2 model, and the third case uses a combination of several Abaqus options to obtain a similar constitutive model within a more general framework. The Lagrangian description is used for both projectile and target. General contact with surface erosion is used for all three cases. Element deletion and node erosion are considered. A tonne-millimeter-second unit system was chosen for all simulations.

Summary of analysis cases

Case 1 JHB (built-in user material)

Case 2 JH-2 (built-in user material)

Case 3 Combination of Drucker-Prager plasticity, equation of state, and Johnson-Cook rate dependence

The sections that follow discuss the analysis considerations that are applicable to all cases.

Analysis types

An Abaqus/Explicit dynamic analysis is used for all the simulations. The total duration for the penetration process is 7 μs.

Mesh design

An 11.5° slice of the cylinders is modeled. There are five elements along the radial direction of the projectile. The element size along the radial direction for the target is nearly the same as for the projectile. Due to the large radius ratio between the projectile and the target, there are 343,980 elements for the target and 2000 elements for the projectile. Figure 2.1.18–2 shows part of the meshes used for the analysis.

Material model

The different material models used for the silicon carbide target are discussed in detail in subsequent sections. The JHB and JH-2 models are available as built-in user materials for Abaqus (i.e., via **`VUMAT`** subroutines that are built-in). These built-in materials are invoked by using material names starting with **`ABQ_JHB`** and **`ABQ_JH2`**, respectively. For descriptions of the ceramic material models, see "Analyzing ceramics with the Johnson-Holmquist and Johnson-Holmquist-Beissel material models" in the Dassault Systèmes Knowledge Base at www.3ds.com/support/knowledge-base.

The material for the projectile is gold. The density is 19,240 kg/m^3. The shear modulus is 27.2 GPa. The hydrodynamic behavior is described by the Mie-Grüneisen equation of state. The linear $U_s - U_p$ Hugoniot form is used. The parameters are c_0 = 2946.16 m/s, s = 3.08623, and Γ_0 = 2.8. The strength

is 130 MPa described as a perfect plasticity. A ductile damage initiation criterion with the equivalent plastic strain of 0.2 at the onset of damage is used. The fracture energy is chosen as 0 for the damage evolution.

Initial conditions

Initial velocity conditions of 4000 m/s are specified for all the nodes of the projectile in the axial direction toward the target.

Boundary conditions

All the nodes on the symmetry axis, which is set up as the global x-direction, can move only along this axis, so zero velocity boundary conditions are prescribed for both the y- and z-directions. To satisfy the axial symmetry boundary conditions, a cylindrical coordinate system is established. The circumferential degrees of freedom for all the nodes on the two side surfaces except the nodes on the symmetry axis of both the target and the projectile are prescribed with zero velocity boundary conditions. The nodes on the non-impacting end of the target are fixed along the axial direction.

Interactions

General contact is used to model the interactions between the projectile and the target. The interior surface of both the target and the projectile is included to enable element removal.

Analysis steps

There is only one explicit dynamic analysis step, during which the penetration takes place.

Output requests

In addition to the standard output identifiers available in Abaqus, the solution-dependent state variables described in Table 2.1.18–1 and Table 2.1.18–2 are also available for output for the JHB and JH-2 models, respectively.

Constitutive models for ceramic materials under high-velocity impact

This section provides a detailed description of the different constitutive models for ceramic materials that are used to model the silicon carbide target for each of the cases considered.

Case 1: Johnson-Holmquist-Beissel model

This case uses the JHB model for the silicon carbide target. The JHB material parameters for silicon carbide given in Holmquist and Johnson (2005) are used in this study. They are listed in Table 2.1.18–3.

The JHB model consists of three main components: a representation of the deviatoric strength of the intact and fractured material in the form of a pressure-dependent yield surface, a damage model that transitions the material from the intact state to a fractured state, and an equation of state (EOS) for the pressure-density relation that can include dilation (or bulking) effects as well as a phase change (not considered in this study).

Strength

The strength of the material is expressed in terms of the von Mises equivalent stress, σ, and is a function of the pressure, P, the dimensionless equivalent strain rate, $\dot{\epsilon}^* = \dot{\bar{\epsilon}}^{pl}/\dot{\epsilon}_0$ (where $\dot{\bar{\epsilon}}^{pl}$ is the equivalent plastic strain rate and $\dot{\epsilon}_0$ is the reference strain rate), and the damage variable, D ($0 \le D \le 1$). For the intact (undamaged) material, $D = 0$, whereas $D = 1$ for a fully damaged material.

For a dimensionless strain rate of $\dot{\epsilon}^* = 1.0$, the strength of the intact material ($D = 0$) takes the form

$$\sigma = \sigma_i(P + T)/(P_i + T) \quad \text{if} \quad \mathrm{P} \le \mathrm{P_i},$$

$$\sigma = \sigma_i + (\sigma_i^{\max} - \sigma_i)\{1 - \exp[-\alpha_i(P - P_i)]\} \quad \text{if} \quad \mathrm{P} \ge \mathrm{P_i},$$

where $\alpha_i = \sigma_i/[(\sigma_i^{\max} - \sigma_i)(P_i + T)]$ and P_i, T, σ_i, and $\sigma_i^{\max}$ are material parameters. The strength of the fractured material ($D = 1$) is given by

$$\sigma = \sigma_f P/P_f \quad \text{if} \quad \mathrm{P} \le \mathrm{P_f},$$

$$\sigma = \sigma_f + (\sigma_f^{\max} - \sigma_f)\{1 - \exp[-\alpha_f(P - P_f)]\} \quad \text{if} \quad \mathrm{P} \ge \mathrm{P_f},$$

where $\alpha_f = \sigma_f/[(\sigma_f^{\max} - \sigma_f)P_f]$ and P_f, σ_f, and $\sigma_f^{\max}$ are material parameters.

The intact and fractured strengths above are for a dimensionless strain rate of $\dot{\epsilon}^* = 1.0$. The effect from strain rates is incorporated by the Johnson-Cook strain rate dependence law as $\sigma = \sigma_0(1 + C \ln \dot{\epsilon}^*)$, where σ_0 is the strength corresponding to $\dot{\epsilon}^* = 1.0$. Plastic flow is volume preserving and is governed by a Mises flow potential.

Damage

The damage initiation parameter, ω, accumulates with plastic strain according to

$$\omega = \sum \frac{\Delta \bar{\epsilon}^{pl}}{\bar{\epsilon}_f^{pl}(P)},$$

where $\Delta \bar{\epsilon}^{pl}$ is the increment in equivalent plastic strain and $\bar{\epsilon}_f^{pl}(P)$ is the equivalent plastic strain to fracture under constant pressure, defined as

$$\bar{\epsilon}_f^{pl} = D_1(P^* + T^*)^{D_2}; \quad \bar{\epsilon}_{f,\min}^{pl} \le \bar{\epsilon}_f^{pl} \le \bar{\epsilon}_{f,\max}^{pl},$$

where D_1 and D_2 are material constants and $P^* = P/\sigma_i^{\max}$ and $T^* = T/\sigma_i^{\max}$. The optional parameters $\bar{\epsilon}_{f,\min}^{pl}$ and $\bar{\epsilon}_{f,\max}^{pl}$ are provided for additional flexibility to limit the minimum and maximum values of the fracture strain.

The JHB model assumes that the material fails immediately, $D = 1$ when $\omega = 1$. For other values of $\omega < 1$, there is no damage ($D = 0$) and the material preserves its intact strength.

Pressure

The equations for the pressure-density relationship without phase change are used in this study and are listed here.

$$P = K_1\mu + K_2\mu^2 + K_3\mu^3 \quad \text{if} \quad \mu \geq 0,$$

$$P = K_1\mu \quad \text{if} \quad \mu \leq 0,$$

where $\mu = \rho/\rho_0 - 1$. In the above, K_1, K_2, and K_3 are constants (K_1 is the initial bulk modulus); ρ is the current density; and ρ_0 is the reference density. The model includes the effects of dilation or bulking that occur when brittle materials fail by including an additional pressure increment, ΔP, such that

$$P = K_1\mu + K_2\mu^2 + K_3\mu^3 + \Delta P.$$

The pressure increment is determined from energy considerations as

$$\Delta P = -K_1\mu_f + \sqrt{(K_1\mu_f)^2 + 2\beta K_1 \Delta U},$$

where μ_f is the current value of μ at the time of failure and β is the fraction of the elastic energy loss converted to potential hydrostatic energy ($0 \leq \beta \leq 1$). The bulking pressure is computed only for failure under compression ($\mu_f > 0$).

Case 2: Johnson-Holmquist model

The second case uses the JH-2 model. Unlike the JHB model, the JH-2 model assumes that the damage variable increases progressively with plastic deformation. The material parameters used for the JH-2 model are listed in Table 2.1.18–4. The JH-2 model similarly consists of three components.

Strength

The strength of the material is expressed in terms of the normalized von Mises equivalent stress as

$$\sigma^* = \sigma_i^* - D(\sigma_i^* - \sigma_f^*),$$

where σ_i^* is the normalized intact equivalent stress, σ_f^* is the normalized fractured equivalent stress, and D is the damage variable. The normalized equivalent stresses (σ^*, σ_i^* and σ_f^*) have the general form $\sigma^* = \sigma/\sigma_{\text{HEL}}$, where σ is the actual von Mises equivalent stress and σ_{HEL} is the equivalent stress at the Hugoniot elastic limit (HEL). The model assumes that the normalized intact and fractured stresses can be expressed as functions of the pressure and strain rate as

$$\sigma_i^* = A(P^* + T^*)^N(1 + C\ln\dot{\epsilon}^*) \leq \sigma_i^{\max},$$

$$\sigma_f^* = B(P^*)^M(1 + C\ln\dot{\epsilon}^*) \leq \sigma_f^{\max}.$$

The material parameters are A, B, C, M, N, and the optional limits for the strengths $\sigma_i^{\max}$ and $\sigma_f^{\max}$. The normalized pressure is defined as $P^* = P/P_{\rm HEL}$, where P is the actual pressure and $P_{\rm HEL}$ is the pressure at the HEL. The normalized maximum tensile hydrostatic pressure is $T^* = T/P_{\rm HEL}$, where T is the maximum tensile pressure that the material can withstand.

Damage

The damage initiation parameter, ω, accumulates with plastic strain according to

$$\omega = \sum \frac{\Delta\bar{\epsilon}^{pl}}{\bar{\epsilon}_f^{pl}(P)},$$

where $\Delta\bar{\epsilon}^{pl}$ is the increment in equivalent plastic strain and $\bar{\epsilon}_f^{pl}(P)$ is the equivalent plastic strain to fracture under constant pressure, defined as

$$\bar{\epsilon}_f^{pl} = D_1(P^* + T^*)^{D_2} \quad \bar{\epsilon}_{f,\min}^{pl} \leq \bar{\epsilon}_f^{pl} \leq \bar{\epsilon}_{f,\max}^{pl}.$$

The JH-2 model assumes that the damage variable increases gradually with plastic deformation by setting $D = \omega$.

Pressure

The equations for the pressure-density relationship are similar to the JHB model.

$$P = K_1\mu + K_2\mu^2 + K_3\mu^3 \quad \text{if} \quad \mu \geq 0,$$

$$P = K_1\mu \quad \text{if} \quad \mu \leq 0,$$

where $\mu = \rho/\rho_0 - 1$. The model includes the effects of dilation or bulking that occur when brittle materials fail by including an additional pressure increment, ΔP, such that

$$P = K_1\mu + K_2\mu^2 + K_3\mu^3 + \Delta P.$$

The pressure increment is determined from energy considerations as

$$\Delta P_{t+\Delta t} = -K_1\mu_{t+\Delta t} + \sqrt{(K_1\mu_{t+\Delta t} + \Delta P_t)^2 + 2\beta K_1 \Delta U},$$

where β is the fraction of the elastic energy loss converted to potential hydrostatic energy ($0 \leq \beta \leq 1$).

Case 3: Drucker-Prager model

We use a calibrated Drucker-Prager plasticity model and an equation of state to obtain a material behavior that is similar to that of the JH-2 model. In this way we are not confined to follow the specific expressions and, subsequently, the material parameters of the JH models. The calibration of the Drucker-Prager plasticity model and the equation of state is described below.

Strength

We use the general exponent form of the extended Drucker-Prager model ("Extended Drucker-Prager models," Section 23.3.1 of the Abaqus Analysis User's Guide) which, after some manipulations, can be written as follows:

$$\sigma = \frac{1}{a^{1/b}}(P + P_t)^{1/b}.$$

We have replaced q with σ and p with P to be consistent with the symbols used in the JH-2 models. This expression is very similar to that of the intact strength of the JH-2 model

$$\sigma_i = \frac{\sigma_{\mathrm{HEL}} A}{P_{\mathrm{HEL}}^N}(P + T)^N.$$

Comparing these two expressions, the equations to calibrate the material parameters in the Drucker-Prager model can be obtained as

$$a = \frac{P_{\mathrm{HEL}}}{(A\sigma_{\mathrm{HEL}})^{1/N}},$$

$$b = \frac{1}{N},$$

$$P_t = T.$$

After substituting the values for silicon carbide in Table 2.1.18–4, we find $a = 3.920173 \times 10^{-3}$ MPa $^{1-\frac{1}{N}}$ and $b = 1.53846$. After obtaining the material parameters a and b, the uniaxial compression yield stress, σ_c, can be calibrated by solving the following equation.

$$P_t = a\sigma_c^b - \frac{\sigma_c}{3} = T.$$

For this silicon carbide, σ_c = 6605.66 MPa. The Johnson-Cook type rate dependence can also be used together with the Drucker-Prager plasticity model.

Damage

The ductile damage initiation criterion in Abaqus ("Damage initiation for ductile metals," Section 24.2.2 of the Abaqus Analysis User's Guide) can be calibrated to reproduce the damage criterion used in the

JH-2 damage model. The ductile criterion requires the specification of the equivalent plastic strain at the onset of damage as a function of the stress triaxiality. Along the intact strength curve of the JH-2 model, the stress triaxiality is given as

$$-\frac{P}{\sigma_i} = -P/[\sigma_{\mathrm{HEL}} A(\frac{P}{P_{\mathrm{HEL}}} + \frac{T}{P_{\mathrm{HEL}}})^N].$$

Given $\bar{\epsilon}_f^{pl}$, the damage evolution relation for the JH-2 model gives the following expression for the pressure P:

$$P = P_{\mathrm{HEL}}(\frac{\bar{\epsilon}_f^{pl}}{D_1})^{1/D_2} - T.$$

Substituting this expression into the stress triaxiality expression, we finally obtain the functional relationship between $\bar{\epsilon}_f^{pl}$ and the stress triaxiality needed for the ductile damage initiation criterion. An additional consideration when specifying the damage initiation criterion for the Drucker-Prager plasticity model is that the definition of the equivalent plastic strain is different from that used in the JH-2 model. The two are related by the following plastic work statement:

$$\sigma \dot{\bar{\epsilon}}^{pl} = \sigma_c \dot{\bar{\epsilon}}_c^{pl}.$$

In summary, we specify some sampling points for $\bar{\epsilon}^{pl}$ and calculate the corresponding pressure P and, in turn, the intact strength σ_i for the JH-2 model. We calculate the stress triaxiality using these P and σ_i values. We convert the $\bar{\epsilon}^{pl}$ values into $\bar{\epsilon}_c^{pl}$ through the above plastic work statement. In this manner, we finally obtain a table of data pairs for the equivalent plastic strain at damage initiation and the stress triaxiality for the Drucker-Prager model.

Pressure

The Mie-Grüneisen equation of state ("Mie-Grüneisen equations of state" in "Equation of state," Section 25.2.1 of the Abaqus Analysis User's Guide) is used to describe the hydrodynamic behavior of the silicon carbide material. The linear $U_s - U_p$ Hugoniot form is used. Without the energy contribution (Γ_0 = 0.0), the pressure is expressed as

$$P = \frac{\rho_0 c_0^2 \eta}{(1 - s\eta)^2},$$

where $\eta = 1 - \rho_0/\rho = \mu/(1 + \mu)$. Using a Taylor expansion with respect to μ, the linear and quadratic coefficients of the polynomial can be identified as K_1 and K_2 in the pressure density relation for the JH-2 model, which gives

$$K_1 = \rho_0 c_0^2,$$

$$K_2 = \rho_0 c_0^2 (2s - 1).$$

We solve for the parameters c_0 and s. The values are c_0 = 8272.2 m/s and s = 1.32.

Discussion of results and comparison of cases

The penetration depths from the three models and the published results in the reference by Holmquist and Johnson (2005) at 3 μs, 5 μs, and 7 μs are listed in Table 2.1.18–5. All results from the three models match the published results well. Especially, the results obtained with the Drucker-Prager model are in satisfactory agreement with all other results obtained with the JH models. The final configurations for the JHB, JH-2, and Drucker-Prager models are shown in Figure 2.1.18–3, Figure 2.1.18–4, and Figure 2.1.18–5, respectively. The wave propagation results can be improved by increasing the angle of the wedge.

Input files

Case 1: JHB material model

exa_impactsiliconcarbide_jhb.inp	Input file to create and analyze the model.

Case 2: JH-2 material model

exa_impactsiliconcarbide_jh2.inp	Input file to create and analyze the model.

Case 3: Drucker-Prager material model

exa_impactsiliconcarbide_dp.inp	Input file to create and analyze the model.

References

Abaqus Analysis User's Guide

- "Progressive damage and failure," Section 24.1.1 of the Abaqus Analysis User's Guide
- "Extended Drucker-Prager models," Section 23.3.1 of the Abaqus Analysis User's Guide

Abaqus Keywords Reference Guide

- *DAMAGE INITIATION
- *DAMAGE EVOLUTION
- *DRUCKER PRAGER

Other

- Holmquist, T. J., Johnson, G. R., "Characterization and Evaluation of Silicon Carbide for High-Velocity Impact," Journal of Applied Physics, vol. 97, 093502, 2005.
- Johnson, G. R., Holmquist, T. J., "An Improved Computational Constitutive Model for Brittle Materials," High Pressure Science and Technology–1993, New York, AIP Press, 1993.

Table 2.1.18–1 Solution-dependent state variables defined in JHB model.

Output variables	Symbol	Description
SDV1	$\bar{\epsilon}^{pl}$	Equivalent plastic strain PEEQ
SDV2	$\dot{\bar{\epsilon}}^{pl}$	Equivalent plastic strain rate
SDV3	ω	Damage initiation criterion
SDV4	D	Damage variable
SDV5	δP	Pressure increment due to bulking
SDV6	σ	Yield strength
SDV7	$\mu_{\max}$	Maximum value of volumetric strain μ
SDV8	μ	Volumetric strain $\mu = \rho/\rho_0 - 1$
SDV9	$\mathrm{MpStatus}$	Material point status: 1 if active, 0 if failed

Table 2.1.18–2 Solution-dependent state variables defined in JH-2 model.

Output variables	Symbol	Description
SDV1	$\bar{\epsilon}^{pl}$	Equivalent plastic strain PEEQ
SDV2	$\dot{\bar{\epsilon}}^{pl}$	Equivalent plastic strain rate
SDV3	ω	Damage initiation criterion
SDV4	D	Damage variable
SDV5	δP	Pressure increment due to bulking
SDV6	σ	Yield strength
SDV7	μ	Volumetric strain $\mu = \rho/\rho_0 - 1$
SDV8	$\mathrm{MpStatus}$	Material point status: 1 if active, 0 if failed

Table 2.1.18–3 Material parameters for JHB model.

Line 1	ρ_0	G	σ_i	P_i	σ_f	P_f	C	$\dot{\epsilon}_0$
	3215 kg/m^3	193 GPa	4.92 GPa	1.5 GPa	0.1 GPa	0.25 GPa	0.009	1.0
Line 2	T	$\sigma_i^{\max}$	$\sigma_f^{\max}$	β				
	0.75 GPa	12.2 GPa	0.2 GPa	1.0				
Line 3	D_1	D_2	$\bar{\epsilon}_{f,min}^{pl}$	FS				
	0.16	1.0	999	0.2				
Line 4	K_1	K_2	K_3					
	220 GPa	361 GPa	0 GPa					
Line 5	$\bar{K}_1$	$\bar{K}_2$	$\bar{K}_3$	μ_1	μ_2	μ_0	h	
	0	0	0	0	0	0	0	

Table 2.1.18–4 Material parameters for JH-2 model.

Line 1	ρ_0	G	A	N	B	M	C	$\dot{\epsilon}_0$
	3215 kg/m^3	193 GPa	0.96	0.65	0.35	1.0	0.009	1.0
Line 2	T	$\sigma_i^{\max}$	$\sigma_f^{\max}$	HEL	p_{HEL}	β		
	0.75 GPa	1.24	0.132	11.7 GPa	5.13 GPa	1.0		
Line 3	D_1	D_2	$\bar{\epsilon}_{f,\max}^{pl}$	$\bar{\epsilon}_{f,\min}^{pl}$	FS	lDamage		
	0.48	0.48	1.2	0.0	0.2	0		
Line 4	K_1	K_2	K_3					
	220 GPa	361 GPa	0 GPa					

Table 2.1.18–5 Results comparison with the reference results.

Results	Penetration depth		
	3 μs	5 μs	7 μs
Published results	7.23 mm	12.05 mm	16.87 mm
JHB model	7.19 mm	12.10 mm	16.88 mm
JH-2 model	7.18 mm	12.03 mm	16.89 mm
Drucker-Prager model	7.18 mm	11.80 mm	16.52 mm

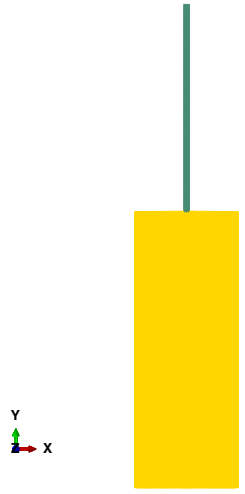


Figure 2.1.18–1 Silicon carbide target and gold projectile: geometry.

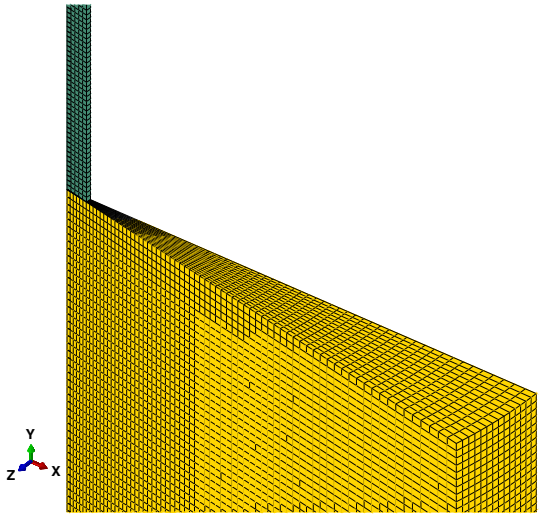


Figure 2.1.18–2 Silicon carbide target and gold projectile: mesh.

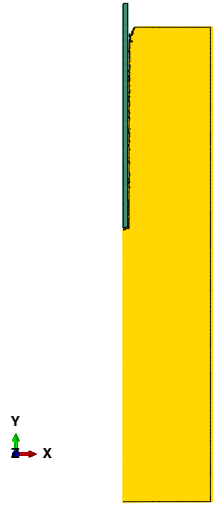


Figure 2.1.18–3 Final configuration at 7 μs using the JHB model.

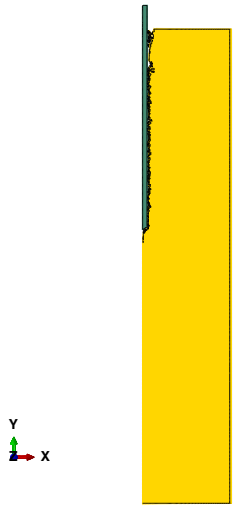


Figure 2.1.18–4 Final configuration at 7 μs using the JH-2 model.

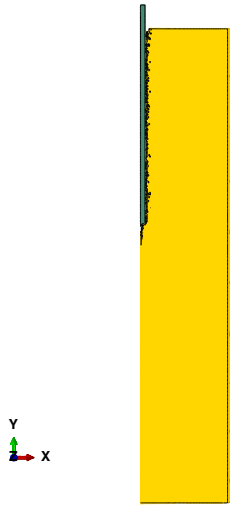


Figure 2.1.18–5 Final configuration at 7 μs using the Drucker-Prager model and the Mie-Grüneisen equation of state.

2.2 Mode-based dynamic analyses

- "Analysis of a rotating fan using substructures and cyclic symmetry," Section 2.2.1
- "Linear analysis of the Indian Point reactor feedwater line," Section 2.2.2
- "Response spectra of a three-dimensional frame building," Section 2.2.3
- "Brake squeal analysis," Section 2.2.4
- "Dynamic analysis of antenna structure utilizing residual modes," Section 2.2.5
- "Steady-state dynamic analysis of a vehicle body-in-white model," Section 2.2.6

2.2.1 ANALYSIS OF A ROTATING FAN USING SUBSTRUCTURES AND CYCLIC SYMMETRY

Product: Abaqus/Standard

This example illustrates the single and multi-level substructure capability of Abaqus for problems where the part being modeled consists of repeated structures. It also demonstrates the capability of Abaqus to analyze cyclic symmetric models using the cyclic symmetry analysis technique. Some of the limitations of modeling a structure using substructures or cyclic symmetry are also discussed.

Geometry and material

The structure is a fan consisting of a central hub and four blades, as shown in Figure 2.2.1–1. The blades and the hub are made up of S4R shell elements. The material is elastic, with a Young's modulus of 200 GPa and a Poisson's ratio of 0.29. The density of the material is 7850 kg/m^3. All nodes along the mounting hole in the hub are fixed.

Models

Four different models are considered, as follows:

1. The fan is modeled as a single structure (no substructures).
2. One quadrant of the fan, consisting of a quarter of the hub and a single blade, is reduced to a substructure. The fan is then modeled with four substructures (a single-level substructure). During substructure generation all degrees of freedom are retained for the nodes along the edges of the hub in each quadrant as well as one node at the blade tip (see Figure 2.2.1–2).
3. A single fan blade is reduced to a substructure, which is then combined with one-quarter of the hub to form a higher level substructure. Four of these substructures are then combined to form the fan (similar to the single-level substructure), thus forming a multi-level substructure. Nodes along the base of the fan blade and one node at the tip of the blade have all their degrees of freedom retained during generation of the fan blade substructure as shown in Figure 2.2.1–2. At the higher level substructure generation stage, nodes along the edge of the hub in each quadrant as well as the node at the blade tip have their degrees of freedom retained.
4. One quadrant of the fan, consisting of a quarter of the hub and a single blade, is modeled with and without substructures as a datum sector for the cyclic symmetry analysis technique. Two surfaces, which are at 90° to each other, are chosen to serve as the slave and master surfaces for the cyclic symmetry surface-based tie constraint. The finite element mesh contains matching nodes on the symmetry surfaces; therefore, both surfaces are defined with lists of nodes or node set labels. The axis of cyclic symmetry is parallel to the global z-axis and passes through the point on the x–y plane with coordinates (3.0, 3.0). The cyclic symmetry model is shown in Figure 2.2.1–3. The entire model consists of four repetitive sectors.

Both a frequency analysis and a static analysis are performed on the first three models. Static analysis followed by a frequency extraction and a modal-based steady-state dynamic analysis are performed for the cyclic symmetry model. Stress- and load-stiffening effects due to the centrifugal loading on the fan are built into the substructure stiffness during generation using a preload step with large-displacement formulation. To get the proper stress stiffening in the hub of the multi-level substructure, the centrifugal load defined in the lowest-level substructure (the blade) needs to be captured with a substructure load case and must be applied as a substructure load in the next-level substructure.

To improve the representation of the substructure's dynamic behavior in the global analysis, m dynamic modes, which are extracted using an eigenfrequency extraction step, are included during the substructure generation using eigenmode selection. The reduced mass matrix obtained with the default value of $m = 0$ corresponds to the Guyan reduction technique, while $m > 0$ corresponds to the restrained mode addition technique. In the "Results and discussion" section below the solution obtained for the model without substructures (the "full model") is used as the reference solution.

For the cyclic symmetry model without substructures the eigenvalue extraction procedure is performed on the preloaded structure. The nonlinear static step has the centrifugal load applied to the blade. Eigenvalues are requested using the Lanczos eigenvalue solver, which is the only eigensolver that can be used for an eigenfrequency analysis with the cyclic symmetry analysis technique. The specification of cyclic symmetry modes in an eigenvalue analysis is demonstrated in one problem. This makes it possible to extract only the eigenmodes that have the requested cyclic symmetry. When this option is omitted, the eigenvalues are extracted for all possible (three) cyclic symmetry modes. In the discussion that follows the solution obtained for the cyclic symmetry model is compared to the solution for the entire 360° model (the reference solution). An eigenvalue analysis without the preload step is performed for the cyclic symmetry model with substructures. Twenty eigenvalues are extracted and compared to the reference solution obtained for the entire 360° model with substructures. The third step in the cyclic symmetry model problems is a frequency-domain, modal-based, steady-state procedure. It calculates the response to pressure loads projected on a specific cyclic symmetry mode.

Results and discussion

Results for the frequency analysis and the static analysis appear below.

Frequency analysis for models with substructures

Frequencies corresponding to the 15 lowest eigenvalues are extracted and tabulated in Table 2.2.1–1 for each model. To study the effect of retaining dynamic modes during substructure generation, the substructure models are run after extracting 0, 5, and 20 dynamic modes during substructure generation.

While the Guyan reduction technique (m =0) yields frequencies that are reasonable compared to those of the full model, the values obtained with 5 retained modes are much closer to full model predictions, especially for the higher eigenvalues. Increasing the number of retained modes to 20 does not yield a significant improvement in the results, consistent with the fact that in the Guyan reduction technique the choice of retained degrees of freedom affects accuracy, while for the restrained mode addition technique the modes corresponding to the lowest frequencies are by definition optimal.

When substructures are used in an eigenfrequency analysis, it is to be expected that the lowest eigenfrequency in the substructure model is higher than the lowest eigenfrequency in the corresponding model without substructures. This is indeed the case for the single-level substructure analysis, but for the multi-level substructure analysis the lowest eigenfrequency is *below* the one for the full model. This occurs because the stress and load stiffness for the lowest-level substructure (the blade) are generated with the root of the blade fixed, whereas in the full model the root of the blade will move radially due to the deformation of the hub under the applied centrifugal load. Hence, the substructure stiffness is somewhat inaccurate. Since the radial displacements at the blade root are small compared to the overall dimensions of the model (of order 10^{-3}), the resulting error should be small, as is observed from the results.

Table 2.2.1–2 shows what happens if the NLGEOM parameter is omitted during the preloading steps. It is clear that the results are significantly different from the ones that take the effect of the preload on the stiffness into account. In this case the lowest eigenfrequency in the substructure models is indeed above the lowest eigenfrequency in the model without substructures.

Static analysis for models with substructures

A static analysis of the fan is carried out about the preloaded base state by applying a pressure load of 10^5 Pa normal to the blades of the fan. The axial displacement of the outer edge of the fan blade due to the pressure load is monitored at nodes along path AB, as shown in Figure 2.2.1–1. The results are shown in Figure 2.2.1–4; there is good agreement between the solutions for the substructure models and the full model.

While substructures can be generated from models that exhibit nonlinear response, it must be noted that, once created, a substructure always exhibits linear response at the usage level. Hence, a preloaded substructure will produce a response equivalent to that of the response to a linear perturbation load on a preloaded full model. Consequently, the full model is analyzed by applying the centrifugal preload in a general step and the pressure load in a linear perturbation step. Since an analysis using substructures is equivalent to a perturbation step, the results obtained do not incorporate the preload deformation. Thus, if the total displacement of the structure is desired, the results of this perturbation step need to be added to the base state solution of the structure.

Steady-state analysis with preload for the cyclic symmetry model

A modal-based, steady-state analysis of the fan is carried out about the preloaded base state, as shown in fan_cyclicsymmodel_ss.inp. In the general static step, which includes nonlinear geometry, the centrifugal load is applied to the datum sector. Only symmetric loads can be applied in general static steps with the cyclic symmetry analysis technique. A sequence of three eigenvalue extraction and steady-state dynamics steps follows the preload step. Each eigenvalue extraction requests only one cyclic symmetry mode that is used in the load projection in the steady-state dynamic analysis that follows. The analysis specifies that modes belonging to the cyclic symmetry modes 0, 1, and 2 should be extracted. The computed eigenvalues are identical to those obtained for the entire 360° model, as shown in Table 2.2.1–1. The additional information obtained during the eigenvalue extraction is the cyclic symmetry mode number associated with each eigenvalue. In the case of 4 repetitive sectors, all the eigenvalues corresponding to cyclic symmetry mode 1 appear in pairs; the eigenvalues corresponding to

modes 0 and 2 are single. The lowest two eigenvalues correspond to cyclic symmetry mode 1, followed by the single eigenvalues corresponding to cyclic symmetry modes 2 and 0. For a comparison with the cyclic symmetry model option, the eigenvalue problem is also modeled with multi-point constraint type CYCLSYM (see fansubstr_mpc.inp). To verify the use of substructures with the cyclic symmetry model, it was determined that the eigenvalues obtained with fansubstr_cyclic.inp were identical to those obtained with fansubstr_1level_freq.inp. The last step is the modal-based, steady-state dynamic analysis. A pressure load is applied to the entire structure as projected onto three different cyclic symmetry modes.

Input files

fan_cyclicsymmodel_ss.inp	Cyclic symmetry model with static, eigenvalue, and steady-state dynamics steps with the load projected onto the cyclic modes 0, 1, and 2, respectively.
fansubstr_1level_freq.inp	Single-level substructure usage analysis with a frequency extraction step.
fansubstr_1level_static.inp	Single-level substructure usage analysis with a static step.
fansubstr_multi_freq.inp	Multi-level substructure usage analysis with a frequency extraction step.
fansubstr_multi_static.inp	Multi-level substructure usage analysis with a static step.
fansubstr_freq.inp	Frequency extraction without substructures.
fansubstr_static.inp	Static analysis without substructures.
fansubstr_mpc.inp	Single-level usage analysis demonstrating the use of cyclic symmetry MPCs.
fansubstr_gen1.inp	Substructure generation for a single blade used in the multi-level substructure generation file fansubstr_gen2.inp.
fansubstr_gen2.inp	Multi-level substructure generation used in fansubstr_multi_freq.inp and fansubstr_multi_static.inp.
fansubstr_gen3.inp	Single-level substructure generation used in fansubstr_1level_freq.inp, fansubstr_1level_static.inp, and fansubstr_mpc.inp.
fansubstr_cyclic.inp	Single-level substructure with the cyclic symmetry model used in a frequency analysis.

Table 2.2.1–1 Comparison of natural frequencies for single-level and multi-level substructures with the values for the model without substructures.

Eigenvalue no. cycles/sec	With substructuring: 1 level			With substructuring: 2 levels			Full model
	m=0	m=5	m=20	m=0	m=5	m=20	
1	6.9477	6.7901	6.7891	6.7655	6.6269	6.6258	6.7890
2	6.9477	6.7901	6.7891	6.7655	6.6269	6.6258	6.7890
3	8.0100	7.7207	7.7198	7.8162	7.5563	7.5552	7.7198
4	8.2009	7.8816	7.8810	8.1986	7.8813	7.8807	7.8810
5	11.341	11.020	11.010	11.123	10.802	10.792	11.009
6	11.341	11.020	11.010	11.123	10.802	10.792	11.009
7	12.529	11.930	11.912	11.539	11.142	11.124	11.910
8	14.751	14.397	14.346	13.450	13.256	13.211	14.348
9	17.787	14.432	14.432	17.208	14.455	14.455	14.431
10	18.922	14.779	14.775	18.797	14.751	14.747	14.774
11	21.250	14.779	14.775	19.860	14.751	14.747	14.774
12	21.250	16.034	15.995	19.860	15.645	15.623	15.991
13	28.250	17.699	17.624	28.066	17.129	17.057	17.624
14	28.691	19.034	19.019	28.628	18.914	18.901	19.008
15	28.691	21.333	21.178	28.628	20.014	19.885	21.176

Table 2.2.1–2 Comparison of natural frequencies for single-level and two-level substructures with the full model values without the use of the NLGEOM parameter.

Eigenvalue no. cycles/sec	With substructuring		Full model
	1 level	2 levels	
1	4.4795	4.4795	4.4795
2	4.4795	4.4795	4.4795
3	4.5511	4.5511	4.5511
4	4.8889	4.8889	4.8889
5	9.5431	9.5431	9.5426
6	9.5431	9.5431	9.5426
7	9.7921	9.7921	9.7918
8	12.632	12.633	12.632
9	14.005	14.005	14.005
10	14.336	14.336	14.336
11	14.336	14.336	14.336
12	15.489	15.489	15.489
13	16.861	16.861	16.860
14	18.241	18.241	18.232
15	19.036	19.036	19.034

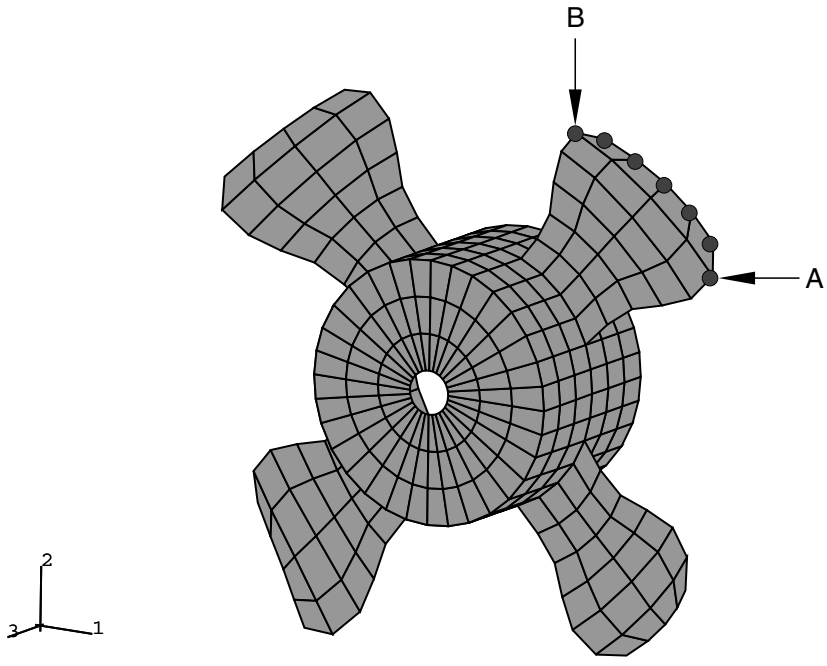
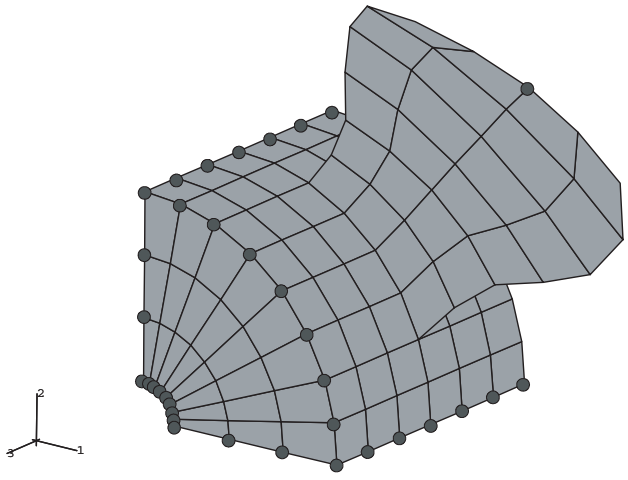
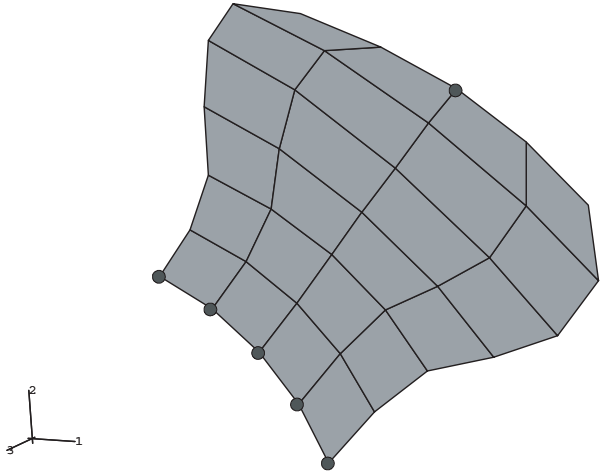


Figure 2.2.1–1 Mesh used for the complete fan model.



(a) One-level substructure showing retained nodes.



(b) First level of the two-level substructure showing retained nodes.

Figure 2.2.1–2 Substructures generated.

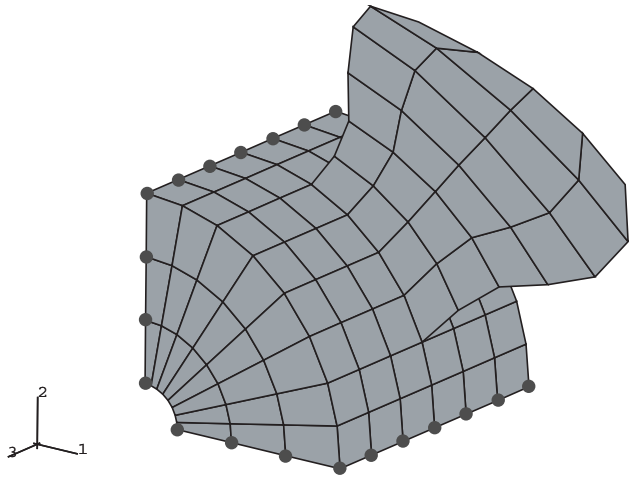


Figure 2.2.1–3 Datum sector for cyclic symmetry model.

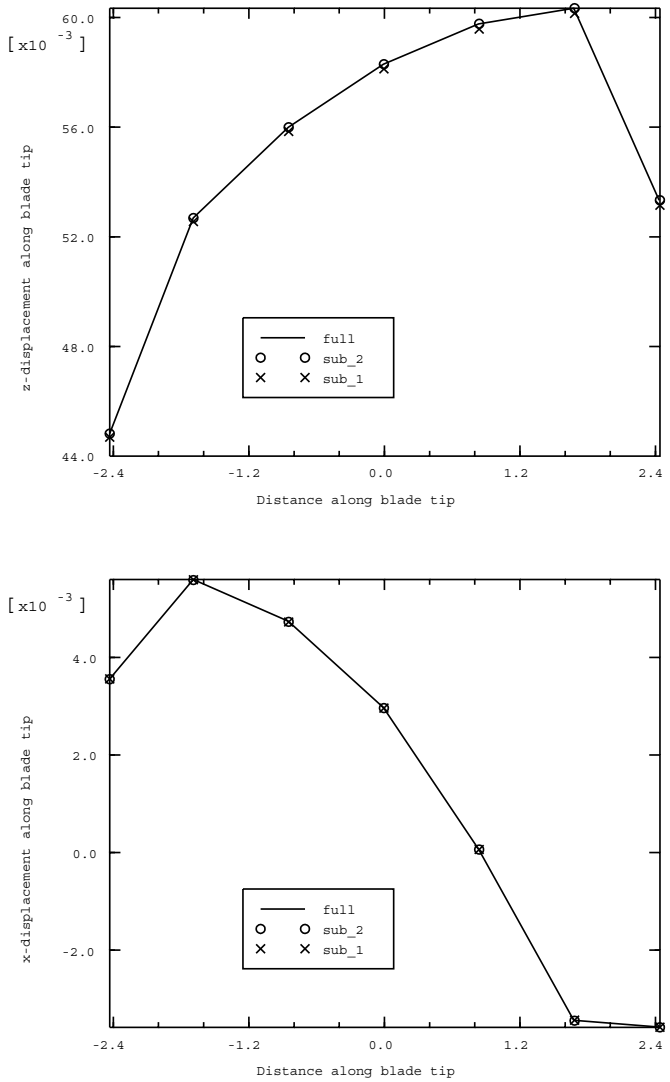


Figure 2.2.1–4 Displacements due to pressure loading along path AB.

2.2.2 LINEAR ANALYSIS OF THE INDIAN POINT REACTOR FEEDWATER LINE

Product: Abaqus/Standard

This example concerns the linear analysis of an actual pipeline from a nuclear reactor and is intended to illustrate some of the issues that must be addressed in performing seismic piping analysis. The pipeline is the Indian Point Boiler Feedwater Pipe fitted with modern supports, as shown in Figure 2.2.2–1. This pipeline was tested experimentally in EPRI's full-scale testing program. The model corresponds to Configuration 1 of the line in Phase III of the testing program. The experimental results are documented in EPRI Report NP–3108 Volume 1 (1983).

We first verify that the geometric/kinematic model is adequate to simulate the dynamic response accurately. For this purpose we compare predictions of the natural frequencies of the system using a coarse model and a finer model, as well as two substructure models created from the coarse mesh. These analyses are intended to verify that the models used in subsequent runs provide accurate predictions of the lower frequencies of the pipeline. We then perform linear dynamic response analysis in the time domain for one of the "snap-back" loadings applied in the physical test (EPRI NP–3108, 1983) and compare the results with the experimental measurements. The linear dynamic response analysis results are also compared with the results of direct integration analysis (integration of all variables in the entire model, as would be performed for a generally nonlinear problem). This is done primarily for cross-verification of the two analysis procedures. These snap-back response analyses correspond to a load of 31136 N (7000 lb) applied at node 25 in the z-direction, with the pipe filled with water. This load case is referred to as test S138R1SZ in EPRI NP–3108.

We also compute the pipeline's response in the frequency domain to steady excitation at node 27 in the z-direction. Experimental data are also available for comparison with these results.

Geometry and model

Geometrical and material properties are taken from EPRI NP–3108 (1983). The supports are assumed to be linear springs for the purpose of these linear analyses, although their actual response is probably nonlinear. The spring stiffness values are those recommended by Tang et al. (1985). The pipe is assumed to be completely restrained in the vertical direction at the wall penetration.

In the experimental snap-back test used for the comparison (test S138R1SZ), the pipe is full of water. The density of the general beam section is, therefore, adjusted to account for the additional mass of the water by computing a composite (steel plus water) mass per unit length of pipe.

The pipeline is modeled with element type B31. This is a shear flexible beam element that uses linear interpolation of displacement and rotation between two nodes, with transverse shear behavior modeled according to Timoshenko beam theory. The element uses a lumped mass matrix because this provides more accurate results in test cases.

The coarse finite element model uses at least two beam elements along each straight run, with a finer division around the curved segments of the pipe to describe the curvature of the pipe with reasonable accuracy. Separate nodes are assigned for all spring supports, external loading locations, and all the points where experimental data have been recorded. The model is shown in Figure 2.2.2–2. This mesh has 74 beam elements.

In typical piping systems the elbows play a dominant role in the response because of their flexibility. This could be incorporated in the model by using the ELBOW elements. However, ELBOW elements are intended for applications that involve nonlinear response within the elbows themselves and are an expensive option for linear response of the elbows, which is the case for this study. Therefore, instead of using elbow elements, we modify the geometrical properties of beam elements to model the elbows with correct flexibility. This is done by calculating the flexibility factor, k, for each elbow and modifying the moments of inertia of the beam cross-sections in these regions. The flexibility factor for an elbow is a function of two parameters. One is a geometric parameter, λ, defined as

$$\lambda = \frac{tR}{r^2\sqrt{1-\nu^2}},$$

where t is the wall thickness of the curved pipe, R is the bend radius of the centerline of the curved pipe, r is the mean cross-sectional radius of the curved pipe, and ν is Poisson's ratio. The other parameter is an internal pressure loading parameter, ψ. For thick sections (like the ones used in this pipe), ψ has negligible effect unless the pressures are very high and the water in this case is not pressurized. Consequently, the flexibility factor is a function of λ only.

For the elbows in this pipeline $\lambda = 0.786$ for the 203 mm (8 in) section and $\lambda = 0.912$ for the 152 mm (6 in) section. The corresponding flexibility factors obtained from Dodge and Moore (1972) are 2.09 and 1.85. These are implemented in the model by modifying the moments of inertia of the beam cross-sections in the curved regions of the pipeline.

Abaqus provides two different options for introducing geometrical properties of a beam cross-section. For general beam sections all geometric properties (area, moments of inertia) can be given without specifying the shape of the cross-section. The material data, including the density, are given on the same option. Alternatively, the geometrical properties of the cross-section can be given by using a beam section. With this option the cross-section dimensions are given, and Abaqus calculates the corresponding cross-sectional behavior by numerical integration, thus allowing for nonlinear material response in the section. When this option is used, the material properties—including density and damping coefficients—are introduced in the material definition associated with the section. This approach is more expensive for systems in which the cross-sectional behavior is linear, since numerical integration over the section is required each time the stress must be computed. Thus, in this case we use a general beam section.

To verify that the mesh will provide results of adequate accuracy, the natural frequencies predicted with this model are compared with those obtained with another mesh that has twice as many elements in each pipe segment. Table 2.2.2–1 shows that these two meshes provide results within 2% for the first six modes and generally quite similar frequencies up to about 30 Hz. Based on this comparison the smaller model, with 74 beam elements, is used for the remaining studies (although the larger model would add little to the cost of the linear analyses, which for either case would be based on the same number of eigenmodes: only in direct integration would the cost increase proportionally with the model size).

Substructure models

In Abaqus the dynamic response of a substructure is defined by a combination of Guyan reduction and the inclusion of some natural modes of the fully restrained substructure. Guyan reduction consists of choosing additional physical degrees of freedom to retain in the dynamic model that are not needed to connect the substructure to the rest of the mesh. In this example we use only Guyan reduction since the model is small and it is easy to identify suitable degrees of freedom to retain. A critical modeling issue with this method is the choice of retained degrees of freedom: enough degrees of freedom must be retained so that the dynamic response of the substructure is modeled with sufficient accuracy. The retained degrees of freedom should be such as to distribute the mass evenly in each substructure so that the lower frequency response of each substructure is modeled accurately. Only frequencies up to 33 Hz are generally considered important in the seismic response of piping systems such as the one studied in this example, so the retained degrees of freedom must be chosen to provide accurate modeling of the response up to that frequency.

In this case the pipeline naturally divides into three segments in terms of which kinematic directions participate in the dynamic response, because the response of a pipeline is generally dominated by transverse displacement. The lower part of the pipeline, between nodes 1 and 23, is, therefore, likely to respond predominantly in degrees of freedom 1 and 2; the middle part, between nodes 23 and 49, should respond in degrees of freedom 2 and 3; and the top part, above node 49, should respond in degrees of freedom 1 and 3. Comparative tests (not documented) have been run to verify these conjectures, and two substructure models have been retained for further analysis: one in which the entire pipeline is treated as a single substructure, and one in which it is split into three substructures. In the latter case all degrees of freedom must be retained at the interface nodes to join the substructures correctly. At other nodes only some translational degrees of freedom are retained, based on the arguments presented above.

The choice of which degrees of freedom to retain can be investigated inexpensively in a case such as this by numerical experiments—extracting the modes of the reduced system for the particular set of retained degrees of freedom and comparing these modes with those of the complete model. The choices made in the substructure models used here are based somewhat on such tests, although insufficient tests have been run to ensure that they are close to the optimal choice for accuracy with a given number of retained variables. For linear analysis of a model as small as this one, achieving an optimal selection of retained degrees of freedom is not critical because computer run times are short: it becomes more critical when the reduced model is used in a nonlinear analysis or where the underlying model is so large that comparative eigenvalue tests cannot be performed easily. In such cases the inclusion of natural modes of the substructure is desirable. The substructure models are shown in Figure 2.2.2–2.

Damping

"Damping" plays an essential role in any practical dynamic analysis. In nonlinear analysis the "damping" is often modeled by introducing dissipation directly into the constitutive definition as viscosity or plasticity. In linear analysis equivalent linear damping is used to approximate dissipation mechanisms that are not modeled explicitly.

Experimental estimates of equivalent linear damping, based on three different methods, are found in EPRI NP–3108 (see Table 7–6, Table 7–7, and Figure 7–15 of that report). For the load case and pipe configuration analyzed here, those results suggest that linear damping corresponding to 2.8% of critical damping in the lowest mode of the system matches the measured behavior of the structure, with the experimental results also showing that the percentage of critical damping changes from mode to mode. In spite of this all the numerical analyses reported here assume the same damping ratio for all modes included in the model, this choice being made for simplicity only.

For linear dynamic analysis based on the eigenmodes, Abaqus allows damping to be defined as a percentage of critical damping in each mode, as structural damping (proportional to nodal forces), or as Rayleigh damping (proportional to the mass and stiffness of the structure). Only the last option is possible when using direct integration, although other forms of damping can be added as discrete dashpots or in the constitutive models. In this case results are obtained for linear dynamic analysis with modal and Rayleigh damping and for direct integration with Rayleigh damping. For linear dynamic analysis based on the eigenmodes, modal damping can be specified using the mode numbers or using the frequency ranges.

Results and discussion

Results are shown for four geometric models: the "coarse" (74 element) model, which has a total of 435 degrees of freedom; a finer (148 element) model, which has a total of 870 degrees of freedom; a model in which the pipeline is modeled as a single substructure (made from the coarse model), with 59 retained degrees of freedom; and a model in which the pipeline is modeled with three substructures (made from the coarse model), with 65 retained degrees of freedom.

The first comparison of results is the natural frequencies of the system, as they are measured and as they are predicted by the various models. The first 24 modes are shown in Table 2.2.2–1. These modes span the frequency range from the lowest frequency (about 4.3 Hz) to about 43 Hz. In typical seismic analysis of systems such as this, the frequency range of practical importance is up to 33 Hz; on this basis these modes are more than sufficient.

Only the first six modes of the actual system have been measured, so any comparison at higher frequencies is between the numerical calculations reported here and other similar computations. The results obtained with the four models correlate quite well between themselves, suggesting that the mesh and the choices of retained degrees of freedom in the substructure models are reasonable. It is particularly noteworthy that the results for the substructure models correspond extremely well with those provided by the original model, considering the large reduction in the number of degrees of freedom for the substructures. The results also correlate roughly with the analysis results obtained by EDS and reported in EPRI NP-3108: except for modes 3 and 4 the frequencies are within 10% of the EDS numbers. For the first three modes the Abaqus results are lower than those reported by EDS. This suggests the possibility that the Abaqus model may be too flexible. The SUPERPIPE values are significantly higher than any of the other data for most modes, and the Abaqus and the EDS results diverge from the test results after the first four modes.

The results of the time history analyses are summarized in Table 2.2.2–2 to Table 2.2.2–5. These analyses are based on using all 24 modes of the coarse model. Typical predicted response plots are shown in Figure 2.2.2–3 to Figure 2.2.2–7. In many cases of regular, beam-type, one-dimensional structures,

the first few modes will generally establish the dynamic behavior. Although the pipeline has an irregular shape, it is worth checking how much the higher modes influence the results. This is done in this case by comparing the results using the first six modes only with the results obtained with 24 modes. The highest discrepancy (20%) is found in the predicted accelerations at certain degrees of freedom. All other results show at most 5–10% differences (see Figure 2.2.2–3 and Figure 2.2.2–4). This conclusion is also supported by the steady-state results.

All the Abaqus results are reasonably self-consistent, in the sense that Rayleigh and modal damping and modal dynamics and direct integration all predict essentially the same values. The choice of 2.8% damping seems reasonable, in that oscillations caused by the snap-back are damped out almost completely in 10 seconds, which corresponds to the measurements.

Unfortunately there is poor correlation between predicted and measured support reactions and maximum recorded displacements. The test results and the corresponding computations are shown in Table 2.2.2–2 and Table 2.2.2–3. All the models give essentially the same values. The initial reactions and displacements are computed for a snap-back load of 31136 N (7000 lb) applied at node 25 (node 417 in EPRI report NP–3108) in the z-direction. The maximum recorded displacements occur at node 27 (node 419 in EPRI report NP–3108) in the y- and z-directions. It is assumed that the supports are in the positions relative to the pipe exactly as shown in Figure 2.2.2–1. The scatter in the experimental measurements makes it difficult to assess the validity of the stiffness chosen for the spring supports. The maximum displacement predicted at node 27 in the z-direction is almost twice that measured. This again implies the possibility that, at least in the area near this node, the model is too flexible.

The generally satisfactory agreement between the natural frequency predictions and poor agreement between the maximum displacements and reactions suggests that improved modeling of the supports may be necessary. In this context it is worthwhile noting that the experimental program recorded significantly different support parameters in different tests on the pipeline system.

Table 2.2.2–4 shows the results for displacement and acceleration for node 27 (which has the largest displacement). All the computed results are higher than the experimental values. The largest discrepancies between the measurements and the analysis results are in the predictions of peak forces in the springs, summarized in Table 2.2.2–5. Results obtained with the various models differ by less than 10%: these differences are caused by the differences in the models, different types of damping, and—for the direct integration results—errors in the time integration (for the modal dynamic procedure the time integration is exact). The principal cause of the discrepancies between the measurements and the computed values is believed to be the assumption of linear response in the springs in the numerical models. In reality the spring supports are either rigid struts or mechanical snubbers (Configuration 2). Especially when snubbers are used, the supports perform as nonlinear elements and must be modeled as such to reflect the support behavior accurately. Interestingly, even with the assumption of linear support behavior, the character of the oscillation is well-predicted for many variables.

The last group of numerical results are frequency domain calculations obtained using the steady-state dynamic analysis. The response corresponds to steady harmonic excitation at node 27 in the z-direction by a force with a peak amplitude of 31136 N (7000 lb). Such frequency domain results play a valuable role in earthquake analysis because they define the frequency ranges in which the structure's response is most amplified by the excitation. Although it is expected that the first few natural frequencies will be where the most amplification occurs, the results show clearly that some variables are strongly

amplified by the fifth and sixth modes. This is observed both in the simulations and in the experimental measurements. Measured experimental results are available for the acceleration of node 33 (node 419 in EPRI NO–3108) in the z-direction and for the force in spring FW-R-21. The character of curves obtained with Abaqus agrees well with the experimental results (see Figure 2.2.2–8 and Figure 2.2.2–9), but the values differ significantly, as in the time domain results. The peak acceleration recorded is 2.0 m/s^2 (78.47 in/s^2), at the first natural frequency, while the analysis predicts 4.0 m/s^2 (157.5 in/s^2). Likewise, the peak force value recorded is 2.0 kN (450 lb), compared to 5.9 kN (1326 lb) predicted. The discrepancies are again attributed to incorrect estimates of the support stiffness or to nonlinearities in the supports.

Input files

indianpoint_modaldyn_coarse.inp	*MODAL DYNAMIC analysis with modal damping using the coarse model.
indianpoint_modaldyn_3sub.inp	*MODAL DYNAMIC analysis using the three substructure model.
indianpoint_3sub_gen1.inp	First substructure generation referenced by the analysis indianpoint_modaldyn_3sub.inp.
indianpoint_3sub_gen2.inp	Second substructure generation referenced by the analysis indianpoint_modaldyn_3sub.inp.
indianpoint_3sub_gen3.inp	Third substructure generation referenced by the analysis indianpoint_modaldyn_3sub.inp.
indianpoint_sstate_sinedwell.inp	*STEADY STATE DYNAMICS analysis corresponding to the sine dwell test performed experimentally using the coarse model.
indianpoint_direct_beam_coarse.inp	Direct integration analysis using the coarse model with the *BEAM SECTION option.
indianpoint_sstate_modaldamp.inp	*STEADY STATE DYNAMICS analysis with modal damping, covering a range of frequencies using the coarse model.
indianpoint_modaldyn_1sub.inp	*MODAL DYNAMIC analysis with one substructure.
indianpoint_1sub_gen1.inp	Substructure generation referenced by the analysis indianpoint_modaldyn_1sub.inp.
indianpoint_direct_beamgensect.inp	Direct integration using the coarse model with *BEAM GENERAL SECTION instead of *BEAM SECTION, which, thus, runs faster on the computer since numerical integration of the cross-section is avoided.
indianpoint_modaldyn_elmatrix1.inp	*MODAL DYNAMIC analysis that reads and uses the substructure matrix written to the results file in indianpoint_3sub_gen1.inp, indianpoint_3sub_gen2.inp, and indianpoint_3sub_gen3.inp.
indianpoint_modaldyn_elmatrix2.inp	Reads and uses the element matrix written to the results file in indianpoint_modaldyn_3sub.inp.

indianpoint_modaldyn_elmatrix3.inp	Reads and uses the substructure matrix written to the results file in indianpoint_1sub_gen1.inp.
indianpoint_modaldyn_elmatrix4.inp	Reads and uses the element matrix written to the results file in indianpoint_modaldyn_1sub.inp.
indianpoint_modaldamp_rayleigh.inp	∗MODAL DAMPING analysis with modal Rayleigh damping using the coarse mesh with the ∗BEAM SECTION option.
indianpoint_dyn_rayleigh_3sub.inp	∗DYNAMIC analysis with Rayleigh damping using the three substructure model.
indianpoint_rayleigh_3sub_gen1.inp	First substructure generation referenced by the analysis indianpoint_dyn_rayleigh_3sub.inp.
indianpoint_rayleigh_3sub_gen2.inp	Second substructure generation referenced by the analysis indianpoint_dyn_rayleigh_3sub.inp.
indianpoint_rayleigh_3sub_gen3.inp	Third substructure generation referenced by the analysis indianpoint_dyn_rayleigh_3sub.inp.
indianpoint_modaldyn_unsorted.inp	One substructure ∗MODAL DYNAMIC analysis with unsorted node sets and unsorted retained degrees of freedom.
indianpoint_unsorted_gen1.inp	Substructure generation with unsorted node sets and unsorted retained degrees of freedom referenced by the analysis indianpoint_modaldyn_unsorted.inp.
indianpoint_lanczos.inp	Same as indianpoint_modaldyn_coarse.inp, except that it uses the Lanczos solver and the eigenvectors are normalized with respect to the generalized mass.
indianpoint_restart_normdisp.inp	Restarts from indianpoint_lanczos.inp and continues the eigenvalue extraction with the eigenvectors normalized with respect to the maximum displacement.
indianpoint_restart_bc.inp	Restarts from indianpoint_lanczos.inp and continues the eigenvalue extraction with modified boundary conditions.
indianpoint_overlapfreq.inp	Contains two steps, which extract eigenvalues with overlapping frequency ranges.

References

- Consolidated Edison Company of New York, Inc., EDS Nuclear, Inc., and Anco Engineers, Inc., *Testing and Analysis of Feedwater Piping at Indian Point Unit 1, Volume 1: Damping and Frequency*, EPRI NP-3108, vol. 1, July 1983.
- Dodge, W. G., and S. E. Moore, “Stress Indices and Flexibility Factors for Moment Loadings in Elbows and Curved Pipe,” WRC Bulletin, no. 179, December 1972.
- Tang, Y. K., M. Gonin, and H. T. Tang, “Correlation Analysis of In-situ Piping Support Reactions,” EPRI correspondence with ABAQUS, May 1985.

Table 2.2.2–1 Comparison of natural frequencies (Hz).

Mode	Anco (experiment)	EDS	SUPER PIPE	Abaqus			
				coarse mesh	finer mesh	single sub	three subs
1	4.20	4.30	5.30	4.25	4.26	4.25	4.25
2	6.80	6.80	8.10	6.27	6.25	6.27	6.27
3	8.30	8.80	12.00	7.29	7.29	7.30	7.30
4	12.60	10.60	13.30	12.80	12.66	12.87	12.86
5	15.40	13.00	14.40	13.18	13.14	13.19	13.20
6	16.70	14.50	15.90	13.90	13.75	13.91	13.92
7		16.20	18.30	15.11	15.98	14.34	14.39
8			19.40	16.30	16.07	16.24	16.31
9			20.20	16.89	16.81	16.43	16.43
10			22.20	17.43	17.82	17.17	17.20
11				18.02	19.07	18.10	18.10
12				19.58	20.10	20.05	20.01
13				23.43	21.45	23.98	24.00
14				23.99	22.13	24.47	24.47
15				24.27	23.58	24.97	24.96
16				24.80	24.15	25.34	25.28
17				26.82	26.84	27.63	27.56
18				29.53	30.18	30.31	30.55
19				30.61	30.60	31.08	31.06
20				30.95	32.58	31.43	31.43
21				31.52	33.11	32.00	31.98
22				33.50	35.08	33.76	33.77
23				39.09	39.65	39.75	39.97
24				39.86	43.25	42.98	42.97

Table 2.2.2–2 Comparison of initial support reactions. Snap-back Test No. S138R1SZ; 31136 N (7000 lb) at node 25, z-direction.

NODE	SUPPORT	Anco TEST		Abaqus	
		N	(lb)	N	(lb)
15	FW-R-11	−8000	(−1798.6)	−11712	(−2633)
22	FW-R-13	30000	(6744.6)	29352	(6599)
23	FW-R-14	−252	(−56.7)	−3754	(−844)
35	FW-R-17	23625	(5311.4)	102	(−22.8)
35	FW-R-18	10025	(2553.8)	−18468	(−4152)
39	FW-R-20	24000	(5395.7)	4212	(947)
39	FW-R-21	−24500	(−5508.1)	25016	(5624)
49	FW-R-23	8000	(1798.6)	24348	(5474)
53	FW-R-24	−4324	(−972.1)	−4057	(−912)
53	FW-R-25	2000	(449.6)	−816	(−183)
56	FW-R-27	432	(97.1)	−1801	(−405)
56	FW-R-28	156	(35.1)	−799	(−180)

Table 2.2.2–3 Comparison of maximum displacements.

Abaqus NODE No.	Anco NODE No.	Measured		Abaqus	
		mm	(in)	mm	(in)
27	419-Y	−16.0	(−.630)	−26.85	(−1.057)
27	419-Z	37.81	(1.49)	65.72	(2.587)

Table 2.2.2–4 Peak displacement and acceleration values at node 27.

Variable	Measured (Anco)	Abaqus		
		Modal, 2.8% modal damping	Modal, Rayleigh damping	Direct integration
u_y (mm)	−0.024/0.024	−0.029/0.029	−0.031/0.031	−0.031/0.031
u_z (mm)	−0.038/0.038	−0.058/0.066	−0.062/0.059	−0.063/0.068
$\ddot{u}_z$ (m/s^2)	−47.6/40.9	−42.1/50.8	−49.6/49.9	−83.8/91.0
The high acceleration amplitude reported for the Abaqus direct integration analysis occurs only during the first few increments, after which it reduces to −31.6/48.6 m/s^2.				

Table 2.2.2–5 Peak reaction forces at supports (in kN).

Support number	Measured (Anco)	Abaqus		
		Modal, 2.8% modal damping	Modal, Rayleigh damping	Direct integration
FW-R-11	−16.44/19.22	−19.80/13.42	−19.90/14.26	−21.82/15.76
FW-R-13	−15.10/29.91	−18.94/24.45	−19.46/23.61	−28.50/21.98
FW-R-14	−7.22/12.00	−9.34/7.35	−10.23/10.00	−12.54/9.03
FW-R-17	34.40/26.20	−7.50/10.59	−.17/9.25	−7.91/10.97
FW-R-18	−14.30/14.40	−33.26/32.06	−33.58/31.61	−33.46/32.63
FW-R-20	−25.60/26.90	−7.54/8.79	−7.98/8.50	−8.07/10.60
FW-R-21	−24.50/23.80	−25.55/24.47	−26.38/25.30	−27.78/25.26
FW-R-23	−15.30/16.00	−25.39/24.63	−26.06/25.36	−25.40/24.35
FW-R-24	−9.61/7.30	−7.17/6.87	−7.69/7.20	−8.23/8.64
FW-R-25	−6.77/6.21	−3.48/4.36	−3.34/4.55	−7.13/4.71
FW-R-27	−3.76/3.04	−4.12/4.00	−3.78/3.80	−4.29/4.43
FW-R-28	−1.10/1.82	−1.53/1.08	−1.62/1.15	−1.79/1.44

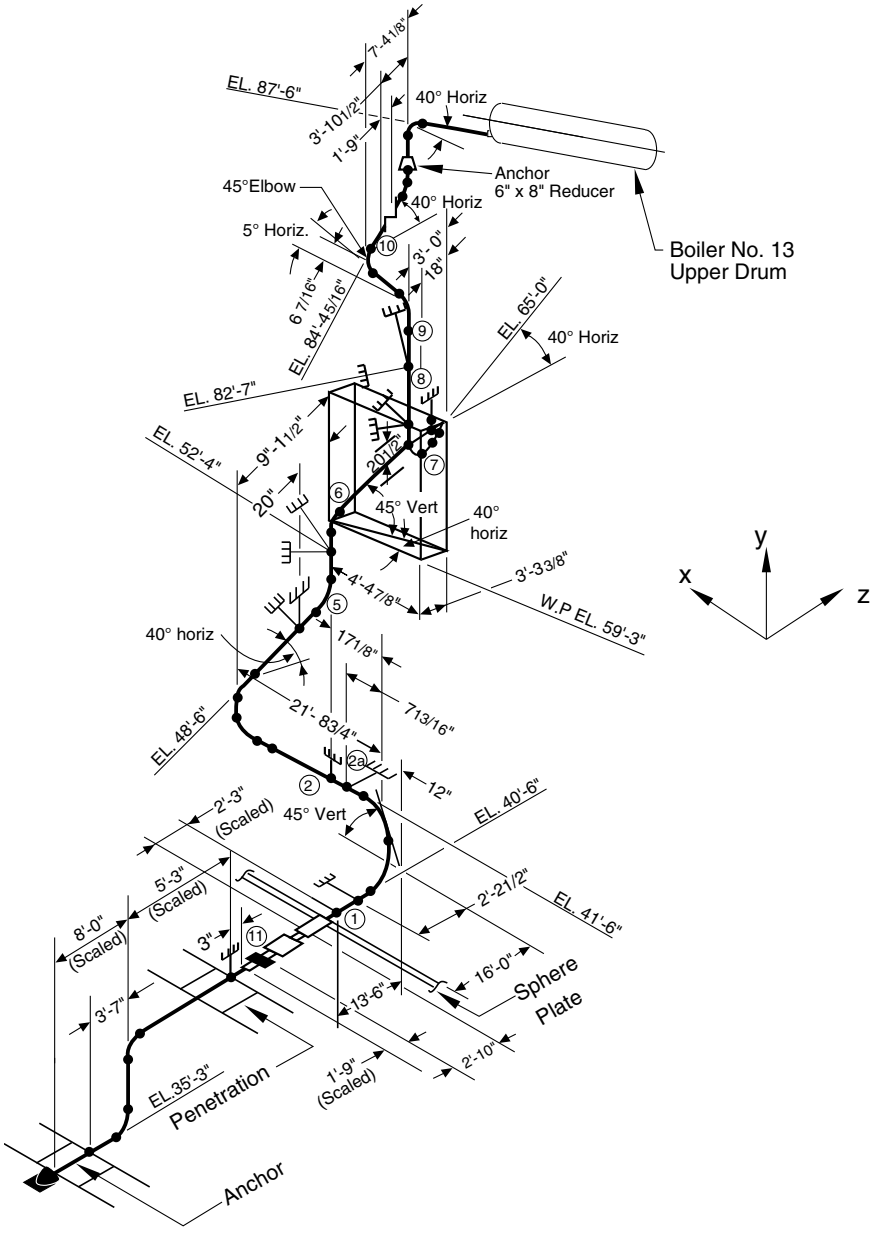


Figure 2.2.2–1 Indian Point boiler feedwater line: modern supports, Configuration 1.

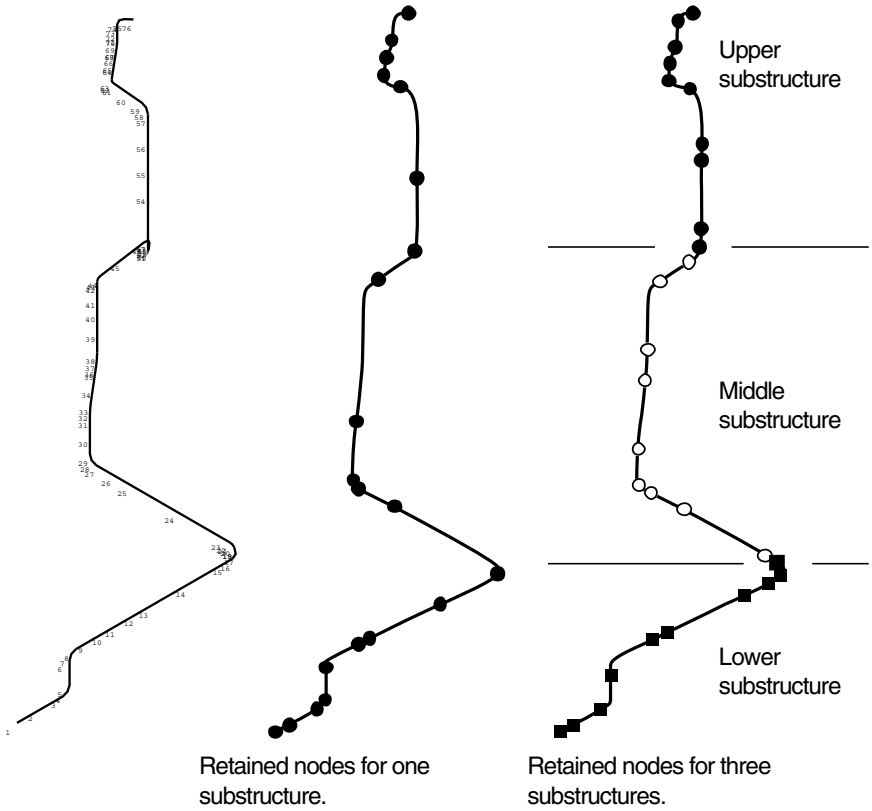


Figure 2.2.2–2 Basic mesh and substructure models.

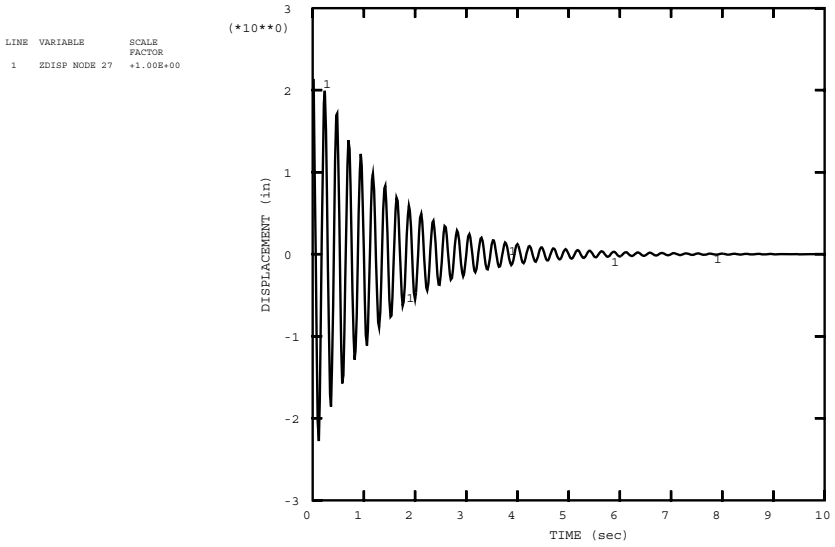


Figure 2.2.2–3 *z*-displacement at node 27, modal analysis with 24 modes.

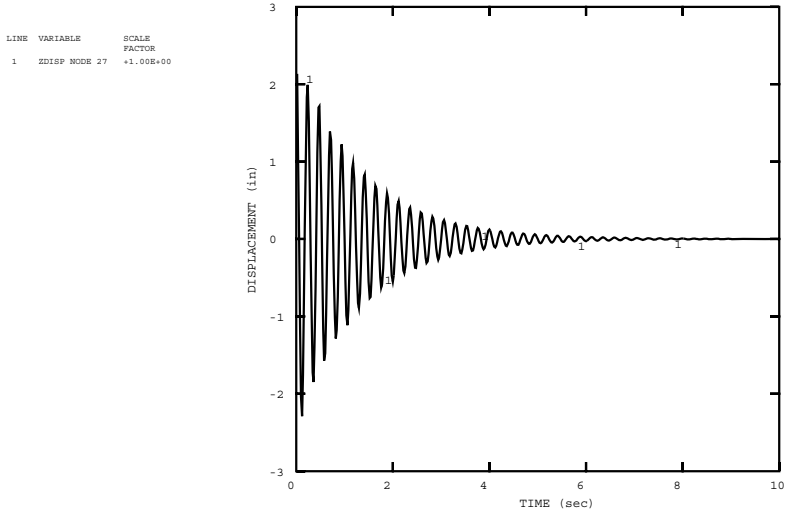


Figure 2.2.2–4 *z*-displacement at node 27, modal analysis with 6 modes.

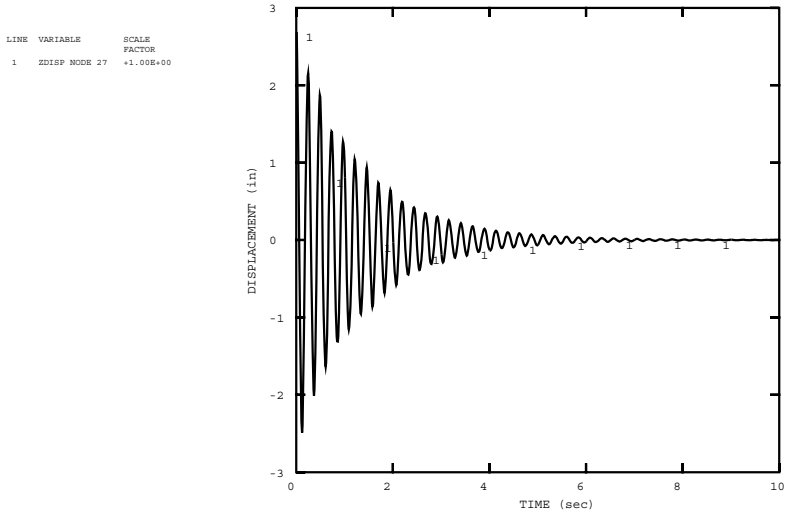


Figure 2.2.2–5 *z*-displacement at node 27, direct integration analysis.

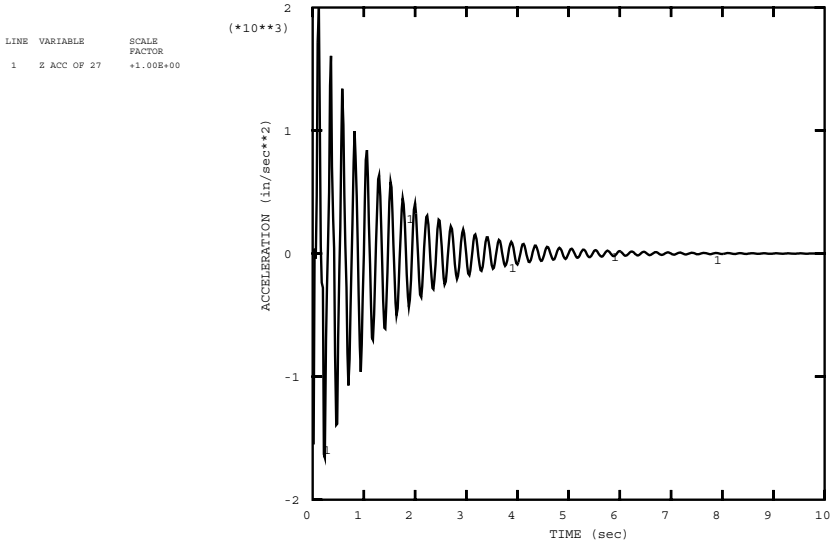


Figure 2.2.2–6 *z*-direction acceleration at node 27, modal analysis with 24 modes.

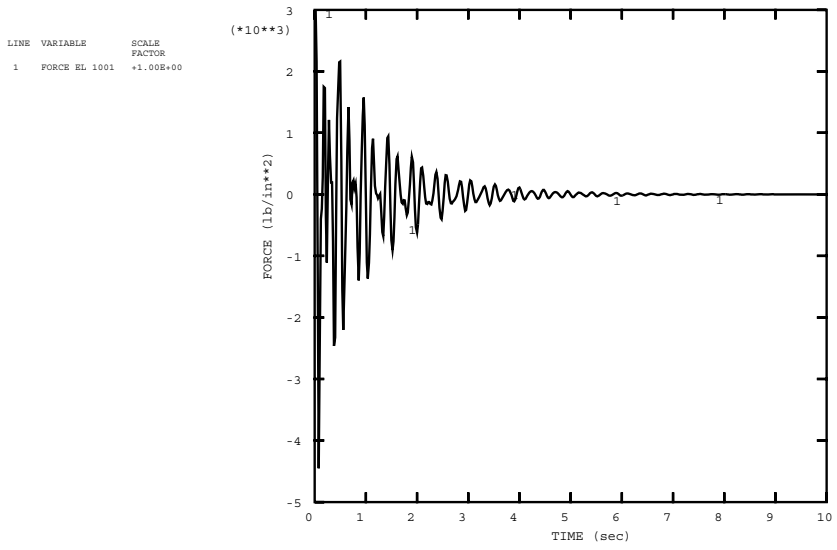


Figure 2.2.2–7 Force in spring support FW–R–11, modal analysis with 24 modes.

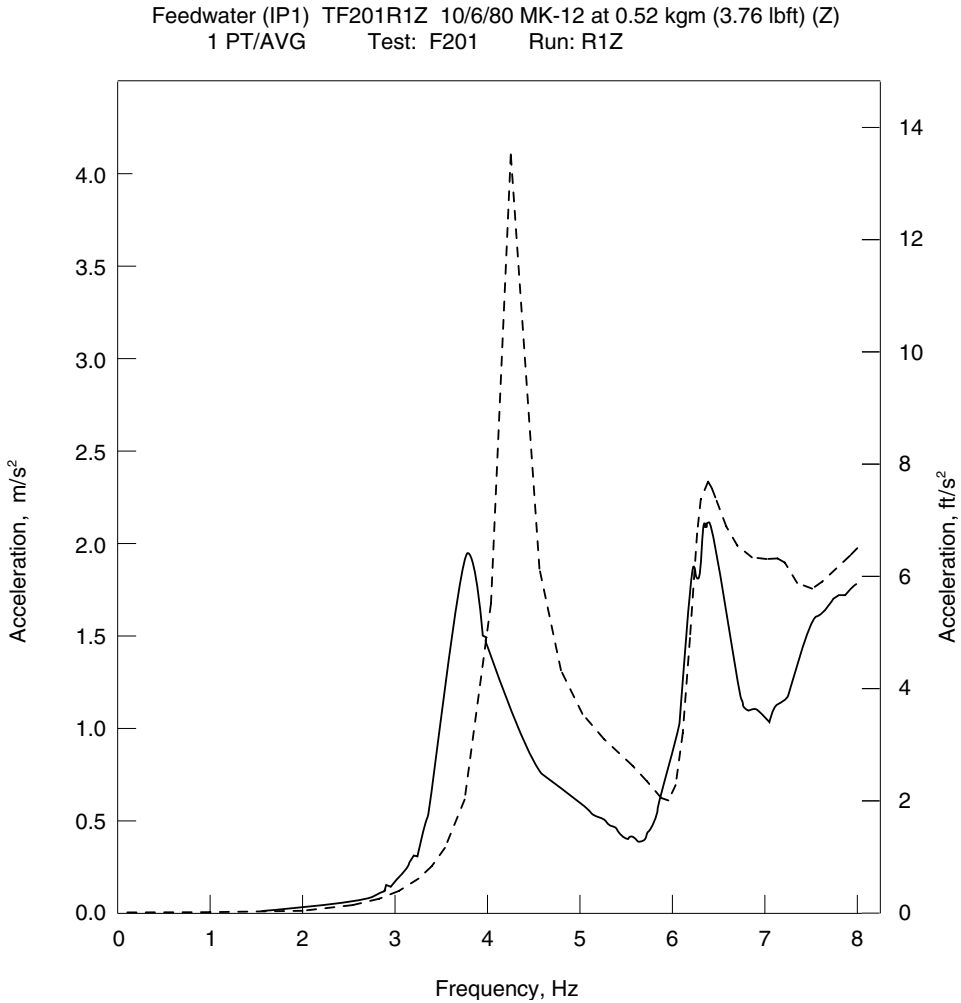


Figure 2.2.2–8 Comparison of *z*-direction acceleration at node 33 between experimental steady-state results (solid line) and Abaqus (dashed line).

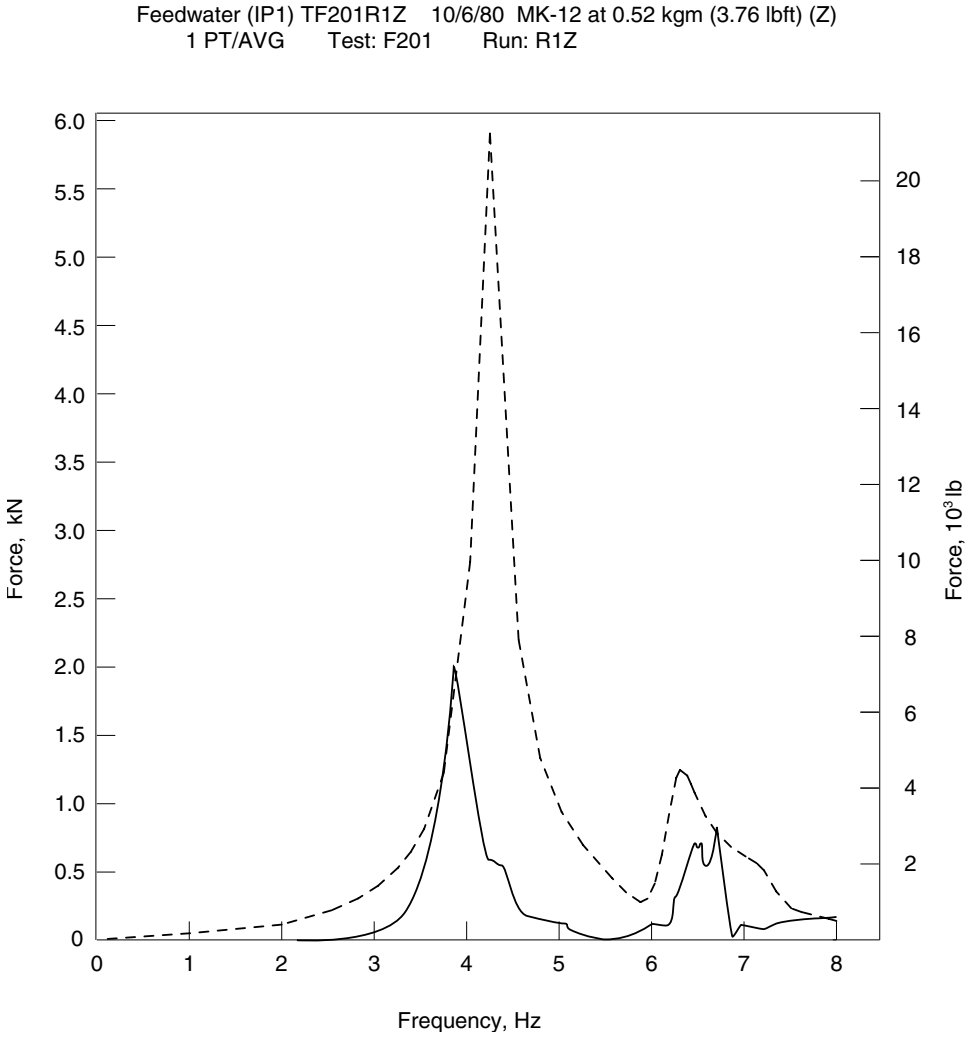


Figure 2.2.2–9 Comparison of force in spring support FW–R–21 between experimental steady-state results (solid line) and Abaqus (dashed line).

2.2.3 RESPONSE SPECTRA OF A THREE-DIMENSIONAL FRAME BUILDING

Product: Abaqus/Standard

The purpose of this example is to verify the different summation methods for natural modes in the response spectrum procedure. To compare the five different methods that are available in Abaqus, a three-dimensional model with closely spaced modes is examined.

Geometry and model

A four-story steel-frame building is analyzed. All columns in the building have the same geometric properties. However, as shown in Figure 2.2.3–1, the properties of the beams in Frames 1 and 2 are different, as compared to those in Frames 3 and 4, to move the center of mass of the structure away from its geometric center. Eigenvalue extraction performed on the model shows that many of the 30 modes that cover the frequency range up to 40 Hz are closely spaced. An acceleration spectrum based on the El Centro earthquake record is applied in the x–y plane. The FORTRAN program given in frameresponsespect_acc.f is used to generate the spectrum. The frequency range is chosen between 0.1 Hz and 40 Hz, and the number of points at which the spectrum is calculated is set at 501. Only one spectrum curve is requested for 2% damping.

Results and discussion

As described in "Linear analysis of a rod under dynamic loading," Section 1.4.9 of the Abaqus Benchmarks Guide, for structures with well-separated modes the TENP and the CQC methods reduce to the SRSS method, while the NRL and the ABS methods give similar results. Hence, for such structures, two summation rules would suffice, with ABS providing the more conservative results. However, when structures with closely spaced modes are analyzed, all five summation rules can yield very different results. This is even more apparent in three-dimensional problems. In the present example, the plane of the earthquake motion lies along the x-axis, so we expect that the structural response will be dominated by Frames 1 and 3 and will result in a significant base shear in the x-direction. All five methods are compared against a modal time history response using the same El Centro acceleration record in Table 2.2.3–1, where the base shear forces are summed up in the plane of each frame S_i, where i is the frame number. This comparison shows that the best approximation is generated by the CQC method. The other methods overestimate the shear in the y-direction, and some of them underestimate the base shear in the x-direction. The CQC method is generally recommended for asymmetrical three-dimensional problems with closely spaced structural modes. This method takes into account the sign of the mode shapes through cross-modal correlation factors and can correctly predict the response in directions perpendicular to the direction of excitation.

Input files

frameresponsespect_freq.inp	*FREQUENCY analysis.
frameresponsespect_rs.inp	*RESPONSE SPECTRUM analysis.

frameresponsespect_modal.inp	∗MODAL DYNAMIC analysis. To run this file, the user must fetch the file cantilever_quakedata.inp and copy it to **QUAKE.AMP**.
frameresponsespect_acc.f	FORTRAN program that will produce the acceleration spectrum needed to run frameresponsespect_rs.inp. To run this program, the user must fetch the file cantilever_quakedata.inp and copy it to **QUAKE.AMP**.

Table 2.2.3–1 Comparison of base shear forces for different summation methods.

Method	S_1 **(kip)**	S_2 **(kip)**	S_3 **(kip)**	S_4 **(kip)**
Time history	−25.5	14.0	−37.0	−22.8
ABS/ALG	48.3	48.3	65.1	65.1
SRSS/ALG	18.4	18.4	24.8	24.8
TENP/ALG	28.9	28.9	35.9	35.9
NRL/ALG	25.9	25.9	34.6	34.6
CQC/ALG	23.3	13.0	29.0	20.7

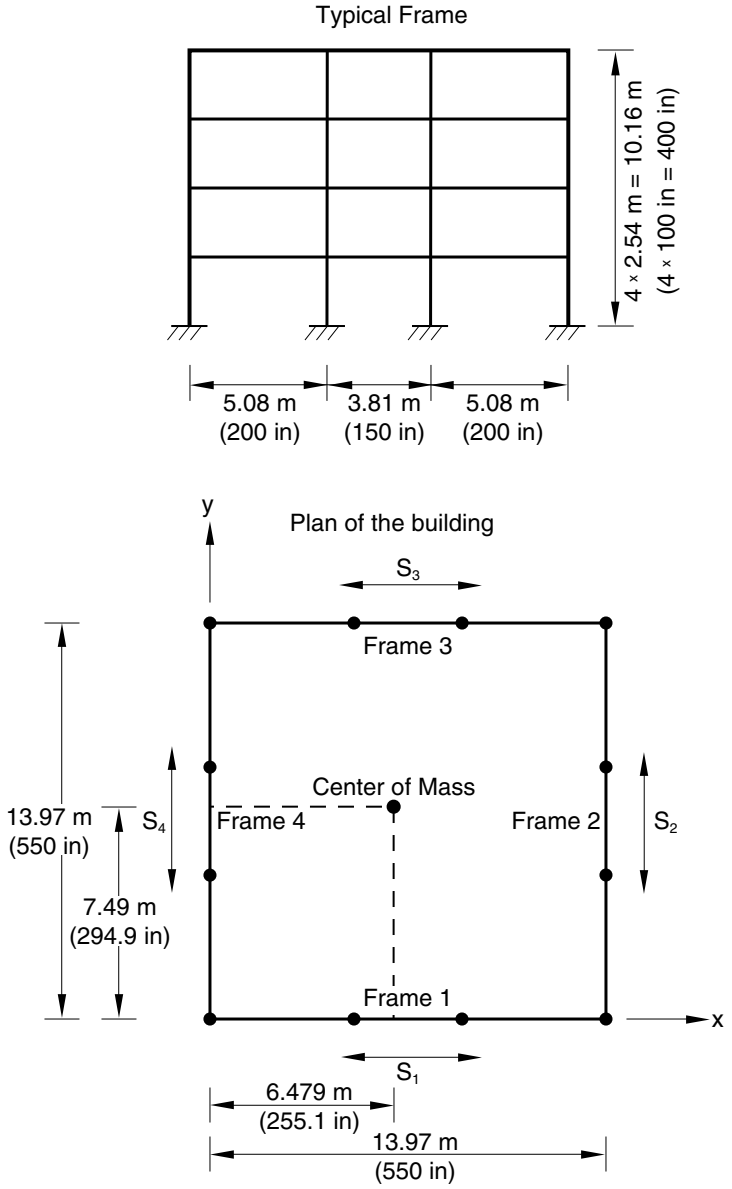


Figure 2.2.3–1 Three-dimensional frame system.

2.2.4 BRAKE SQUEAL ANALYSIS

Product: Abaqus/Standard

This example illustrates the use of the complex eigenvalue extraction procedure ("Complex eigenvalue extraction," Section 6.3.6 of the Abaqus Analysis User's Guide) in a brake squeal analysis. Disc brakes operate by pressing a set of brake pads against a rotating disc. The friction between the pads and the disc causes deceleration, but it may also induce a dynamic instability of the system, known as brake squeal. One possible explanation for the brake squeal phenomenon is the coupling of two neighboring modes. Two modes, which are close to each other in the frequency range and have similar characteristics, may merge as the friction contribution increases. When these modes merge at the same frequency (become coupled), one of them becomes unstable. The unstable mode can be identified during complex eigenvalue extraction because the real part of the eigenvalue corresponding to an unstable mode is positive. The brake system design can be stabilized by changing the geometry or material properties of the brake components to decouple the modes.

The purpose of this analysis is to identify the unstable modes (if they exist) in a particular disc brake system.

Problem description and model definition

The brake model used in this example is a simplified version of a disc brake system used in domestic passenger vehicles. The simplified model consists of a rotor and two pads positioned on both sides of the rotor. The pads are made of an organic friction material, which is modeled as an anisotropic elastic material. The rotor has a diameter of 288 mm and a thickness of 20 mm and is made of cast iron. The back plates and insulators are positioned behind the pads and are made of steel. In this problem material damping is ignored. The mesh (shown in Figure 2.2.4–1) is generated using C3D6 and C3D8I elements. Contact is defined between both sides of the rotor and the pads using the small-sliding contact formulation. Initially, the friction coefficient is set to zero.

Contact between the rotor and the pads is established initially in the first step by applying pressure to the external surfaces of the insulators. In the next step a rotational velocity of ω=5 rad/s is imposed on the rotor using the prescribed rotational motion. The imposed velocity corresponds to braking at low velocity. The friction coefficient, μ, is also increased to 0.3 using a change to friction properties. In general, the friction coefficient can depend on the slip rate, contact pressure, and temperature. If the friction coefficient depends on the slip rate, the velocity imposed by the motion predefined field is used to determine the corresponding value of μ. The friction coefficient is ramped from zero up to the desired value to avoid the discontinuities and convergence problems that may arise because of the change in friction coefficient that typically occurs when bodies in contact are moving with respect to each other. This issue is described in detail in "Static stress analysis," Section 6.2.2 of the Abaqus Analysis User's Guide. The unsymmetric solver is used in this step. At the end of the step a steady-state braking condition is obtained.

In the next step the eigenvalue extraction procedure is performed in this steady-state condition. Because the complex eigensolver uses the subspace projection technique, the real eigenvectors are extracted first to define the projection subspace. In the eigenvalue extraction procedure the tangential

degrees of freedom are not constrained at the contact nodes at which a velocity differential is defined. One hundred real eigenmodes are extracted and, by default, all of them are used to define the projection subspace. The subspace can be reduced by selecting the eigenmodes to be used in the complex eigenvalue extraction step. The complex eigenvalue analysis is performed up to 10 kHz (the first 55 modes).

Results and discussion

Figure 2.2.4–2 shows the damping ratio as a function of frequency. The damping ratio is defined as $-2 * Re(\lambda)/|Im(\lambda)|$, where λ is a complex eigenvalue. A negative value of the damping ratio indicates an unstable mode. In the range of interest the complex eigensolver found an unstable mode at the frequency 2.0 kHz with a damping ratio of −0.0138. Figure 2.2.4–3 presents the unstable mode as the combined magnitude of both the real and imaginary components of the complex eigenvector.

In the second analysis the value of the friction coefficient between the rotor and the pads is increased to 0.5. Because the real eigenvectors do not differ significantly between the problems with different friction coefficients, the eigenspace determined in the first analysis can be reused as the projection subspace for the second complex eigenvalue extraction procedure. The analysis is restarted after the frequency extraction step of the preceding run, and two new procedure steps are defined. In the first nonlinear static step the friction coefficient is increased by changing friction properties, followed by the complex eigenvalue extraction in the second step. The results are presented in the form of a damping ratio plot, shown in Figure 2.2.4–4. In this case four unstable modes at 2.0, 3.0, 8.0, and 9.0 kHz are found. As the friction coefficient is increased, more neighboring modes couple and become unstable.

Acknowledgments

SIMULIA would like to thank Dr. Li Jun Zeng of TRW Automotive for supplying the disc brake model used in this example.

Input files

brake_squeal.inp	The initial brake squeal analysis with a friction coefficient μ=0.3.
brake_squeal_node.inp	Nodal coordinates for the brake model.
brake_squeal_elem.inp	Element definitions for the brake model.
brake_squeal_res.inp	The restarted brake squeal analysis with a friction coefficient μ=0.5.

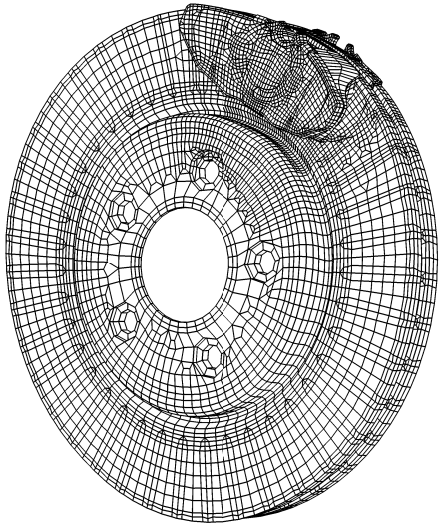


Figure 2.2.4–1 Geometry and mesh of the disc brake system.

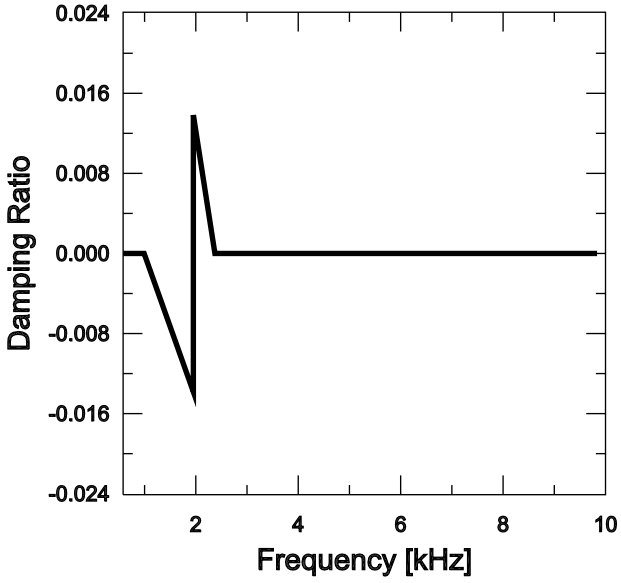


Figure 2.2.4–2 Damping ratios for the analysis with μ=0.3.

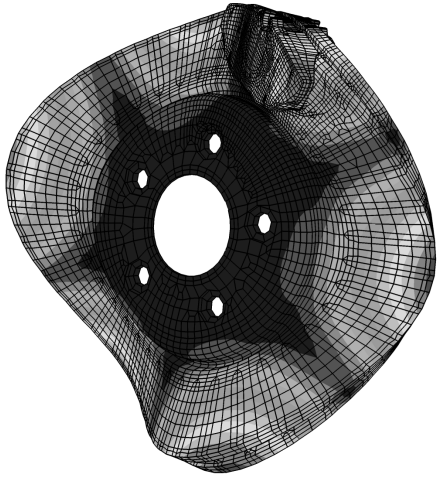


Figure 2.2.4–3 The unstable mode at 2.0 kHz in the analysis with μ=0.3.

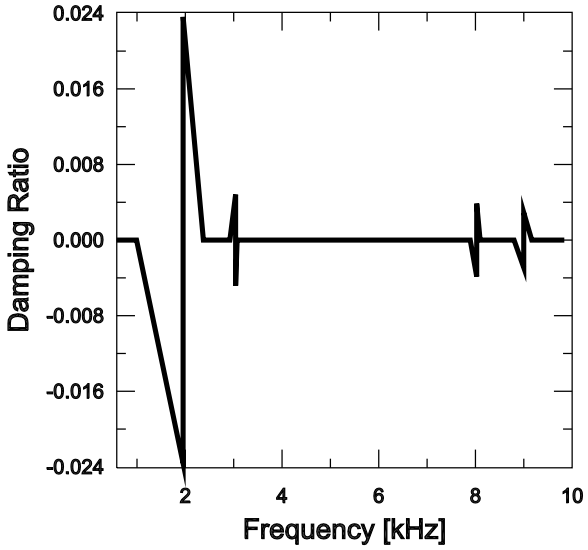


Figure 2.2.4–4 Damping ratios for the analysis with μ=0.5.

2.2.5 DYNAMIC ANALYSIS OF ANTENNA STRUCTURE UTILIZING RESIDUAL MODES

Product: Abaqus/Standard

This example illustrates the use of residual modes in a modal dynamic analysis. The residual modes capability is a cost-effective approach to correct inaccuracies due to modal truncation that are often prevalent in modal dynamic analyses. For performance reasons, usually only a relatively small subset of the total possible eigenmodes is extracted for the model. This set of eigenmodes typically is chosen to cover adequately the frequency content of the applied loads in the modal dynamic analysis. However, this criterion alone does not guarantee that the system will be represented adequately by this set of eigenmodes; consequently, inaccuracies can occur in the modal dynamic solution. Fortunately, adding a few residual modes to the set of eigenmodes can significantly improve the solution at a relatively low cost compared to extracting additional eigenmodes arbitrarily.

Geometry and problem description

The model used for this study is a relatively simple antenna structure that has shell elements for the dish with beam elements for the support, as shown in Figure 2.2.5–1. The objective is to calculate the total force in the main support beam due to a force dynamically applied to the antenna. The bottom of the antenna is completely fixed to the ground. The force is applied in the vertical direction as a step function of magnitude 1.78 lbf, as shown in Figure 2.2.5–2.

Models

Four models are created to demonstrate modal truncation inaccuracies and comparisons of the results using residual modes with a relatively small number of eigenmodes to the results using no residual modes but many eigenmodes.

The finite element discretization and excitation environment are the same in all models. The only difference between the models is the number of eigenmodes extracted and subsequently used in the modal dynamic analysis. More specifically, the first three models use 10, 20, and 50 eigenmodes, respectively; and the fourth model uses 10 eigenmodes in addition to a single residual mode.

Procedure

Following the usual procedure for modal dynamics, the eigenmodes are first extracted in a frequency step. This step is followed by a modal dynamics step in which the forcing excitation is applied.

To activate the residual mode capability, a static perturbation step must precede the frequency extraction step. The same loading pattern that is given in the modal dynamics step must be applied in the static perturbation step. In addition, residual modes must be extracted in the frequency step. The modal dynamics step remains the same as in the case without residual modes.

Results and discussion

Since the magnitude of the applied load is 1.78 lbf, the static solution for the total axial force in the main support beam is also 1.78 lbf. When the modal dynamic analysis is performed, modal truncation inaccuracies become evident. Figure 2.2.5–3 shows the force results using 10, 20, and 50 eigenmodes. For the 10-eigenmode case, the peak force in the support beam is only 0.4 lbf. If 20 eigenmodes are included, the peak force jumps to 1.7 lbf. If 50 eigenmodes are used, the peak force is 2.5 lbf and the correct static response of 1.78 lbf is recovered.

The answers vary because of modal truncation. The excitation history is a step function and, thus, includes the full span of frequency content. Similarly, since the excitation is applied at a single point, it can be accurately represented in the modal analysis only by using a large span of eigenmodes. Since a very large number of eigenmodes are potentially excited due to the temporal and spatial characteristics of the excitation, it is necessary to extract and use many eigenmodes. It is possible to determine how many eigenmodes are needed by looking at the total effective mass of the extracted modes or by performing a convergence study by systematically increasing the number of eigenmodes used in the modal dynamics analysis, similar to the study conducted above.

A much less time-consuming approach (in terms of human intervention and analysis cost) for achieving more accurate results is to use the residual modes capability. Due to the nature of the loading in this analysis, the residual mode capability is particularly useful in capturing the mode shape that dominates the static response. Figure 2.2.5–4 shows that a single residual mode combined with only 10 eigenmodes yields the correct solution. The residual modes approach (with a total of only 11 modes) produces essentially the same accuracy as using 50 pure eigenmodes.

Input files

antenna_10.inp	Modal dynamic analysis of antenna structure using 10 system eigenmodes.
antenna_20.inp	Modal dynamic analysis of antenna structure using 20 system eigenmodes.
antenna_50.inp	Modal dynamic analysis of antenna structure using 50 system eigenmodes.
antenna_10_resvec.inp	Modal dynamic analysis of antenna structure using 10 eigenmodes plus residual modes.

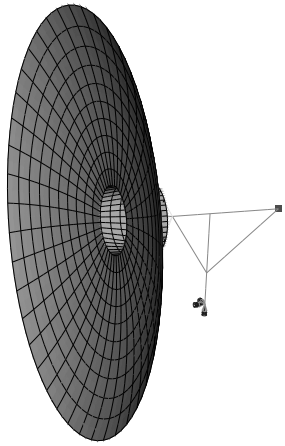


Figure 2.2.5–1 Antenna model.

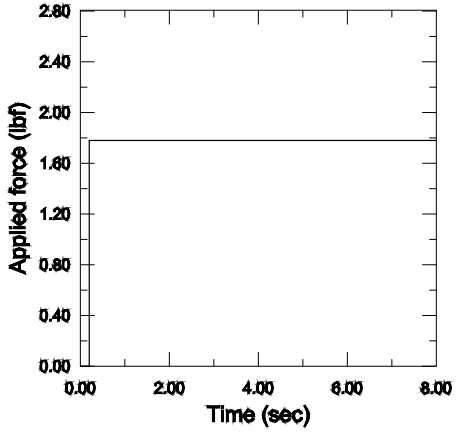


Figure 2.2.5–2 Excitation history.

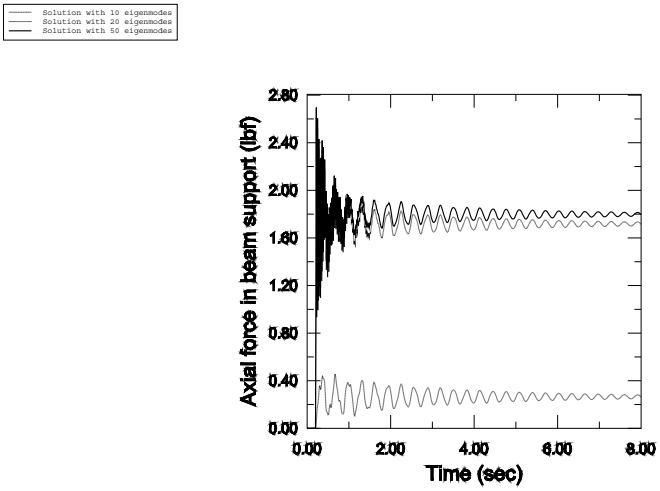


Figure 2.2.5–3 Axial force in main support beam due to dynamic excitation using 10, 20, and 50 eigenmodes.

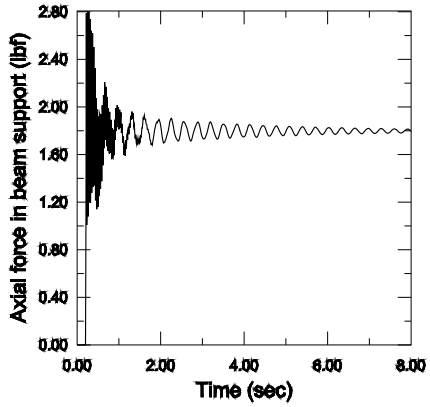


Figure 2.2.5–4 Axial force in main support beam due to dynamic excitation using 10 eigenmodes and residual modes.

2.2.6 STEADY-STATE DYNAMIC ANALYSIS OF A VEHICLE BODY-IN-WHITE MODEL

Products: Abaqus/Standard Abaqus/AMS

Objectives

This example demonstrates the following Abaqus features and techniques for frequency extraction and steady-state dynamic analysis:

- using the automatic multi-level substructuring (AMS) eigensolver in the frequency extraction step along with residual modes;
- projecting a global material structural damping operator during the frequency extraction step using the AMS eigensolver;
- using the SIM-based steady-state dynamic analysis procedure with material structural damping; and
- demonstrating the performance benefit of the SIM-based steady-state dynamic analysis procedure using the AMS eigensolver compared to the subspace-based steady-state dynamic analysis procedure with the Lanczos eigensolver.

Application description

This example examines the structural behavior of a vehicle body-in-white (BIW) model in terms of eigenmodes and frequency response functions. In addition, this example demonstrates the use of the steady-state dynamic analysis procedure with the AMS eigensolver and the SIM architecture (see "Using the SIM architecture for modal superposition dynamic analyses" in "Dynamic analysis procedures: overview," Section 6.3.1 of the Abaqus Analysis User's Guide) for an automobile structure. The model shown in Figure 2.2.6–1 was obtained from the National Highway Traffic Safety Administration (NHTSA) web site (http://www-nrd.nhtsa.dot.gov/departments/nrd-11/FEA_models.html). This model includes material structural damping.

The primary goal of this example is to demonstrate the significant performance improvement of the SIM-based steady-state dynamic analysis procedure using the AMS eigensolver compared to the subspace-based steady-state dynamic analysis procedure. Prior to performing the steady-state dynamic analysis, the undamped eigensolution is computed using the AMS eigensolver. The global cutoff frequency of this model is 300 Hz, so the global eigenmodes below 300 Hz are extracted. During the reduction phase of the AMS eigensolver, all of the substructure eigenmodes below 1500 Hz are extracted and used to calculate the global eigensolution with the AMS_{cutoff_1} default value of 5. In addition, the material structural damping operator is projected onto the global eigenmode subspace.

Geometry

The model consists of the bare metal shell of the frame body including fixed windshields. This model has 127,213 elements and 794,292 active degrees of freedom, and a total of 1107 fasteners are defined to model spot welds in the frame body.

Materials

Linear elastic materials are used for all shell elements, and material structural damping with a value of 0.01 is applied to the materials for the ceiling, floor, hood, and side body of the vehicle. This type of damping results in a full system of modal equations that must be solved for each frequency point.

Boundary conditions and loading

This structure is not constrained, so there are six rigid body modes in the model. Two concentrated loads are applied to the nodes at the two pivot points on the bottom of the vehicle floor to simulate the rolling motion of the vehicle in the steady-state dynamic analysis.

Abaqus modeling approaches and simulation techniques

The objective of this analysis is an understanding of the overall structural response of a body-in-white model due to rolling motion. The response is evaluated by studying the frequency responses at both a drive point node and a transfer point node.

Compared to the subspace-based steady-state dynamic analysis procedure with the Lanczos eigensolver, the SIM-based steady-state dynamic analysis procedure using the AMS eigensolver shows significant performance benefit with acceptable accuracy. The accuracy of the SIM-based steady-state dynamic solution is evaluated by comparing the frequency response functions for the SIM-based and subspace-based steady-state dynamic analyses.

Drive point and transfer point responses are calculated at nodes 840950 and 874962, respectively, to demonstrate the accuracy of the SIM-based steady-state dynamic analysis procedure with eigenmodes computed by the AMS eigensolver. This model has 1210 global eigenmodes below the global cutoff frequency, including 6 rigid body modes. The AMS eigensolver approximates 1188 global eigenmodes below the global cutoff frequency of 300 Hz, and the frequency response solutions below 150 Hz are calculated at 1 Hz increments. Two residual modes are also added to demonstrate a means of improving efficiency by using fewer global modes.

Summary of analysis cases

Case 1	SIM-based steady-state dynamic analysis using AMS.
Case 2	Subspace-based steady-state dynamic analysis using Lanczos.
Case 3	Direct-solution steady-state dynamic analysis.

Analysis types

The frequency extraction step is followed by a steady-state dynamic analysis step. The global cutoff frequency for the frequency extraction step is 300 Hz, and the frequency range considered is 1–150 Hz at 1 Hz increments. For the subspace-based steady-state dynamic analysis, 1000 eigenmodes below 300 Hz are used. For the SIM-based steady-state dynamic analysis, all the eigenmodes below 300 Hz (approximately 1200 eigenmodes) are used.

Discussion of results and comparison of cases

The accuracy of the natural frequencies computed by the AMS eigensolver is evaluated assuming that the natural frequencies computed by the Lanczos eigensolver are exact, and relative errors are shown in Figure 2.2.6–2. These errors are less than 1.6% at frequencies below the global cutoff frequency of 300 Hz. Below 150 Hz, the region of particular interest, the errors are less than 0.25% and the resonance peaks in the frequency response function are reasonably accurate.

The accuracy of the steady-state dynamics solutions is evaluated qualitatively by comparing the frequency response curves. The response curves are compared at two different output locations: a drive point node and a transfer point node.

Figure 2.2.6–3 shows the point inertance at node 840950; the curve labeled **SSD DIRECT** represents the direct-solution steady-state dynamic analysis procedure, the curve labeled **SSD SP** represents the subspace-based steady-state dynamic analysis procedure, the curve labeled **SSD AMS** represents the SIM-based steady-state dynamic analysis procedure, and the curve labeled **SSD AMS RES** represents the SIM-based steady-state dynamic analysis procedure with residual modes. The curves are almost indistinguishable for frequencies below 125 Hz. To improve the accuracy of the response functions, residual modes can be added to compensate for high-frequency truncation errors. Calculation of residual modes is requested in the AMS eigenvalue extraction step by specifying the nodal points where forces are applied. For this solution (**SSD AMS RES**), the global cutoff frequency is set to 225 Hz instead of 300 Hz to demonstrate that the global cutoff frequency can be lowered without losing accuracy. By lowering the global cutoff frequency to 225 Hz and adding two residual modes in the global eigenmode subspace, a speed up of approximately 25% can be achieved.

Figure 2.2.6–4 shows the transfer inertance at the node located at the bottom of the driver's seat; the same curve labels discussed above are used in this figure. The frequency response computed by the SIM-based steady-state dynamic analysis procedure nearly matches those computed by the subspace-based steady-state dynamic analysis procedure and the direct-solution steady-state dynamic analysis procedure for the frequencies below 125 Hz. To improve accuracy for frequencies higher than 125 Hz, more eigenmodes should be extracted and used.

The overall performance comparison among the three steady-state dynamic analysis procedures is summarized in Table 2.2.6–1 (analysis cost is independent of whether or not residual modes are used). Performance was evaluated on a Linux/x86-64 server with a single processor. The amount of available memory was set to 16 GB for each run. It is clear that the SIM-based steady-state dynamic analysis procedure using the AMS eigensolver shows excellent performance compared to the subspace-based steady-state dynamic analysis procedure using the Lanczos eigensolver or the direct-solution steady-state dynamic analysis procedure.

Input files

biw_modeldata.inp	Input file for the BIW model data.
biw_freq_ams.inp	Frequency extraction analysis using the AMS eigensolver for the BIW model.

biw_ssd_ams.inp	SIM-based steady-state dynamic analysis using the AMS eigensolution for the BIW model.
biw_freq_ams_res.inp	Frequency extraction analysis (including residual modes) using the AMS eigensolver for the BIW model.
biw_ssd_ams_res.inp	SIM-based steady-state dynamic analysis using the AMS eigensolution including residual modes for the BIW model.
biw_freq_lnz.inp	Frequency extraction analysis using the Lanczos eigensolver for the BIW model.
biw_ssd_lnz.inp	Subspace-based steady-state dynamic analysis using the Lanczos eigensolution for the BIW model.
biw_ssd_dir.inp	Direct-solution steady-state dynamic analysis for the BIW model.

References

Abaqus Analysis User's Guide

- "Using the SIM architecture for modal superposition dynamic analyses" in "Dynamic analysis procedures: overview," Section 6.3.1 of the Abaqus Analysis User's Guide
- "Direct-solution steady-state dynamic analysis," Section 6.3.4 of the Abaqus Analysis User's Guide
- "Natural frequency extraction," Section 6.3.5 of the Abaqus Analysis User's Guide
- "Mode-based steady-state dynamic analysis," Section 6.3.8 of the Abaqus Analysis User's Guide
- "Subspace-based steady-state dynamic analysis," Section 6.3.9 of the Abaqus Analysis User's Guide

Abaqus Keywords Reference Guide

- *FREQUENCY
- *STEADY STATE DYNAMICS

Abaqus Verification Guide

- "Steady-state dynamics with nondiagonal damping using the AMS eigensolver," Section 3.24.1 of the Abaqus Verification Guide

Table 2.2.6–1 Overall performance comparison of the steady-state dynamic analysis procedures (hh:mm:ss).

Performance	SSD with AMS	SSD SP with Lanczos	SSD DIRECT
Total wallclock time	00:06:31	03:20:45	05:58:54
Wallclock time for eigensolver	00:05:35	01:36:24	N/A
Wallclock time for SSD	00:00:56	01:44:21	05:58:54

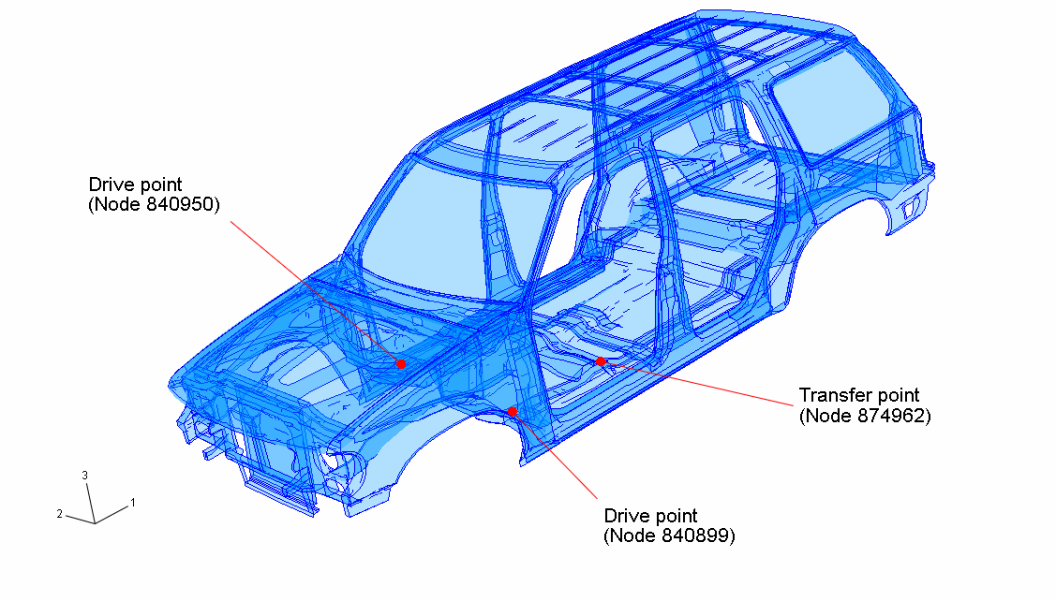


Figure 2.2.6–1 Body-In-White model of sport utility vehicle.

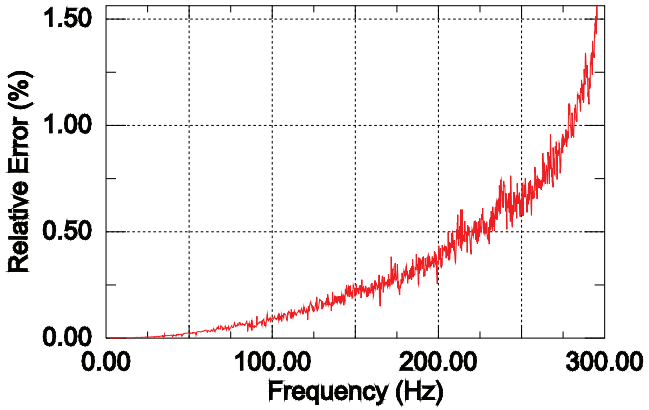


Figure 2.2.6–2 The relative errors of the natural frequencies computed by AMS with respect to the natural frequencies computed by Lanczos below the global cutoff frequency of 300 Hz.

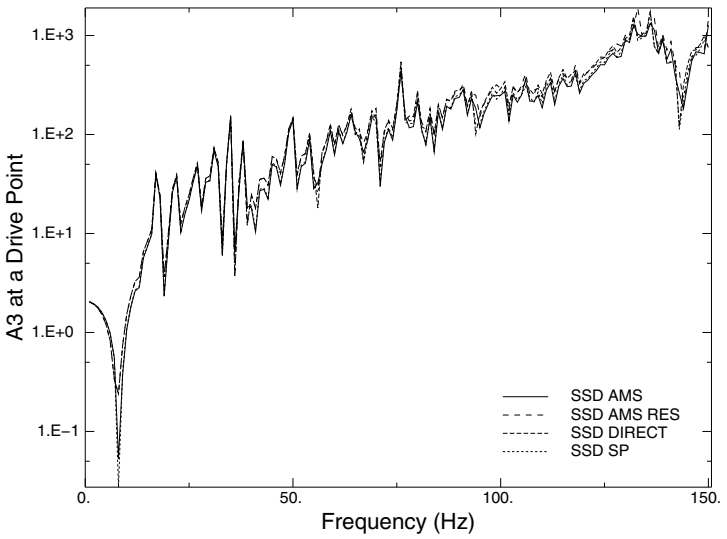


Figure 2.2.6–3 Point inertance at node 840950.

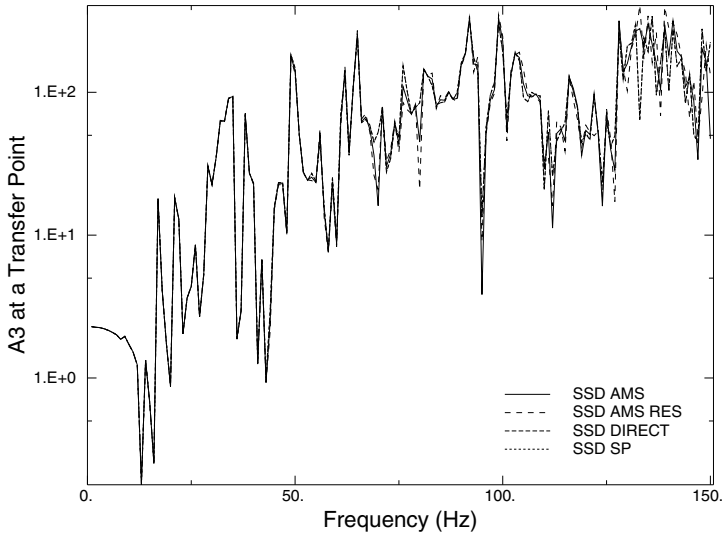


Figure 2.2.6–4 Transfer inertance at node 874962.

2.3 Eulerian analyses

- “Rivet forming,” Section 2.3.1
- “Impact of a water-filled bottle,” Section 2.3.2

2.3.1 RIVET FORMING

Products: Abaqus/Explicit Abaqus/CAE

Objectives

This example problem demonstrates the following aspects of a forming analysis in Abaqus/Explicit:

- using the coupled Eulerian-Lagrangian (CEL) analysis formulation to analyze a solid mechanics model undergoing extreme deformation, and
- comparing results from the CEL-based analysis to those from an identical model using the traditional Lagrangian formulation.

Application description

Rivets are a type of fastener designed to create permanent attachments between two or more sheets of material. A rivet design typically consists of a cylindrical body with two diameters: the smaller diameter is inserted through a hole in the overlapping sheets, then both ends of the rivet are compressed. The compression effectively expands the diameters of the rivet body, pinching the sheets of material between the two ends of the rivet (see Figure 2.3.1–1). Different rivet designs and applications will undergo different deformations, but the basic principle remains the same in all cases.

This example investigates the effectiveness of a particular rivet by simulating its compression (also referred to as the forming process). Three questions are of particular importance in this study:

- Does the rivet deform appropriately during the forming process?
- After the forming process, does the rivet retain enough strength to maintain a hold on the fastened materials?
- Is the rivet installation tool capable of forming the rivet?

The displacements during the forming simulation indicate whether or not the rivet deforms appropriately. After the deformation, the strength of the rivet is based largely on its material properties; examining the equivalent plastic strain in the rivet gives some indication of potential damage or strength degradation in the material. To assess the effect of the rivet on the installation tool, reaction forces in the tool can be compared to known force capacities in standard installation tools.

Geometry

The rivet used in these analyses is a simple multidiameter cylinder, as described above. To aid the deformation of the smaller end of the rivet, a hemispherical portion is removed from the center of the cylinder. Figure 2.3.1–2 shows the dimensions of the rivet model.

To simulate the forming, the rivet is placed in a hole at the center of a circular plate. Circular dies representing the installation tool are positioned at the top and bottom end of the rivet (see Figure 2.3.1–3).

Materials

The rivet in this model is composed of an elastic-plastic steel with a density of 7.85×10^{-9} t/mm^3, Young's modulus of 2.1×10^5 N/mm^2, Poisson's ratio of 0.266, and the onset of plastic yielding occuring at 3.0×10^5 N/mm^2.

The plate and dies are assumed to be significantly harder than the rivet, and no deformation of these parts is expected.

Boundary conditions and loading

The forming process is simulated through the enforcement of displacement boundary conditions. The plate is constrained to a fixed location. The top die is displaced downward a distance of 3 mm, while at the same time the bottom die is displaced upward a distance of 2 mm.

Interactions

Contact interactions must be enforced between the rivet and all installation tool components; the deformation of the rivet depends on contact loads delivered through the displacement of the tool. The tool components never come into contact with each other, so interactions between the plate and dies can be neglected.

Abaqus modeling approaches and simulation techniques

The rivet forming simulation is conducted in Abaqus/Explicit using two fundamentally different element formulations. The traditional Lagrangian formulation generally offers accuracy and computational efficiency, but pure Lagrangian models tend to exhibit mesh distortion and an associated loss of accuracy when undergoing extreme deformations. The Eulerian formulation trades some accuracy of geometry and results for robustness in analyses involving very large deformations; in situations where the Lagragian formulation yields an unreliable solution or no solution at all, the Eulerian formulation can be used to obtain a reasonable solution.

Lagrangian and Eulerian elements can be combined in the same model using a technique known as coupled Eulerian-Lagrangian (CEL) analysis. In a CEL analysis bodies that undergo large deformations are meshed with Eulerian elements, while stiffer bodies in the model are meshed with more efficient Lagrangian elements.

The rivet forming analysis is performed using both a pure Lagrangian approach, in which the rivet, plate, and dies are all modeled with Lagrangian elements; and a coupled Eulerian-Lagrangian approach, in which the rivet is modeled with Eulerian elements while the plate and dies are modeled with Lagrangian elements.

Summary of analysis cases

Case 1 Pure Lagrangian rivet forming analysis.

Case 2 Coupled Eulerian-Lagrangian (CEL) rivet forming analysis.

The following sections detail some modeling techniques that are common to both analysis cases.

Analysis types

Both analysis cases are conducted using quasi-static explicit dynamic procedures. The forming takes place over the course of a single step lasting 1 ms.

Material model

The material for the rivet uses an isotropic hardening Mises plasticity model. The stress-strain data points used to define the plastic behavior are shown in Table 2.3.1–1.

Boundary conditions

In both analysis cases the plate and dies are modeled as Lagrangian bodies imposed with rigid body constraints. A boundary condition preventing displacements and rotations is imposed on the reference point for the plate body. Boundary conditions are also applied to each die reference point to prevent them from rotating or displacing, except in the vertical 3-direction: the boundary condition on the top reference point displaces it 3 mm in the negative 3-direction, and the boundary condition on the bottom reference point displaces it 2 mm in the positive 3-direction. The application of the boundary conditions is governed by an amplitude that ramps the displacement linearly from zero to full displacement over the course of 0.8 ms; the dies are then fixed in place for the final 0.2 ms of the analysis.

Constraints

As mentioned above, rigid body constraints are applied to the plate and two dies. These components are assumed to be significantly harder than the rivet, and they do not deform during the forming process. The rigid body constraints improve computational efficiency and allow the use of simple boundary conditions to initiate the forming.

Output requests

Field output is specifically requested for the equivalent plastic strains in the model (PEEQ). In addition, history output for the reaction force in the 3-direction (RF3) is requested at the reference point for each of the dies.

Pure Lagrangian analysis case

The first analysis case uses four Lagrangian bodies meshed from discrete geometric part instances. In the pure Lagrangian case the geometry of the model corresponds directly to the shapes of the parts being modeled, making the assembly process fairly intuitive.

Mesh design

The rivet is meshed with C3D8R elements using a global mesh seed of 0.25 mm (see Figure 2.3.1–4).

The plate and dies are also meshed with C3D8R elements, but the rigid body constraints applied to these parts makes the element selection somewhat arbitrary. Unmeshed analytical rigid surfaces could have been used to model the plate and dies, but rigid body constraints are used to maintain consistency with the CEL model.

Interactions

A general contact definition enforces contact interactions between all bodies in the model. A frictionless, hard contact model governs all interactions.

Solution controls

Although large deformations are expected in the analysis, no special solution controls or analysis techniques (such as adaptive meshing) are applied to the model, allowing a straightforward comparison between the pure Lagrangian model and the CEL model.

CEL analysis case

In the second analysis case the rivet is modeled using Eulerian elements. The plate and dies are still rigid bodies. The modeling approach in the CEL analysis has some distinct differences from the pure Lagrangian case.

Mesh design

In the Eulerian formulation the mesh does not generally correspond to the geometry of the part being modeled; rather, the placement of the material within the Eulerian mesh defines the geometry of the part. The Eulerian mesh does not deform or displace; only the materials within the mesh can move. Typically, the Eulerian mesh is an arbitrary collection of regular hexahedral elements that fully encompasses the region in which material may exist during the analysis.

In this example the Eulerian part is a rectangular prism measuring $17 \times 17 \times 11.5$ mm meshed with EC3D8R elements. A global mesh seed of 0.25 mm dictates the element size.

This mesh does not define the geometry of the rivet; rather, the mesh defines the domain in which the rivet material can exist. The rivet geometry is defined by assigning steel material to a portion of this mesh corresponding to the shape of the rivet, as discussed in the "Initial conditions" section below. One strength of the Eulerian technique is the ability to define a regular, high-quality mesh independent of the geometry of the part being modeled.

It is important that the Eulerian mesh is large enough to contain the rivet material completely as it deforms; if material reaches the edge of the mesh, it flows out of the model and is lost to the simulation.

Initial conditions

The rivet geometry is defined using a material assignment initial condition on the Eulerian mesh. The material assignment specifies which elements in the mesh initially contain steel. Each element is designated a percentage (known as the volume fraction), which represents the portion of that element filled with steel. For partially filled elements Abaqus positions the material in the element such that it forms a continuous surface with the material in adjacent elements. The end result is a distribution of material in the mesh corresponding to the rivet geometry, as seen in Figure 2.3.1–5. You can use the view cut manager in the Visualization module of Abaqus/CAE to visualize the extent of a material within an Eulerian mesh, as discussed in "Viewing output from Eulerian analyses," Section 28.7 of the Abaqus/CAE User's Guide.

The material assignment is created with the aid of the volume fraction tool in Abaqus/CAE. The volume fraction tool calculates the overlap between an Eulerian mesh and some reference geometric part. To use the volume fraction tool for this analysis case, the entire Lagrangian assembly (including the Lagrangian rivet) is copied from the previous analysis case and positioned inside the Eulerian mesh (see Figure 2.3.1–6). The Lagrangian rivet serves as the reference part, and the volume fraction tool creates a discrete field that associates each element in the Eulerian mesh with a volume fraction based on the amount of space occupied by the rivet within that element. This discrete field can then be used as the basis for a material assignment predefined field in Abaqus/CAE.

Interactions

A general contact definition enforces contact interactions between all rigid bodies and Eulerian materials in the model. General contact does not enforce contact between the rigid bodies and the Eulerian elements; the rigid bodies can pass unimpeded through the Eulerian mesh until they encounter a material within the mesh. As with the pure Lagrangian case, a frictionless, hard contact model governs all interactions.

It is generally not advisable to model Lagrangian-Eulerian contact near the boundary of the Eulerian mesh. The inflow or outflow of materials at the mesh boundary can lead to improper contact constraint enforcement. Therefore, the Eulerian mesh extends a few elements past the contact interfaces between the dies and the rivet.

General contact does not enforce interactions between analytical rigid surfaces and Eulerian materials, which is why the tool components must be modeled as Lagrangian parts with rigid body constraints.

Output requests

In addition to the field and history output requests used in the pure Lagrangian analysis case, the Eulerian volume fraction output variable (EVF) is requested as field output to visualize geometric results.

Discussion of results and comparison of cases

Figure 2.3.1–7 shows the deformed meshes for the pure Lagrangian and the CEL analysis cases. (To view the results of the CEL analysis, use the view cut manager as described in "Viewing output from Eulerian analyses," Section 28.7 of the Abaqus/CAE User's Guide.) The pure Lagrangian analysis runs to completion, but the mesh becomes extremely distorted along the bottom of the rivet—results from such an irregular mesh may be unreliable. The Eulerian analysis exhibits a similar deformed shape but retains a high-quality, regular mesh.

Computational efficiency

In general, an Eulerian analysis is more expensive than a comparable Lagrangian analysis in terms of run times and file sizes. This performance tradeoff should be weighed against the benefits of Eulerian robustness for large deformations when choosing an analysis formulation.

Difficulties with contact

Figure 2.3.1–8 shows the contact interface between the rivet and the plate for both the pure Lagrangian and the CEL cases. Both cases demonstrate some undesirable penetration of the rivet into the plate.

In the pure Lagrangian case the penetrations are a direct result of mesh distortion. As the rivet's facets spread out, there are fewer constraint points per given area, and portions of the facets are able to pass unconstrained into the plate surface. While the Lagrangian formulation is generally adept at simulating contact, a severely deformed mesh can cause noisy, inconsistent contact enforcement.

In the CEL case the penetrations result largely from approximations used to visualize Eulerian material. The boundary of an Eulerian material does not correspond to a discrete element face. As discussed previously, Abaqus determines the location of material in an Eulerian mesh based on a volume fraction within each element; the volume fractions are averaged and interpolated to calculate a smooth material surface during visualization. Therefore, the material boundary displayed in the Visualization module of Abaqus/CAE is an approximation based on numerical averaging, not geometric properties. The approximation causes the apparent penetrations at contact interfaces and explains why sharp corners in a Lagrangian model appear rounded in an Eulerian model. Despite the apparent penetrations, coupled Eulerian-Lagrangian contact does not suffer from the inconsistent constraint enforcement associated with pure Lagrangian mesh distortion, and contact between a solid Lagrangian body and an Eulerian material typically delivers reliable results.

In both analysis cases contact penetrations can be alleviated through the use of a finer mesh: smaller elements lead to reduced mesh distortion in the pure Lagrangian case, and additional elements provide additional sample points for more accurate volume fraction averaging in the Eulerian case.

Interpreting results

A contour plot of the equivalent plastic strains in a cross-section of the deformed rivet appears in Figure 2.3.1–9 for both analysis cases. The results are similar, but the regions of maximum plastic strain occur in slightly different areas. In the Eulerian rivet the peak strain occurs near the corner where the rivet meets the bottom of the plate; this region undergoes extreme bending and stretching during the forming. In the Lagrangian rivet the peak strain occurs in the elements with the most severe distortion. For small to moderate deformation, the Eulerian approach provides results that are comparable to the traditional Lagrangian approach (though at a higher computational cost); for large deformation the Eulerian results appear more reliable than the Lagrangian results.

Figure 2.3.1–10 plots the reaction forces in both dies over the course of the forming process. The results for the two formulations are comparable during the first half of the analysis. However, the force plots for the pure Lagragian analysis case exhibit some noise during the latter half of the forming and subsequently diverge from the Eulerian force plots. The noise is likely the result of contact difficulties in the model (discussed above), which in turn lead to an uneven transfer of force between the tooling and the rivet. After 0.8 ms, when the dies have fully displaced, the forces in the Eulerian case exhibit some relaxation; the forces in the Lagrangian case remain steadily higher than the corresponding Eulerian forces due to the extreme deformation in the mesh.

Python script

rivet_forming.py	Script to generate the pure Lagrangian and the CEL models in Abaqus/CAE.

Input files

rivet_forming_lag.inp	Input file for the pure Lagrangian model.
rivet_forming_cel.inp	Input file for the CEL model.

References

Abaqus Analysis User's Guide

- "Eulerian analysis," Section 14.1.1 of the Abaqus Analysis User's Guide
- "Eulerian elements," Section 32.14 of the Abaqus Analysis User's Guide

Abaqus/CAE User's Guide

- Chapter 28, "Eulerian analyses," of the Abaqus/CAE User's Guide

Abaqus Keywords Reference Guide

- *EULERIAN SECTION
- *INITIAL CONDITIONS

Table 2.3.1–1 Plasticity data for rivet material.

Yield stress (N/mm^2)	Plastic strain
3.0×10^5	0
4.5×10^5	0.02
7.5×10^5	0.34
1.4×10^6	0.35

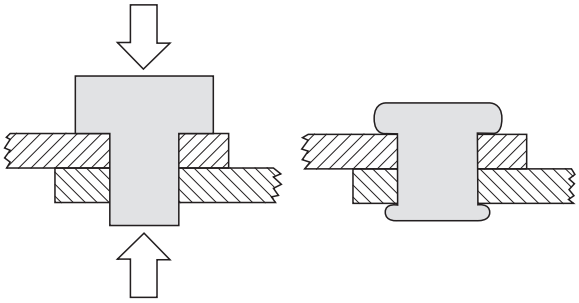


Figure 2.3.1–1 Using a rivet to fasten two sheets of material.

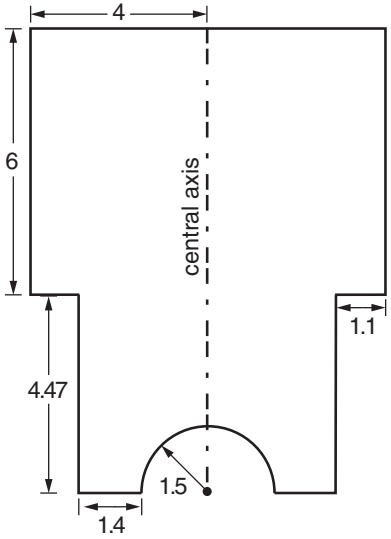


Figure 2.3.1–2 Rivet model geometry. All dimensions are in millimeters.

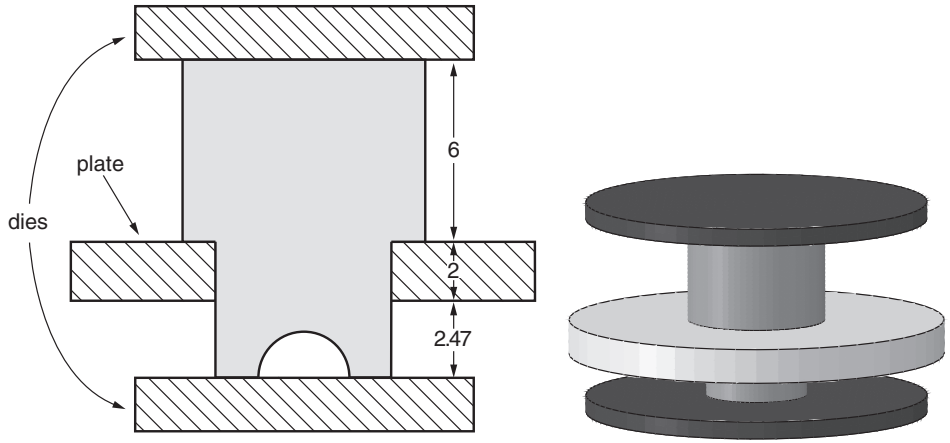


Figure 2.3.1–3 Rivet assembled with the forming tools. All dimensions are in millimeters.

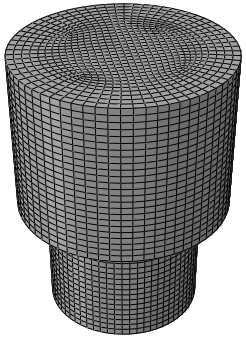


Figure 2.3.1–4 Rivet mesh in the pure Lagrangian case.

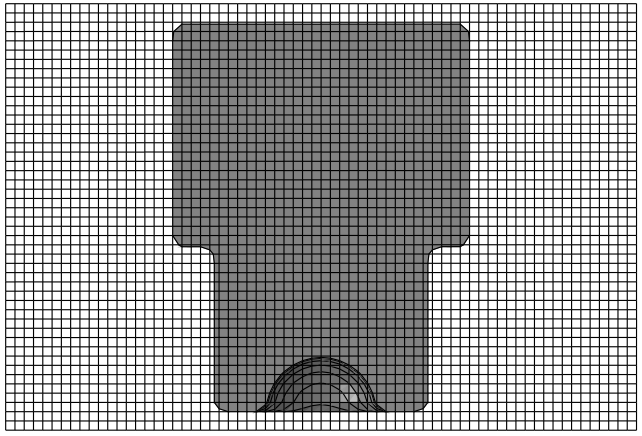


Figure 2.3.1–5 Cross-section of the Eulerian mesh with the rivet material assigned.

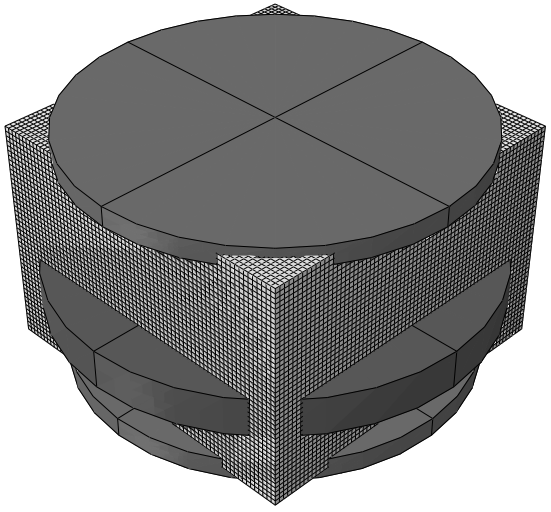


Figure 2.3.1–6 The Lagrangian parts are assembled inside of the Eulerian mesh.

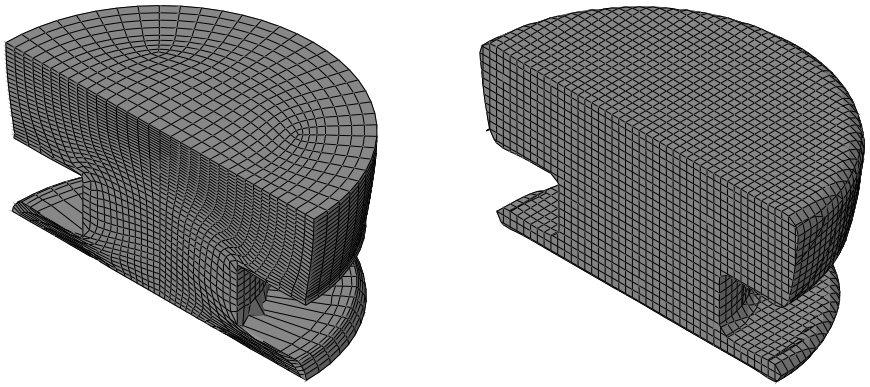


Figure 2.3.1–7 Deformed configuration for the pure Lagrangian case (left) and the CEL case (right).

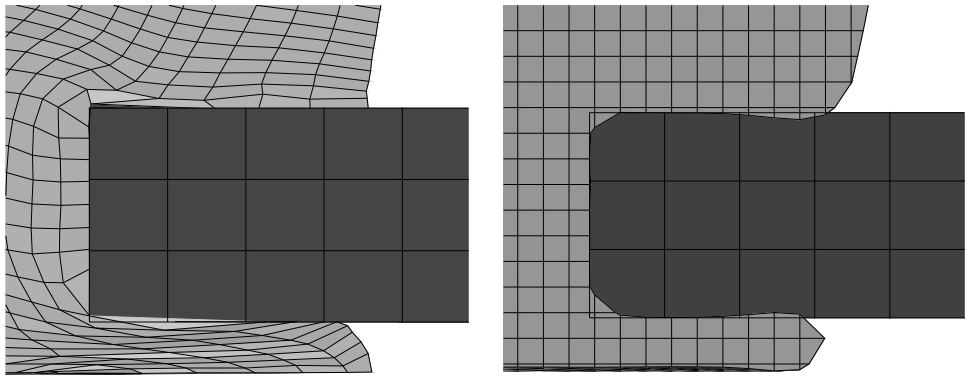


Figure 2.3.1–8 Contact penetrations between the rivet and plate for the pure Lagrangian case (left) and the CEL case (right).

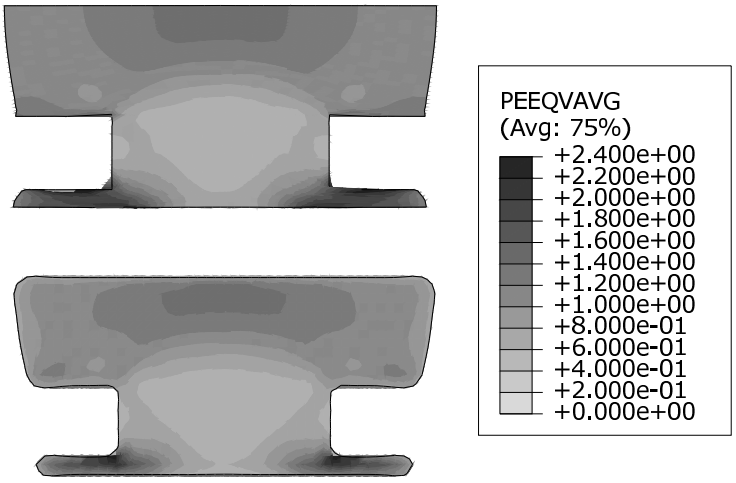


Figure 2.3.1–9 Equivalent plastic strain in the pure Lagrangian model (top) and the CEL model (bottom).

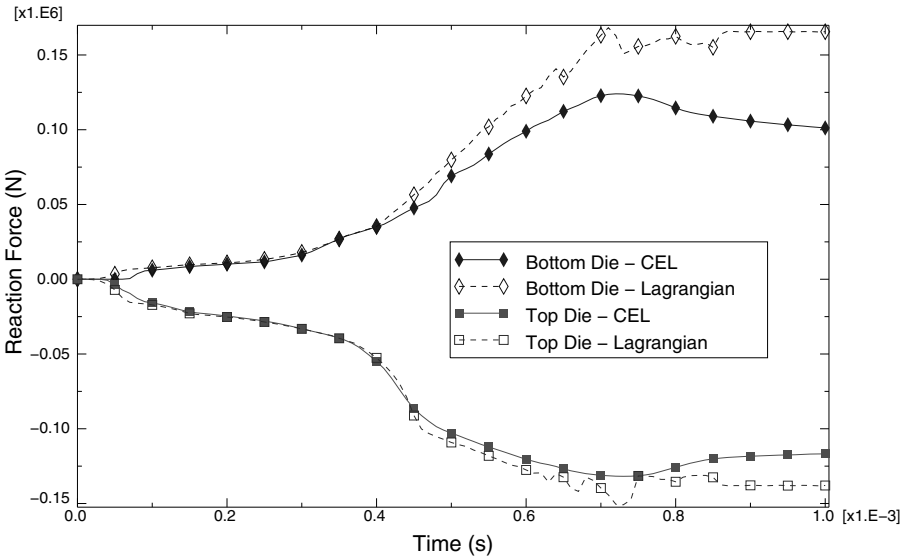


Figure 2.3.1–10 Reaction forces in the forming dies.

2.3.2 IMPACT OF A WATER-FILLED BOTTLE

Product: Abaqus/Explicit

Objectives

This example problem demonstrates the following techniques in Abaqus:

- using the volume fraction tool in Abaqus/CAE to model complex material distributions in an Eulerian element mesh,
- using the Eulerian-Lagrangian contact formulation to simulate a highly dynamic event involving a fluid material (modeled using Eulerian elements) interacting with structural boundaries (modeled using Lagrangian elements),
- using the smoothed particle hydrodynamic (SPH) method to analyze a highly dynamic event in a purely Lagrangian environment, and
- converting continuum finite elements to SPH particles to analyze the highly dynamic event.

Application description

Simulation is commonly used in the consumer packaging industry to reduce the time and cost associated with physical prototyping. Drop tests, which simulate an object falling and impacting a hard surface, are often used to investigate the object's response under harsh handling conditions.

This example involves a fluid-filled plastic bottle falling from a height of roughly 300 mm onto a flat, rigid floor. The bottle, as shown in Figure 2.3.2–1, is a rectangular jug made of high-density polyethylene. The bottle is filled almost completely (about 95%) with water. A realistic simulation for the bottle must account for both the exterior forces on the bottle from the floor impact, as well as the interior forces of the water pushing against the bottle. Resulting stresses and strains in the bottle can be used to determine its structural feasibility.

Geometry

Figure 2.3.2–1 shows the pertinent dimensions for the bottle and cap model. The bottle has a uniform thickness of 0.5 mm, with the exception of the rim around the bottle's mouth; the rim is 0.65 mm thick. The bottle's cap is modeled as a separate part instance and positioned on the mouth of the bottle; the cap is uniformly 1 mm thick.

The bottle strikes the floor at a skew angle, with one of the bottom corners experiencing the initial impact. Figure 2.3.2–2 shows the assembled model. Initially the water in the bottle is distributed according to a gravitational response; that is, the boundary of the water is parallel to the horizontal floor, not to the bottom of the skewed bottle.

Materials

The bottle is constructed of high-density polyethylene that follows an isotropic plastic hardening model.

The water is treated as a nearly incompressible, nearly inviscid Newtonian fluid.

Boundary conditions and loading

The entire model is subject to a gravity load. The rigid floor is fixed in place.

Interactions

The bottle contacts three different model components over the course of the analysis: the floor, the bottle cap, and the water within the bottle. All of these contact interactions are assumed to be frictionless.

Abaqus modeling approaches and simulation techniques

The primary challenge to solving this problem is the highly transient fluid-structure interaction between the water and the bottle. In this example two methods are studied. The coupled Eulerian-Lagrangian (CEL) analysis technique in Abaqus/Explicit is well suited to handling problems of this nature. The SPH method can be used to model the violent sloshing associated with the impact.

Summary of analysis cases

Case 1	Coupled Eulerian-Lagrangian analysis.
Case 2	SPH analysis.
Case 3	Finite element conversion to SPH particle analysis.

Mesh design

The bottle is imported as an orphan mesh of S3R and S4R elements. The cap geometry is meshed with S4R elements. Figure 2.3.2–3 shows the assembled mesh for the bottle and cap. The floor is meshed with surface elements of type SFM3D4R; a rigid body constraint is subsequently applied to these elements (see "Constraints" below).

Material model

An elastic-plastic material definition is used for the polyethylene in the bottle, with the isotropic hardening curve defined by the data points in Table 2.3.2–1; the onset of plastic yielding occurs at 8.618 N/mm^2, and failure occurs at a strain of 0.59. Damage is incorporated using a ductile damage definition. The high-density polyethylene has a density of 8.76×10^{-7} kg/mm^3, Young's modulus of 903.114 N/mm^2, and Poisson's ratio of 0.39.

The water is modeled using the linear $U_s - U_p$ Hugoniot form of the Mie-Grüneisen equation of state; the equation parameters appear in Table 2.3.2–2. The water has a density of 9.96×10^{-7} kg/mm^3 and bulk modulus of 2.094 GPa.

Boundary conditions

An encastre boundary condition is applied to the reference point of the rigid floor to fix it in place for the duration of the analysis.

Loading

A gravity load is applied to the bottle, cap, and water instances. An acceleration of -9800 mm/s^2 is applied in the z-direction.

Predefined fields

Instead of simulating the full dropping event from the initial position, the bottle, cap, and water instances are positioned close to the floor and prescribed an initial velocity predefined field. An initial velocity of -2444 mm/s in the z-direction corresponds to the speed that would be attained by an object falling about 300 mm from rest under typical gravitational acceleration.

Constraints

A rigid body constraint applied to the floor part instance makes the floor a simple, undeformable surface.

Interactions

General contact is defined for the model. General contact enforces interactions between the bottle and other Lagrangian components, such as the cap and floor. The default frictionless hard contact property governs all interactions.

In an actual bottle, the cap would be firmly attached to the neck of the bottle. This interaction between the bottle and the cap is considered insignificant for the purposes of this analysis. Instead of adding undue cost to the analysis by modeling a threaded connection or tie constraint between the two parts, the cap is allowed to freely detach from the bottle during the simulation.

Output requests

Displacements, velocities, and accelerations are requested for the bottle and cap. Logarithmic strain (LE) and equivalent plastic strain (PEEQ) are requested for the bottle to assess its structural response. Damage initiation criteria (DMICRT) and element status (STATUS) are also requested for the polyethylene bottle to track potential failure in these components. Finally, contact stresses (CSTRESS) and contact forces (CFORCE) are requested for the Lagrangian part instances.

Case 1 Coupled Eulerian-Lagrangian analysis

This case shows the use of the Eulerian-Lagrangian contact formulation to simulate the bottle drop with the fluid material (modeled using Eulerian elements) interacting with structural boundaries (modeled using Lagrangian elements). In addition, the volume fraction tool in Abaqus/CAE is used to model complex material distributions in an Eulerian element mesh.

Analysis types

The full simulation is conducted in a single explicit dynamic step lasting 0.05 s.

A few distorted Lagrangian elements in the model are controlling the stable time increment, thus dictating the time to complete the analysis. Using the semi-automatic mass scaling method, the time increment can be increased to 7.528×10^{-7} s, reducing the analysis cost by nearly half, while increasing

the total mass by only 0.11 percent. Alternatively, using the mass adjustment method, the mass can be redistributed among the elements of the bottle to achieve the same time increment without affecting the total mass of the bottle. A large number of elements have mass in excess of what is required to achieve the above stable time increment. Only 0.12 percent of this excess mass is redistributed to the remaining elements to raise their time increment to the specified value. In both methods, the change in the mass distribution is very small and does not significantly influence the analysis results.

Analysis techniques

The Eulerian element formulation allows the analysis of bodies undergoing severe deformation without the difficulties traditionally associated with mesh distortion. In an Eulerian mesh material flows through fixed elements, so a well-defined mesh at the start of an analysis remains well-defined throughout the analysis. Although Eulerian material boundaries are more approximate than traditional Lagrangian element boundaries, the Eulerian formulation allows you to capture extreme deformation phenomena such as fluid flow. The water is modeled using an Eulerian element domain. The bottle—which, although significantly stiffer than the water, is not rigid—is modeled using traditional Lagrangian shell elements. The general contact algorithm in Abaqus/Explicit tracks and enforces contact between the Eulerian material boundary and the Lagrangian elements, enabling effective simulation of the fluid-structure interaction.

Mesh design

The Eulerian mesh, which serves as the domain through which the water material can flow, is a $300 \times 250 \times 200$ mm rectangular prism of EC3D8R elements. Each Eulerian element is a regular cube measuring 5 mm on an edge. The Eulerian mesh completely encompasses the bottle and cap, and it extends slightly below the floor. Any interface that is expected to experience Eulerian-Lagrangian contact must be located within the Eulerian mesh; once an Eulerian material passes beyond the boundaries of the Eulerian mesh, it is lost to the simulation and contact is not enforced. The overlapping of Lagrangian and Eulerian elements is acceptable because these two element types do not interact with each other. Lagrangian elements interact only with Eulerian material within the mesh. The initial position of this Eulerian material must be defined within the Eulerian mesh, as discussed in "Initial conditions" below.

Initial conditions

Because an Eulerian mesh is void of any material by default, the desired distribution of material within the Eulerian mesh must be specified using an initial condition. This distribution is defined using the concept of an Eulerian volume fraction, or the percentage of an element that is occupied by a given material. For each Eulerian element that initially contains material, an initial Eulerian volume fraction for that material must be specified. Abaqus evaluates all of the element volume fractions to determine the distribution and boundaries of Eulerian materials within the Eulerian mesh.

Abaqus/CAE provides tools that greatly simplify the definition of initial material volume fractions, particularly for complex geometries. The desired material geometry can be modeled as a separate part (the reference part), then instanced into the Eulerian mesh in a position that corresponds to the intended material position. The volume fraction tool performs a Boolean comparison between the Eulerian mesh

and the reference part, then creates a discrete field that associates each element in the mesh with a percentage based on the amount of space occupied by the reference part within that element. This discrete field can be used in a material assignment predefined field to specify the appropriate Eulerian volume fractions for a particular material.

A part corresponding to the geometry of the water in the bottle is provided with this example (**`cel_bottle_drop_watergeom.sat`**). This part is imported into Abaqus/CAE, then instanced and oriented in the model assembly with the other part instances. The volume fraction tool uses the water geometry as the reference part instance within the Eulerian part instance (the Eulerian part must be meshed before using the tool). The resulting discrete field is used to create a water material assignment predefined field in the Load module. This material assignment distributes the water in the Eulerian mesh in a geometry corresponding to the water geometry part instance. The water geometry instance is subsequently suppressed to remove it from the model.

Although it is not used in the script for this example, a solid version of the bottle geometry is also available (**`cel_bottle_drop_solidgeom.sat`**). The water reference geometry can be created from this solid bottle using geometry cuts in the Assembly module (see "Performing Boolean operations on part instances," Section 13.7 of the Abaqus/CAE User's Guide).

Predefined fields

The initial material assignment for the Eulerian mesh is also defined in Abaqus/CAE with a predefined field. A material assignment predefined field associates the discrete field that was created by the volume fraction tool (see "Initial conditions") with the water material definition.

Interactions

The contact between the bottle and the Eulerian material initially positioned inside of the bottle is also modeled using the general contact algorithm.

Output requests

In addition to the field and history output requests listed above, the Eulerian volume fraction output variable (EVF) is requested as field output to visualize geometric results.

Case 2 SPH analysis

This case shows the use of a smoothed particle hydrodynamic (SPH) analysis to analyze the fluid-filled bottle drop in a purely Lagrangian environment.

Analysis types

The full simulation is conducted in a single explicit dynamic step lasting 0.05 s.

Analysis techniques

The water is modeled using continuum pseudo-particles. Internally, Abaqus/Explicit determines automatically every increment of the analysis which are the active neighbors associated with a given particle of interest in order to apply the SPH formulation. Since the internally determined connectivity

is allowed to change every increment, the method robustly handles the severe deformations associated with the sloshing water. The other Lagrangian bodies (bottle, cap, and floor) are modeled the same as in the coupled Eulerian-Lagrangian method.

Mesh design

The pseudo-particles are modeled using PC3D elements that are spaced in a relatively uniform fashion every 5 mm in all directions. These one-node elements fill only the space initially occupied by the water just before the impact. Therefore, fewer PC3D elements are needed when compared to the number of EC3D8R elements in the coupled Eulerian-Lagrangian method.

Interactions

A node-based surface associated with the water pseudo-particle is included in the contact domain to model the interactions between the water and the bottle.

Case 3 Finite element conversion to SPH particles analysis

This case shows the use of the technique for converting reduced-integration continuum elements to SPH particles.

Analysis types

The full simulation is conducted in a single explicit dynamic step lasting 0.05 s.

Analysis techniques

The water is modeled using reduced-integration continuum C3D4 elements. A time-based criterion is used to trigger the conversion from a user-defined mesh to SPH particles (PC3D elements) at the beginning of the analysis. Upon conversion the continuum elements modeling the fluid will become inactive (deleted from the mesh) while the PC3D elements get activated. Since all particles get activated at the same time, the SPH functionality will model the fluid as described in Case 2.

Mesh design

The water is modeled using reduced-integration continuum C3D4 elements using regular preprocessing techniques; the mesh is shown in Figure 2.3.2–4. Thus, cumbersome meshing approaches based on unconnected particles are avoided.

Interactions

Contact between the fluid and the bottle is defined by including an element-based surface that includes the fluid's interior and its initial external surface in the general contact domain. Internally, Abaqus/Explicit generates a node-based surface for the generated SPH particles that will be also included in the contact domain.

Discussion of results and comparison of cases

The results of the drop test in the CEL analysis appear in Figure 2.3.2–5. (For tips on viewing the results of CEL analyses, see "Viewing output from Eulerian analyses," Section 28.7 of the Abaqus/CAE User's Guide.) The water contributes significantly to the behavior of the bottle: the bottle flexes and bulges as the water sloshes. Figure 2.3.2–6 shows the logarithmic strains in the bottle. Strains appear in the sides of the bottle as they bulge outward, but these strains are largely recovered when the water sloshes upward. The most significant strains occur on the bottom of the bottle, along the bottle-floor interface. Despite the deformation caused by the floor and the water, the damage criteria for the polyethylene is not met at any location on the bottle.

A comparison between the CEL method and the SPH method at different stages during impact is illustrated in Figure 2.3.2–7 through Figure 2.3.2–10. The comparison between the two methods is quite reasonable.

The effect of sloshing water can also be seen in the reaction force on the floor surface. The reaction force in the z-direction is plotted in Figure 2.3.2–11 using a Butterworth filter with a cutoff frequency of 2000. The reaction force peaks sharply at approximately 0.02 s, which corresponds roughly with the appearance of strains in the sides of the bottle. After this point, the water surges upward, offsetting the downward momentum of the bottle and reducing the reaction forces in the floor. Good agreement between the two methods used is observed.

The significant deformation of the fluid in this problem clearly influences the analysis results. The coupled Eulerian-Lagrangian and SPH techniques in Abaqus/Explicit provide an effective way to realistically capture the complexity of the fluid-structure dynamics.

Files

The basic system of units in this problem is kilograms (kg), millimeters (mm), and seconds (s). Under this system, the base unit of force is kg mm/s^2, or 10^{-3} newtons; values for all force-derived quantities are entered in the sample files using this base unit. Values for all force-derived quantities are reported in the documentation in units of standard newtons (kg m/s^2) and scaled appropriately.

Case 1 Coupled Eulerian-Lagrangrian analysis

cel_bottle_drop.py	Script to generate the model in Abaqus/CAE using the orphan mesh from cel_bottle_drop_mesh.inp and the geometry from cel_bottle_drop_watergeom.sat.
cel_bottle_drop_mesh.inp	Orphan mesh for the bottle.
cel_bottle_drop_watergeom.sat	Reference geometry for the water.
cel_bottle_drop_solidgeom.sat	Reference geometry for the solid bottle.
cel_bottle_drop.inp	Complete input file for the model.
cel_bottle_drop_massadjust.inp	Complete input file for the model with the target time increment maintained by mass adjustment instead of mass scaling.

Case 2 SPH analysis

sph_bottle_drop.inp	Complete input file for the model.

Case 3 Finite element conversion to SPH particles analysis

sphconv_bottle_drop.inp	Finite element conversion to SPH particles input file for the model.

References

Abaqus Analysis User's Guide

- "Eulerian analysis," Section 14.1.1 of the Abaqus Analysis User's Guide
- "Smoothed particle hydrodynamics," Section 15.2.1 of the Abaqus Analysis User's Guide
- "Finite element conversion to SPH particles," Section 15.2.2 of the Abaqus Analysis User's Guide
- "Classical metal plasticity," Section 23.2.1 of the Abaqus Analysis User's Guide
- "Equation of state," Section 25.2.1 of the Abaqus Analysis User's Guide

Abaqus/CAE User's Guide

- Chapter 28, "Eulerian analyses," of the Abaqus/CAE User's Guide
- "Creating discrete fields for material volume fractions," Section 63.4 of the Abaqus/CAE User's Guide, in the HTML version of this guide

Table 2.3.2–1 Isotropic plastic hardening data for polyethylene material.

Yield stress (N/mm^2)	Plastic strain
8.618	0.0
13.064	0.007
16.787	0.025
18.476	0.044
20.337	0.081
24.543	0.28
26.887	0.59

Table 2.3.2–2 Material parameters for water.

Parameter	Value
Density (ρ)	9.96×10^{-7} kg/mm^3
Viscosity (η)	1×10^{-8} N s/mm^2
c_0	1.45×10^{6} mm/s
s	0
Γ_0	0

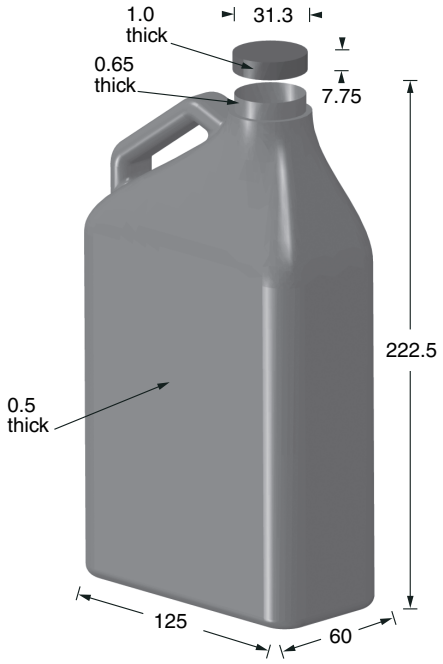


Figure 2.3.2–1 Bottle used in the drop test. All dimensions are in millimeters.



Figure 2.3.2–2 Assembly of the bottle drop model; the water is instanced inside of the bottle.

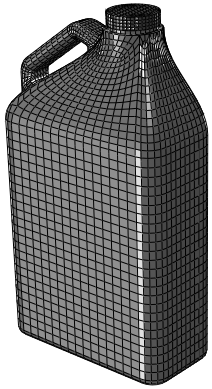


Figure 2.3.2–3 Mesh of the bottle and cap.

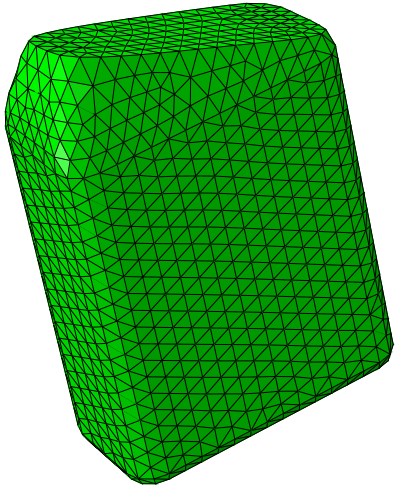


Figure 2.3.2–4 Tetrahedral mesh to model the fluid in the bottle in the undeformed configuration.

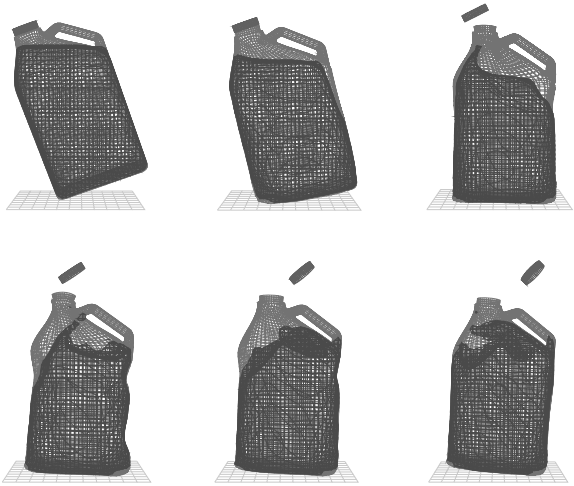


Figure 2.3.2–5 Deformed shape plots of the CEL model at 0.01 s intervals.

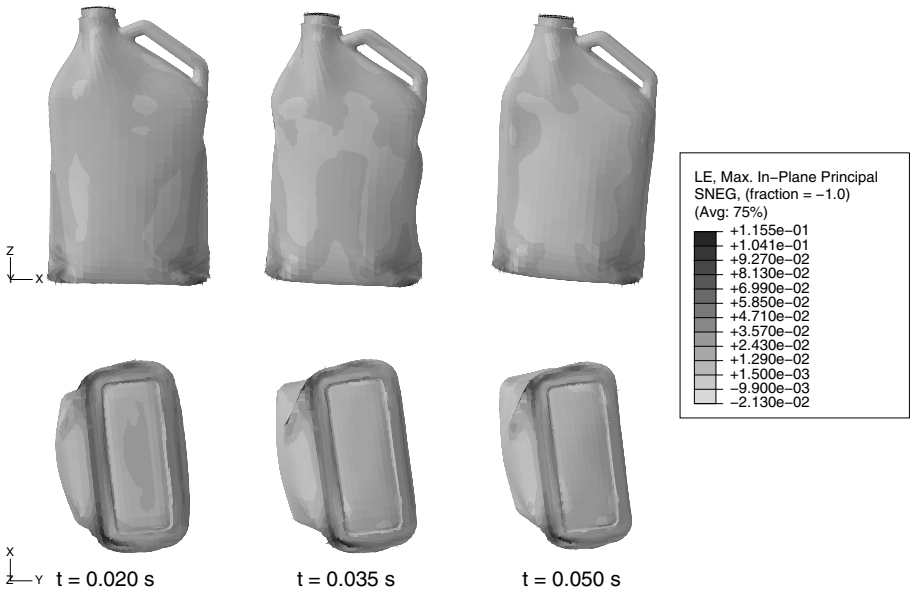


Figure 2.3.2–6 Logarithmic strains in the bottle during deformation in the CEL analysis.

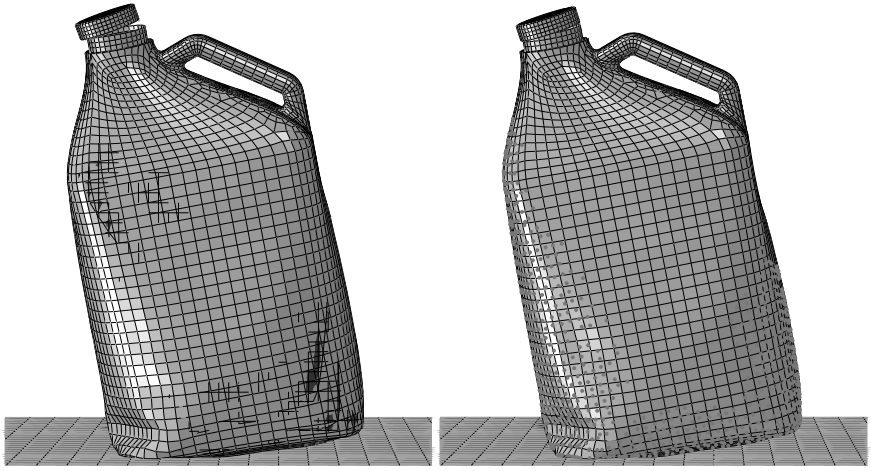


Figure 2.3.2–7 Comparison between coupled Eulerian-Lagrangian and SPH results shortly after the impact.

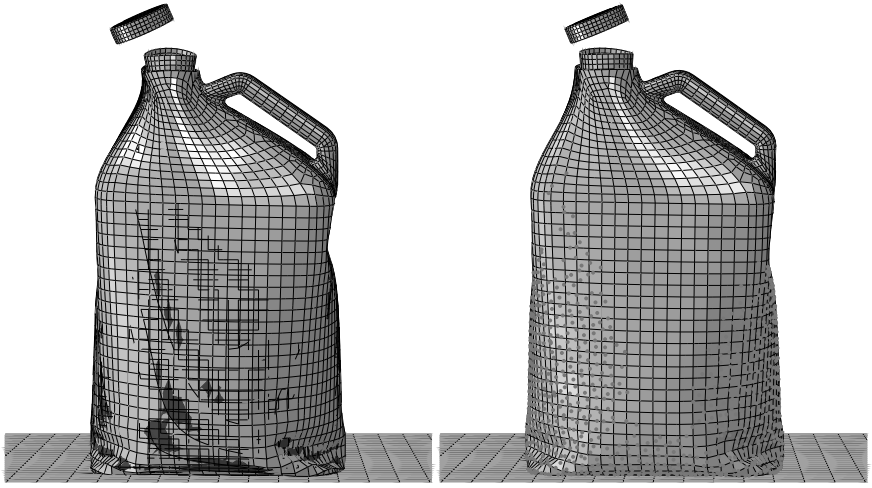


Figure 2.3.2–8 Comparison between coupled Eulerian-Lagrangian and SPH results after full contact with the floor.

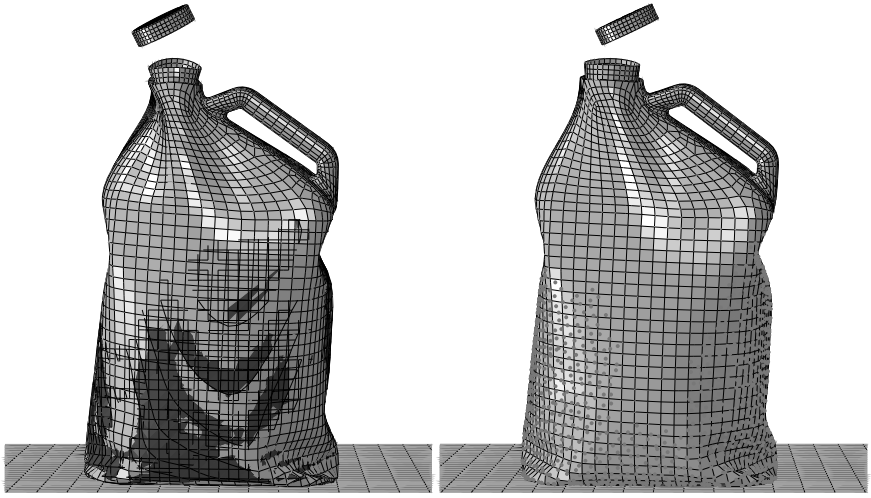


Figure 2.3.2–9 Comparison between coupled Eulerian-Lagrangian and SPH results in the middle stages of the analysis.

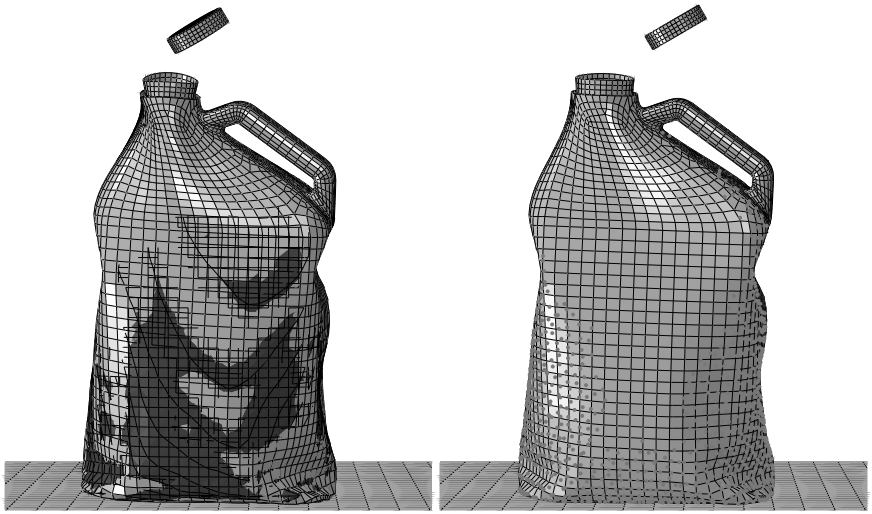


Figure 2.3.2–10 Comparison between coupled Eulerian-Lagrangian and SPH results towards the end of the analysis.

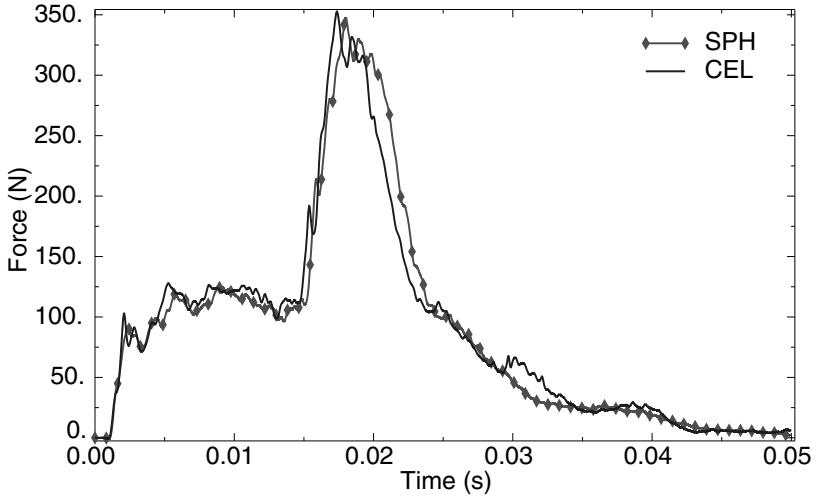


Figure 2.3.2–11 Reaction force (filtered) in the floor.

2.4 Co-simulation analyses

- “Dynamic impact of a scooter with a bump,” Section 2.4.1

2.4.1 DYNAMIC IMPACT OF A SCOOTER WITH A BUMP

Products: Abaqus/Standard Abaqus/Explicit

Objectives

This example demonstrates the combined use of Abaqus/Standard and Abaqus/Explicit to provide a more cost effective solution than by using either Abaqus/Standard or Abaqus/Explicit alone. Abaqus features and techniques demonstrated include:

- Abaqus/Standard to Abaqus/Explicit co-simulation, where
- Abaqus/Standard employs substructuring, a feature not available in Abaqus/Explicit, to efficiently handle modeling of a component subjected to small strains, and
- Abaqus/Explicit is used to efficiently simulate high-speed contact interactions.

Application description

This example considers the impact of a recreational scooter with a bump. An analysis of the transient response of an interaction with a bump is used to determine the accelerations felt by the scooter operator. With this analysis a product designer can make informed design decisions by varying certain design parameters, such as the frame component cross-section properties, tire material, or inflation pressure and observing their influence on the operator acceleration. Effective use of this simulation technique requires that the simulation turnaround time be as quick as possible while retaining essential fidelity in the results.

Geometry

The scooter consists of an operator deck, a frame with handlebars, and two tires, as shown in Figure 2.4.1–1. The overall length of the scooter is 1200 mm, and the overall height is 800 mm. The handlebars are oriented straight.

Materials

The scooter deck and frame tubing are made of mild steel. The tires are a Butyl rubber material.

Initial conditions

The simulation begins with the scooter having inflated tires and traveling toward the bump at a speed of 3 m/s.

Boundary conditions and loading

Weight loading is due to the scooter and the operator.

Interactions

The scooter travels on a rough surface. This surface includes a bump that is 7.5 cm high and is positioned in front of the scooter.

Abaqus modeling approaches and simulation techniques

Several Abaqus analysis approaches can be used to simulate the transient behavior of the scooter: using only Abaqus/Standard, using only Abaqus/Explicit, or using Abaqus/Standard to Abaqus/Explicit co-simulation. To illustrate the computational cost savings of the co-simulation approach, the analysis cases that follow focus on comparing the co-simulation approach to an Abaqus/Explicit-only simulation approach.

The Abaqus/Standard model of the co-simulation analysis consists of the scooter deck (Figure 2.4.1–2) and frame (Figure 2.4.1–3). The deck is modeled using substructure techniques to further reduce the solution cost. The Abaqus/Explicit model of the co-simulation analysis consists of the tires and the road with the bump (Figure 2.4.1–4). Co-simulation regions across which data will be exchanged during the co-simulation analysis are identified on each model at the location of the wheel axles.

Summary of analysis cases

Case 1 A reference analysis performed using only Abaqus/Explicit.

Case 2 A co-simulation analysis using the subcyling coupling scheme.

Both analysis cases address the same transient simulation.

Analysis types

The tire transient response and contact with the road are simulated using the explicit dynamics procedure. In Case 1 the explicit dynamics procedure applies to the rest of the model as well. The scooter frame transient response is simulated using the implicit dynamics procedure.

Analysis techniques

Static stabilization, substructuring, and import analysis techniques are used in this example.

Static stabilization

The inflation of the tires uses the Abaqus/Standard static stabilization option.

Substructuring

The scooter deck is expected to experience small strains during the analysis, allowing the use of a substructure to model the deck as a technique to lower computational cost.

Import

The tires are inflated in an Abaqus/Standard static procedure. The inflated state and configuration of the tires are imported into Abaqus/Explicit for the subsequent transient analysis.

Mesh design

The meshed models are shown in Figure 2.4.1–5. The frame components are modeled using truss and connector elements. The deck is modeled using S4R shell elements. The tires are modeled using C3D8I continuum elements. The operator is represented by a point mass connected to foot and hand-hold locations through a distributing coupling constraint definition.

Material model

The deck is modeled as a simple linear elastic material, with an elastic modulus of 5 GPa, a Poisson's ratio of 0.3, and a density of 5000 kg/m^3.

The tires are modeled as a hyperelastic material with viscoelastic properties. The tire material density is 1100 kg/m^3.

Hyperelastic material constants	
C_{10}	1 MPa
C_{01}	0.0 Pa
D_1	5.085×10^{-8} Pa

Viscoelastic material constants	
$\bar{g}_1^P$	0.3
$\bar{k}_1^P$	0
τ_1	0.1

Initial conditions

An initial tire pressure of 20 KPa is applied. The tires begin the analysis in a statically equilibrated footprint configuration as a result of import from an earlier static analysis. The scooter is traveling toward the bump at an initial velocity of 3 m/s.

Boundary conditions

The scooter steering assembly is fixed in a straight configuration.

Loading

Forces are applied consistent with a total scooter weight of 42.4 N and a total operator weight of 222 N. To simplify the analysis setup, gravity loading is not applied; instead, the weight forces are applied at

the axle locations. The main consequence of this loading approach is that the static sag of the deck due to the operator is neglected.

Constraints

The scooter deck is attached to the frame using tie constraints.

Interactions

Contact interactions define the tire contact with the road and bump.

Case 1 Abaqus/Explicit reference analysis

This analysis case uses Abaqus/Explicit exclusively for the transient analysis and is provided as a reference for comparing the results and computational cost of the co-simulation solution.

Analysis types

Abaqus/Explicit is used for the transient analysis with the tire inflation occurring using the static procedure in Abaqus/Standard.

Analysis steps

The following analysis step types are used.

Static analysis

An Abaqus/Standard static procedure is used to inflate the tires and establish the tire footprint.

Explicit dynamics

The Abaqus/Explicit step does not employ mass scaling. The default bulk viscosity parameters are used.

Results and discussion

The simulation results show the scooter impacting the bump, the operator shifting forward slightly, and the deck flexing.

Case 2 Co-simulation analysis using subcycling

In this case co-simulation occurs between Abaqus/Standard and Abaqus/Explicit, with each program advancing its simulation time according to its own automatic incrementation scheme and exchanging data as needed. Co-simulation data are exchanged at each Abaqus/Explicit time increment.

Analysis types

Abaqus/Explicit is used for the transient analysis of the tires with the tire inflation occurring using the static procedure in Abaqus/Standard. Abaqus/Standard is used for the transient analysis of the scooter frame and deck.

Analysis steps

The following analysis step types are used.

Static analysis

An Abaqus/Standard static procedure is used to inflate the tires and establish the tire footprint.

Explicit dynamics

The Abaqus/Explicit step does not employ mass scaling. The default bulk viscosity parameters are used.

Substructure generation

The substructure generation of the scooter deck uses 100 retained eigmodes, obtained using the Lanczos eigensolver.

Implicit dynamics

The Abaqus/Standard transient dynamic steps use a half-increment residual limit setting of 200 N.

Solution controls

Co-simulation controls are used to specify a subcycling approach.

Run procedure

The analyses in the co-simulation are run concurrently. The coupling is achieved using the Abaqus co-simulation execution procedure (see "Abaqus/Standard, Abaqus/Explicit, and Abaqus/CFD co-simulation execution," Section 3.2.4 of the Abaqus Analysis User's Guide).

For example, use the following command to run the Abaqus/Standard and Abaqus/Explicit jobs:

```
abaqus cosimulation cosimjob=scooter
   job=scooter_cosim_std,scooter_cosim_xpl
   oldjob=NONE,scooter_tire_inflation
   configure=scooter_cosim_config
```

Results and discussion

The results for a co-simulation analysis appear in multiple output files. To make effective use of the Abaqus/Standard and Abaqus/Explicit output databases, you should use the Abaqus/Viewer overlay functionality to view the combined results. For more information, see Chapter 79, "Overlaying multiple plots," of the Abaqus/CAE User's Guide.

Discussion of results and comparison of cases

The results show that Case 1 is more computationally expensive. Case 2 provides a significant cost improvement. Further, the results show that the solution fidelity, when compared to the reference solution, is not significantly affected by the use of the co-simulation technique.

Final configuration of the assembly

As the scooter collides with the bump, the entire assembly leaves the ground and flexes slightly, as shown in Figure 2.4.1–6.

Dynamic response of the operator

We consider measures of operator comfort to relate to the displacement, velocity, and acceleration magnitude histories at the operator node location. Figure 2.4.1–7, Figure 2.4.1–8, and Figure 2.4.1–9 show these respective response histories, comparing the results for the Case 1 and Case 2 analyses. The results are plotted after applying the Butterworth filter with a cutoff frequency of 100 Hz (see "Applying Butterworth filtering to an *X*–*Y* data object," Section 47.4.26 of the Abaqus/CAE User's Guide, in the HTML version of this guide) and show very good agreement between the Case 1 and Case 2 workflows.

Computational cost

Table 2.4.1–1 lists the relative computation cost of the two simulation approaches and clearly shows the value of co-simulation in this analysis.

Files

Common files

scooter_tire_inflation.inp	Abaqus/Standard input file to inflate the tires and establish the static footprint due to the assembly weight.
scooter_parameters.inp	Job parameters common to all scooter analyses.

Case 1 Abaqus/Explicit reference analysis

scooter_xpl.inp	Abaqus/Explicit input file to model all components, importing the inflated tires from scooter_tire_inflation.inp results and simulating the transient impact with the bump.

Case 2 Co-simulation analysis using subcycling

Co-simulation configuration

scooter_cosim_config.xml	Co-simulation configuration file defining the subcycling algorithm.

Tire and road modeling

scooter_cosim_xpl.inp	Abaqus/Explicit input file to model the tires and road, importing the inflated tires from scooter_tire_inflation.inp results and simulating the transient impact with the bump through co-simulation coupling with scooter_cosim_std.inp.

Frame and deck modeling

scooter_subgen.inp	Abaqus/Standard input file modeling the deck and creating the substructure representation of the deck.
scooter_cosim_std.inp	Abaqus/Standard input file modeling the frame components, referring to the scooter_subgen.inp substructure definition and simulating the transient impact with the bump through co-simulation coupling with scooter_cosim_xpl.inp.

References

Abaqus Analysis User's Guide

- "Co-simulation: overview," Section 17.1.1 of the Abaqus Analysis User's Guide
- "Preparing an Abaqus analysis for co-simulation," Section 17.2.1 of the Abaqus Analysis User's Guide
- "Structural-to-structural co-simulation," Section 17.3.1 of the Abaqus Analysis User's Guide

Abaqus Keywords Reference Guide

- *CO-SIMULATION
- *CO-SIMULATION CONTROLS
- *CO-SIMULATION REGION

Abaqus Verification Guide

- "Abaqus/Standard to Abaqus/Explicit co-simulation," Section 3.21.2 of the Abaqus Verification Guide

Table 2.4.1–1 Comparison of relative CPU times (normalized with respect to the CPU time for the Abaqus/Explicit analysis).

Analysis job	Relative CPU time	
	Co-simulation workflow	Abaqus/Explicit workflow
Substructure generation	0.007	N/A
Tire inflation and footprint	0.002	0.002
Co-simulation Abaqus/Explicit analysis	0.060	N/A
Co-simulation Abaqus/Standard analysis	0.057	N/A
Complete Abaqus/Explicit analysis	N/A	1.0
Total simulation cost	0.126	1.002

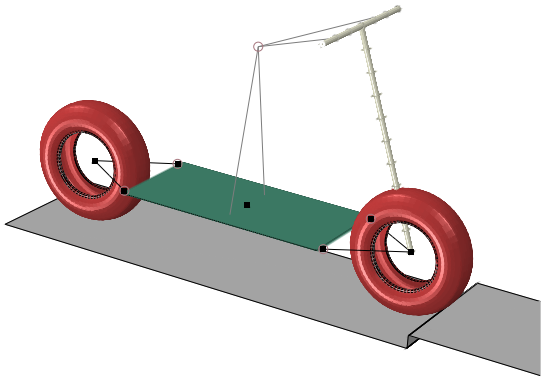


Figure 2.4.1–1 Scooter assembly geometry. The operator is modeled as a point mass.

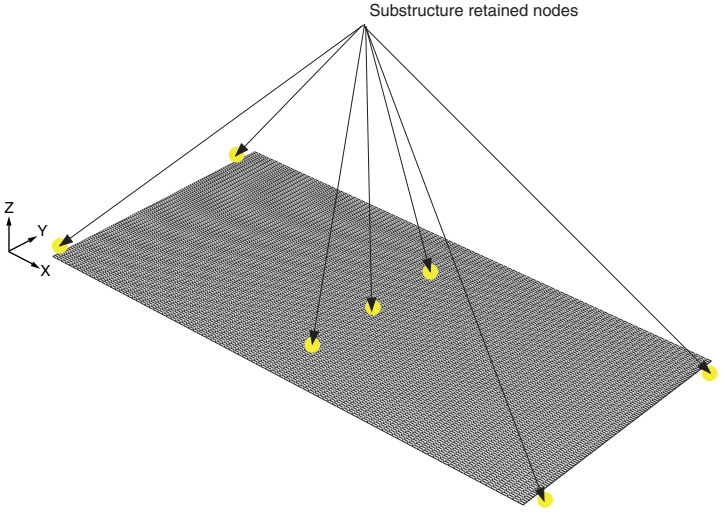


Figure 2.4.1–2 Deck analysis region. Substructure retained nodes represent the attachment locations to the scooter frame, the operator foot locations, and a center location used to constrain roll about the x-direction for the global model.

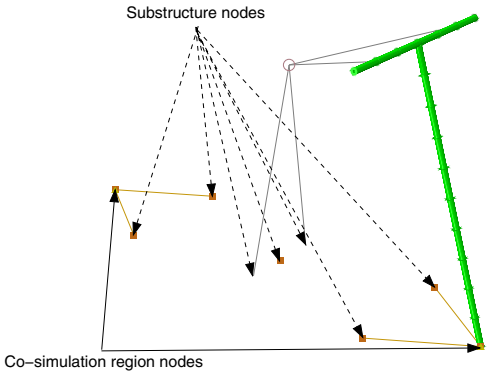


Figure 2.4.1–3 Frame analysis region.

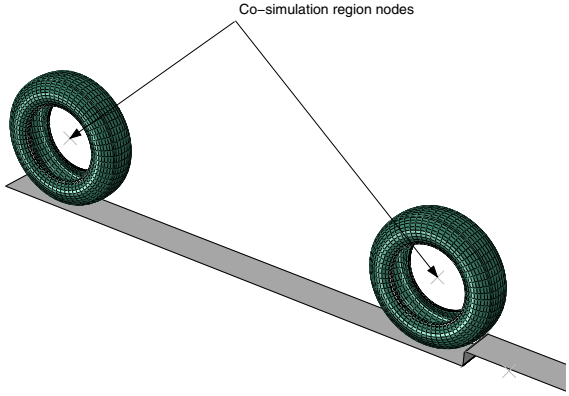


Figure 2.4.1–4 Tire and road analysis region.

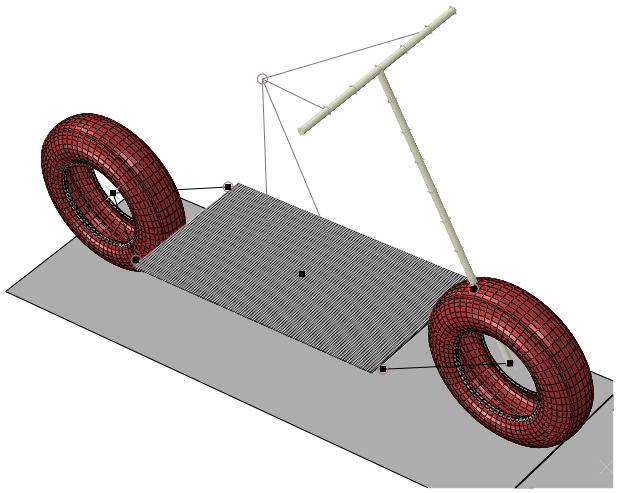


Figure 2.4.1–5 The complete scooter assembly mesh.

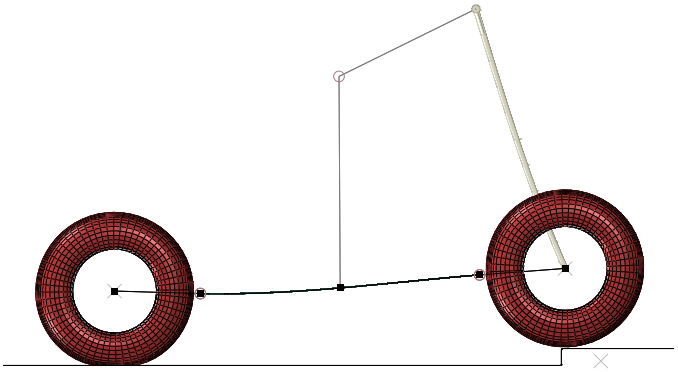


Figure 2.4.1–6 Deformed configuration after colliding with the bump.

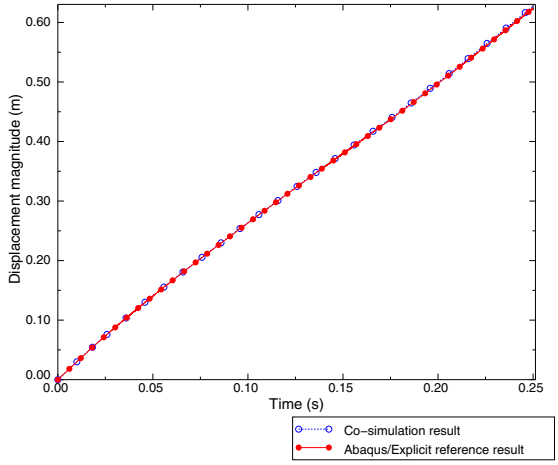


Figure 2.4.1–7 Operator displacement magnitude history.

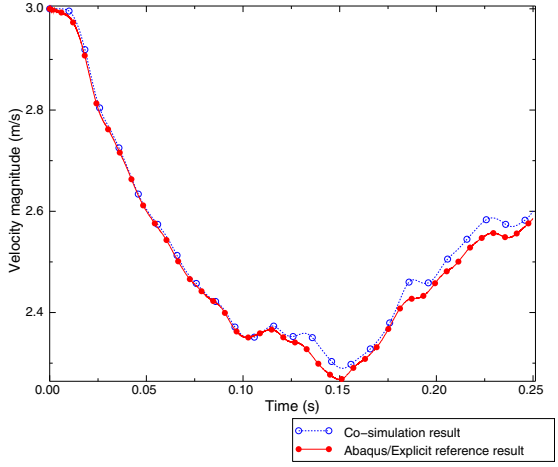


Figure 2.4.1–8 Operator velocity magnitude history.

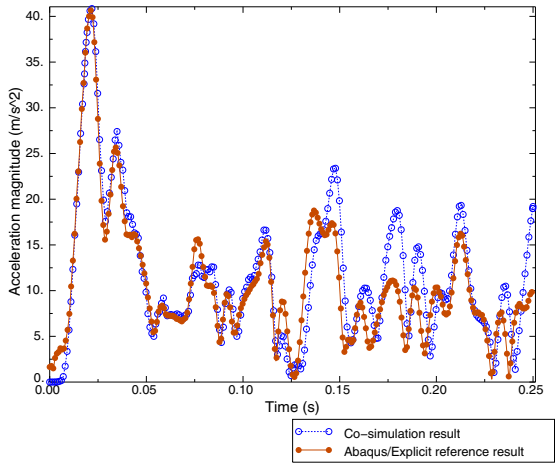


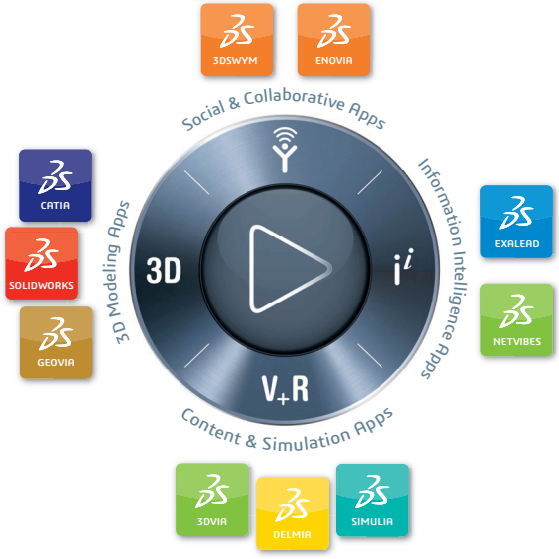
Figure 2.4.1–9 Operator acceleration magnitude history.

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