

ABAQUS 6.14

RELEASE NOTES



3DEXPERIENCE

Abaqus

Release Notes

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.

Contents

1. Introduction to Abaqus 6.14

Key features of Abaqus 6.14	1.1
Abaqus products	1.2
Changes in interpretation of input data	1.3
Changes in interpretation of input data for Abaqus/CAE	1.4

2. General enhancements

Accessing the Learning Community from Abaqus/CAE	2.1
--	-----

3. Modeling

Sizing optimization	3.1
Optimization across multiple models	3.2
Viewing the progression of an optimization by combining output database files	3.3
Monitoring an optimization process	3.4
Additional criteria for creating soft elements	3.5
New design responses	3.6
Specifying the rate at which optimization data are saved	3.7
Writing optimization files	3.8
Switching the context for model instances in Abaqus/CAE	3.9
Associating geometric faces created from an orphan mesh with the mesh	3.10
Bidirectional import of parameters using the CATIA V6 Associative Interface	3.11

4. Analysis procedures

The AMS eigensolver can extract coupled structural-acoustic eigenmodes	4.1
AMS eigensolver performance improvements	4.2
Specifying additional volume in a fluid cavity	4.3
K-epsilon realizable turbulence model in Abaqus/CFD	4.4
Matrix quality check	4.5
Flexible body generation	4.6

5. Analysis techniques

Defining a local coordinate system for materials in Eulerian elements	5.1
Enhancement to adaptive mesh refinement	5.2
Enhancements to the XFEM-based crack propagation capability	5.3
Enhancements to the XFEM-based crack propagation capability in Abaqus/CAE	5.4
Parallel enhancement for DEM analysis	5.5

CONTENTS

Enhancements to CZone for Abaqus for composite crush simulation	5.6
Vehicle ride comfort and durability co-simulation	5.7
Logical-Physical co-simulation using Abaqus and Dymola	5.8
Enhancements to the SIMULIA Co-Simulation Engine	5.9
SIMULIA Co-Simulation Engine API availability	5.10
Enhancements for co-simulation	5.11
MpCCI connector to the SIMULIA Co-Simulation Engine	5.12
6. Materials	
Enhancements to parallel rheological framework	6.1
Enhancements to creep models	6.2
User-defined frequency domain viscoelastic behavior	6.3
New hybrid formulations for user-defined material behavior	6.4
Different stiffness in tension versus compression for traction-separation elasticity	6.5
Eulerian elements with anisotropic materials	6.6
Element deletion controlled by state variables	6.7
7. Elements	
Support for coupled temperature–pore pressure elements in Abaqus/CAE	7.1
Linear pyramid acoustic element	7.2
8. Prescribed conditions	
Initial conditions on damage initiation	8.1
Enhancements for prescribed conditions in Abaqus/CFD	8.2
Acoustic base motion for SIM-based procedures	8.3
9. Constraints	
New attachment method for mesh-independent fasteners	9.1
Enhanced domain decomposition of thermal ties	9.2
10. Interactions	
Edge-to-edge contact for general contact in Abaqus/Standard	10.1
Improved contact at complex intersections in Abaqus/Explicit	10.2
Enhancement to the contact detection tool in Abaqus/CAE	10.3
11. Execution	
Job execution control enhancements	11.1
GPGPU acceleration of the AMS eigensolver	11.2
GPGPU accelerated direct equation solver	11.3
User control of domain decomposition	11.4

SIM database utilities	11.5
Enhancements for translating Abaqus data to modal neutral files	11.6
12. Output and visualization	
Output of mode mix ratios during mixed mode delamination	12.1
Output of last converged increment	12.2
Output of contact-related energies	12.3
Magnitude output of vector nodal variables	12.4
Equivalent plastic strain rate	12.5
Principal stresses with bounding values for output field filtering	12.6
Writing results from an Abaqus analysis to a SIM database	12.7
Requesting reference sink temperature and film coefficient field output in Abaqus/CAE	12.8
Generating tabular reports for free body cuts from multiple frames	12.9
Selecting the area centroid as the summation point for free body cuts	12.10
Reading <i>X–Y</i> data from free bodies defined on view cuts	12.11
Performing display group Boolean operations on linked viewports	12.12
13. User subroutines, utilities, and plug-ins	
VEEXTERNALDB: User subroutine that gives control to the user at key moments of the analysis	13.1
UCHARLENGTH: User subroutine to define the characteristic element length at a material point	13.2
Enhancements to user subroutine UMAT	13.3
Allocatable arrays	13.4
Parallel user subroutines	13.5
14. Abaqus Scripting Interface	
Enhancements to the <i>addData</i> method	14.1
Enhancement to the <i>nodes</i> member of the OdbSet object	14.2
15. Summary of changes	
Changes in Abaqus elements	15.1
Changes in Abaqus options	15.2
Changes in Abaqus user subroutines	15.3
Changes in Abaqus output variable identifiers	15.4
I.1 Product Index	



1. Introduction to Abaqus 6.14

This document introduces features in Abaqus that have been added, enhanced, or updated since the Abaqus 6.13 release.

Chapter 1 provides a brief overview of the Abaqus products included in this release. Chapters 2–14 provide short descriptions of new Abaqus 6.14 features in Abaqus/Standard, Abaqus/Explicit, Abaqus/CFD, and Abaqus/CAE, categorized by subject:

- Chapter 2, “General enhancements”: general changes to the Abaqus interface.
- Chapter 3, “Modeling”: features related to creating your model.
- Chapter 4, “Analysis procedures”: features related to defining an analysis.
- Chapter 5, “Analysis techniques”: features related to analysis techniques in Abaqus.
- Chapter 6, “Materials”: new material models or changes to existing material models.
- Chapter 7, “Elements”: new elements or changes to existing elements.
- Chapter 8, “Prescribed conditions”: loads, boundary conditions, and predefined fields.
- Chapter 9, “Constraints”: kinematic constraints.
- Chapter 10, “Interactions”: features related to contact and interaction modeling.
- Chapter 11, “Execution”: commands and utilities for running any of the Abaqus products.
- Chapter 12, “Output and visualization”: obtaining, postprocessing, and visualizing results from Abaqus analyses.
- Chapter 13, “User subroutines, utilities, and plug-ins”: additional user programs that can be run with Abaqus.
- Chapter 14, “Abaqus Scripting Interface”: using the Abaqus Scripting Interface to write user scripts.

Each entry in these chapters clearly indicates the Abaqus product or products to which the feature applies and includes cross-references to more detailed information. Chapter 15, “Summary of changes,” summarizes in tabular format the changes to Abaqus elements, keyword options, user subroutines, and output variable identifiers.

1.1 Key features of Abaqus 6.14

This section provides a list of the most significant new capabilities and enhancements available in Abaqus 6.14; refer to the table of contents for a complete list of new features.

- **General performance improvements:**
 - GPGPU accelerated direct equation solver
 - User control of domain decomposition
 - *addData* method enhancements



- **Contact enhancements:**
 - Edge-to-edge contact for general contact in Abaqus/Standard
 - Improved contact at complex intersections in Abaqus/Explicit
- **Output:**
 - Abaqus output database in SIM format for use in the Physics Results Explorer app on the 3DEXPERIENCE Platform
- **AMS eigensolver:**
 - GPGPU acceleration performance improvements
 - Constraint equation handling performance improvements
 - Coupled structural-acoustic eigenmodes extraction
- **Materials:**
 - Parallel rheological framework enhancements
 - Local coordinate system for materials in Eulerian elements
 - Use of anisotropic materials in Eulerian elements
 - Initial conditions on damage initiation
- **Crack modeling and propagation:**
 - XFEM enhancements
- **Fluid analysis:**
 - $k-\varepsilon$ realizable turbulence model
 - Prescribed conditions for turbulence models
- **Co-simulation:**
 - SIMULIA Co-Simulation Engine API general availability
- **Optimization:**
 - Sizing optimization
 - Optimization across multiple models
 - Combining output database files
 - New design responses
- **Abaqus/CAE geometry and modeling:**
 - Bidirectional support for CATIA V6 Associative Interface
 - Associate geometric faces with mesh
 - Switch context for model instances
- **Visualization:**
 - Display group operations on linked viewport



- Free body cut summation at area centroid
- X – Y data from free bodies defined on view cuts
- Free body reports from multiple frames

The remaining chapters in this book provide details on these and other new features of Abaqus 6.14. In addition to the enhancements listed here, most of the known bugs in Abaqus 6.13 are corrected.

1.2 Abaqus products

Individual components of the Abaqus suite are described in this section.

Analysis

- **Abaqus/Standard:** This general-purpose analysis product can solve a wide range of linear and nonlinear problems involving the static, dynamic, thermal, electrical, and electromagnetic response of components. Abaqus/Standard includes all analysis capabilities except those provided in the Abaqus/Explicit and Abaqus/CFD programs and the add-on analysis functionality described below.
- **Abaqus/Explicit:** This product provides nonlinear, transient, dynamic analysis of solids and structures using explicit time integration. Its powerful contact capabilities, reliability, and computational efficiency on large models also make it highly effective for quasi-static applications involving discontinuous nonlinear behavior.
- **Abaqus/CFD:** This product is a computational fluid dynamics program with support for preprocessing, simulation, and postprocessing in Abaqus/CAE. Abaqus/CFD provides scalable parallel CFD simulation capabilities to address a number of nonlinear coupled fluid-thermal and fluid-structural problems.

Preprocessing and postprocessing

- **Abaqus/CAE:** This product is a Complete Abaqus Environment that provides a simple, consistent interface for creating, submitting, monitoring, and evaluating results from Abaqus simulations. Abaqus/CAE is divided into modules, where each module defines a logical aspect of the modeling process; for example, defining the geometry, defining material properties, generating a mesh, submitting analysis jobs, and interpreting results.
- **Abaqus/Viewer:** This subset of Abaqus/CAE contains only the postprocessing capabilities of the Visualization module. It uses the output database (**.odb**) to obtain results from the analysis products. The output database is a neutral binary file. Therefore, results from an Abaqus analysis run on any platform can be viewed on any other platform supporting Abaqus/Viewer. It provides deformed configuration, contour, vector, and X – Y plots, as well as animation of results.



Add-on analysis

- **Abaqus/Aqua:** This add-on analysis capability for Abaqus/Standard and Abaqus/Explicit provides a capability for calculating drag and buoyancy loads based on steady current, wave, and wind effects for modeling offshore piping and floating platform structures. Abaqus/Aqua is applicable for structures that can be idealized using line elements, including beam, pipe, and truss elements.
- **Abaqus/Design:** This add-on analysis capability for Abaqus/Standard allows the user to perform design sensitivity analysis (DSA). The derivatives of output variables are calculated with respect to specified design parameters.
- **Optimization Module:** This capability is available in Abaqus/CAE to perform shape and topology optimization. This functionality requires an additional license to submit an optimization process for analysis.
- **Abaqus/Foundation:** This analysis option offers more efficient access to the linear static and dynamic analysis functionality in Abaqus/Standard.
- **CZone for Abaqus:** This add-on capability for Abaqus/Explicit provides access to a state-of-the-art methodology for crush simulation based on CZone technology from Engenuity, Ltd. Targeted toward the design of composite components and assemblies, CZone for Abaqus provides for inclusion of material crush behavior in simulations of composite structures subjected to impact.

Optional analysis functionality

- **Abaqus/AMS:** This add-on analysis capability for Abaqus/Standard allows the user to select the automatic multi-level substructuring (AMS) eigensolver when performing a natural frequency extraction.
- **Co-simulation with MpCCI:** This add-on analysis capability for Abaqus can be used to solve multiphysics problems by coupling Abaqus with any third-party analysis program that supports the MpCCI interface.

Associative interfaces and geometry translators

- **SIMULIA Associative Interface for Abaqus/CAE:** This add-on capability for Abaqus/CAE creates a connection between a CATIA V6 session and an Abaqus/CAE session. This connection can be used to transfer model information from CATIA V6 to Abaqus/CAE. Subsequent modifications to the model in CATIA V6 can be propagated to the Abaqus/CAE model while retaining any analysis features (such as loads or boundary conditions) that were defined on the model in Abaqus/CAE. The CATIA V6 model in an assembly file (**.eaf**) format can also be imported directly into Abaqus/CAE.
- **CATIA V5 Associative Interface:** This add-on capability for Abaqus/CAE creates a connection between a CATIA V5 session and an Abaqus/CAE session. This connection can be used to transfer model information from CATIA V5 to Abaqus/CAE. Subsequent modifications to the model in CATIA V5 can be propagated to the Abaqus/CAE model while retaining any analysis features (such as loads or boundary conditions) that were defined on the model in Abaqus/CAE. The geometry of



CATIA V5-format Part (**.CATPart**) and Product (**.CATProduct**) files can also be imported directly into Abaqus/CAE.

- **SolidWorks Associative Interface:** This add-on capability for Abaqus/CAE creates a connection between a SolidWorks session and an Abaqus/CAE session. This connection can be used to transfer model information from SolidWorks to Abaqus/CAE. Subsequent modifications to the model in SolidWorks can be propagated to the Abaqus/CAE model while retaining any analysis features (such as loads or boundary conditions) that were defined on the model in Abaqus/CAE.
- **Pro/ENGINEER Associative Interface:** This add-on capability for Abaqus/CAE creates a connection between a Pro/ENGINEER session and an Abaqus/CAE session. This connection can be used to transfer model information between Pro/ENGINEER and Abaqus/CAE. Modifications to the model in Pro/ENGINEER can be propagated to the Abaqus/CAE model without affecting any analysis features (such as loads or boundary conditions) that were defined on the model in Abaqus/CAE, and certain geometric modifications can be made in Abaqus/CAE and propagated to the model in Pro/ENGINEER.
- **Abaqus/CAE Associative Interface for NX:** This add-on capability for Abaqus/CAE creates a connection between an NX session and an Abaqus/CAE session. This connection can be used to transfer model data and to propagate design changes between NX and Abaqus/CAE. The Abaqus/CAE Associative Interface for NX can be purchased and downloaded from Elysium Inc. (www.elysiuminc.com).
- **Geometry Translator for CATIA V4:** This add-on capability allows the user to import the geometry of CATIA V4-format parts and CATIA V4 assemblies (**.model**, **.catdata**, and **.exp** files) directly into Abaqus/CAE.
- **Geometry Translator for Parasolid:** This add-on capability allows the user to import the geometry of Parasolid-format parts and Parasolid assemblies (**.x_t**, **.x_b**, and **.xmt** files) directly into Abaqus/CAE.

Translator utilities

- Abaqus translators are provided with the release. They are invoked through the Abaqus execution procedure (the “driver”). The translators and the commands to invoke them are described below:
 - abaqus fromansys** translates an ANSYS input file to an Abaqus input file.
 - abaqus fromdyna** translates an LS-DYNA keyword file to an Abaqus input file.
 - abaqus fromnastran** translates a Nastran bulk data file to an Abaqus input file.
 - abaqus frompamcrash** translates a PAM-CRASH input file to a partial Abaqus input file.
 - abaqus fromradioss** translates a RADIOSS input file to a partial Abaqus input file.
 - abaqus adams** translates the results in an Abaqus SIM database file into an MSC.ADAMS modal neutral (**.mnf**) file, the format required by ADAMS/Flex.
 - abaqus moldflow** translates finite element model information from a Moldflow analysis into a partial Abaqus input file.



abaqus tonastran translates an Abaqus input file to Nastran bulk data file format.

abaqus toOutput2 translates an Abaqus output database file to the Nastran Output2 file format.

abaqus tozaero enables you to exchange aeroelastic data between the Abaqus and ZAERO analysis products.

Other utilities

- Additional programs are included with the release. They are all invoked through the Abaqus execution procedure (the “driver”). The utilities and the commands to invoke these programs are described below:

abaqus append joins separate results files into a single file.

abaqus ascfil translates Abaqus results files between ASCII and binary formats, which is useful for moving results files between different computer types.

abaqus cosimulation runs a co-simulation using a single command where the analysis job options specify two Abaqus jobs.

abaqus cse runs the SIMULIA Co-Simulation Engine (CSE) Director process that governs co-simulation between Abaqus and third-party solvers. Typically, when performing a co-simulation between Abaqus solvers only, you are not required to invoke the CSE Director process; it is invoked automatically when you run the Abaqus co-simulation procedure using **abaqus cosimulation**.

abaqus doc accesses the Abaqus documentation collection using a web browser.

abaqus dymola runs a co-simulation between an Abaqus/Standard or Abaqus/Explicit model and a model exported from Dymola.

abaqus emloads converts results output from an electromagnetic analysis for use as loads in a subsequent analysis.

abaqus encrypt creates an encoded, password-protected version of an Abaqus input file, while **abaqus decrypt** converts an encrypted file back into its original, unencoded format.

abaqus fetch extracts example input files from the libraries included with the release.

abaqus findkeyword provides a list of sample problems that use the specified Abaqus options. This utility will help users find examples of features they may be using for the first time.

abaqus free converts all fixed format data in an input file to free format.

abaqus licensing provides management and monitoring tools for FLEXnet and Dassault Systèmes (DS) licensing.

abaqus make compiles and links user-written postprocessing programs for Abaqus and creates user-defined libraries of Abaqus/Standard and Abaqus/Explicit user subroutines.

abaqus mtxasm assembles element matrices contained in a SIM document and, optionally, writes the assembled matrices to text files.

abaqus networkDBConnector creates a connection to a network ODB server that can be used to access a remote output database.

abaqus restartjoin appends an output database file produced by a restart analysis of a model to the output database produced by the original analysis of that model.



abaqus odbcombine combines the results data in two or more Abaqus output database files into a single output database file.

abaqus odbreport creates organized reports of output database information in text, HTML, or CSV file formats.

abaqus python accesses the Python interpreter.

abaqus resume resumes an Abaqus analysis job.

abaqus script initiates a Python scripting session.

abaqus sim_version converts a SIM database file from one release to another, queries a SIM database file for its SIM release level, or determines the SIM release level of the current code you are using.

abaqus substructurecombine combines the model and results data produced by two of a model's substructures into a single output database file.

abaqus suspend suspends an Abaqus analysis job.

abaqus terminate terminates an Abaqus analysis job.

abaqus upgrade upgrades an input file or output database file from previous versions of Abaqus to the current version.

Platform support

Analysis products (Abaqus/Standard, Abaqus/Explicit, and Abaqus/CFD) and interactive products (Abaqus/CAE and Abaqus/Viewer) are supported on the following platforms:

- Windows/x86-64
- Linux/x86-64

Changes to documentation

Because the translation functionality in the Abaqus Interface for Moldflow has been integrated into Abaqus/Standard as the **abaqus moldflow** execution procedure, the Abaqus Interface for Moldflow User's Guide has been removed from the Abaqus documentation collection. For information about running the **abaqus moldflow** execution procedure, see "Translating Moldflow data to Abaqus input files," Section 3.2.37 of the Abaqus Analysis User's Guide. For translation examples, see "Moldflow translation examples," Section 1.3.19 of the Abaqus Example Problems Guide.

1.3 Changes in interpretation of input data

A list of changes to the Abaqus input file interface is provided in Chapter 15, "Summary of changes."



1.4 Changes in interpretation of input data for Abaqus/CAE

The following change in Abaqus/CAE 6.14 is a change from previous releases:

- Python interprets numbers with leading zeros as octal numbers (for example, **0123** is interpreted as **83.0**). Previously, numbers with leading zeros entered in text fields were interpreted as octal numbers. Now, Abaqus/CAE will ignore leading zeros in numbers in text fields before Python interprets them; such numbers are evaluated as decimals. For more information, see “Entering expressions,” Section 3.2.2 of the Abaqus/CAE User’s Guide.



2. General enhancements

This chapter describes the following general enhancement that has been made to Abaqus:

- “Accessing the Learning Community from Abaqus/CAE,” Section 2.1

2.1 Accessing the Learning Community from Abaqus/CAE

Product: Abaqus/CAE

Benefits: Abaqus/CAE provides easy access to the Learning Community.

Description: You can now access the Learning Community (<http://www.3ds.com/products-services/simulia/academics/simulia-learning-community>) from the main menu bar in Abaqus/CAE. The Learning Community contains online tutorials and technical content. The community also hosts a question-and-answer area that enables the global community of users to share their expertise and learn how to leverage the latest features and enhancements available in the SIMULIA portfolio.

Abaqus/CAE Usage:

All modules:

Help→Learning Community

Reference:

Abaqus/CAE User's Guide

- “Accessing the Learning Community,” Section 2.6.5



3. Modeling

This chapter discusses features related to creating your model, such as part and assembly definition in Abaqus/CAE, importing and exporting models to or from Abaqus/CAE, and optimization. It provides an overview of the following enhancements:

- “Sizing optimization,” Section 3.1
- “Optimization across multiple models,” Section 3.2
- “Viewing the progression of an optimization by combining output database files,” Section 3.3
- “Monitoring an optimization process,” Section 3.4
- “Additional criteria for creating soft elements,” Section 3.5
- “New design responses,” Section 3.6
- “Specifying the rate at which optimization data are saved,” Section 3.7
- “Writing optimization files,” Section 3.8
- “Switching the context for model instances in Abaqus/CAE,” Section 3.9
- “Associating geometric faces created from an orphan mesh with the mesh,” Section 3.10
- “Bidirectional import of parameters using the CATIA V6 Associative Interface,” Section 3.11

3.1 Sizing optimization

Product: Abaqus/CAE

Benefits: You can now create a sizing optimization that optimizes a shell model by modifying the thickness of the shell elements.

Description: In addition to topology and shape optimization, you can now create a sizing optimization. Sizing optimization is applied to shell parts and optimizes the thickness of the shell elements in the design area. When you configure a sizing optimization, you can specify that selected regions should contain “clusters” of shell elements of equal thickness. You can use clustering to generate strengthening ribs or rings in the sheet metal structure you are optimizing or to define borders between regions of equal thickness. Clustered regions can be reproduced in manufacturing using sheets of constant thickness; for example, a vehicle “body in white” formed by welding and stamping individual sheet metal structures.

When you view the results of a sizing optimization, by default the Visualization module displays the thickness of the shell elements.

Abaqus/CAE Usage:

Optimization module:

Task→Create, Name: *optimization task name*, **Type:** Sizing optimization



MODELING

References:

Abaqus/CAE User's Guide

- “Creating an optimization task,” Section 18.6.1, in the HTML version of this guide
- “Configuring a sizing optimization task,” Section 18.6.4, in the HTML version of this guide

Abaqus Example Problems Guide

- “Sizing optimization of a gear shift control holder,” Section 11.3.1
- “Sizing optimization of a car door,” Section 11.3.2

3.2 Optimization across multiple models

Product: Abaqus/CAE

Benefits: You can now create an optimization design response that covers multiple models. This allows you to create an optimization with nonlinear load cases.

Description: You can incorporate multiple models into your optimization when linear perturbations about a base state are no longer sufficient as load cases. For example, you can simulate nonlinear load cases (which are not supported by Abaqus/CAE) by creating multiple copies of your nonlinear model and by creating a step in each model during which different loads and boundary conditions are applied. Each model must have the same Abaqus/CAE geometry, the same mesh, and the same section assignments, and the models must be open in your Abaqus/CAE session.

Abaqus/CAE Usage:

Optimization module:

Design Response→**Create, Name:** *design response name*,

Type: Single-term, **Steps, Specify:** Model, Step and Load Case

References:

Abaqus/CAE User's Guide

- “Creating and editing a design response,” Section 18.7.1, in the HTML version of this guide
- “Selecting the data source of a design response,” Section 18.7.2, in the HTML version of this guide

3.3 Viewing the progression of an optimization by combining output database files

Product: Abaqus/CAE



Benefits: Each design cycle of an optimization process now creates a separate output database file. To view the results of the optimization in the Visualization module, you must combine the separate output database files with the optimization results into a single output database file.

Description: Previous releases of Abaqus/CAE stored the analysis results from each design cycle into a single output database file, resulting in very large output database files that consumed a lot of disk space and were slow to process. By default, Abaqus/CAE continues to save optimization data after every design cycle in optimization files. However, the individual output database files created during each design cycle are now saved only at specified intervals. Alternatively, you can control the rate at which analysis data is saved in output database files as follows:

- after every design cycle,
- from the initial, first, or last design cycles, or
- after every n design cycles.

To view the results of an optimization in the Visualization module, you must first combine the separate output database files and the optimization files into a single output database file. When you merge output database files, you can select the steps and load cases to merge; you can also select the field variables to include in the combined output database file. In addition, if your optimization process included multiple models, you can select the models to include; Abaqus/CAE creates a combined output database file for each model.

Abaqus/CAE Usage:

Job module:

Optimization→**Combine**, **Name:** *design response name*

Reference:

Abaqus/CAE User's Guide

- “Combining optimization results,” Section 19.12.8, in the HTML version of this guide

3.4 Monitoring an optimization process

Product: Abaqus/CAE

Benefits: You now have more flexibility when monitoring the progression of an optimization process.

Description: Abaqus/CAE can now display output data in a table that allows you to monitor the progression of an optimization process to ensure it is converging on an acceptable solution. You can select which variables to monitor, and you can display selected variables in an X - Y plot in a new viewport.

Abaqus/CAE Usage:

Job module:

Optimization→**Monitor**, **Name:** *task name*



MODELING

Reference:

Abaqus/CAE User's Guide

- “Monitoring your optimization process,” Section 19.12.6, in the HTML version of this guide

3.5 Additional criteria for creating soft elements

Product: Abaqus/CAE

Benefits: During a topology optimization, Abaqus creates soft elements that have a negligible influence on the stiffness of the resulting structure but are still relevant for the number of degrees of freedom of the structure. Choosing to delete soft elements is recommended when you are optimizing a nonlinear model because those elements would otherwise degenerate or collapse and prevent Abaqus from converging on a solution. Additional criteria have been added that define when an element is considered soft.

Description: You specify the criteria for creating soft elements when you create a topology optimization task. The additional criteria are:

- Maximum shear strain
- Minimum principal strain
- Maximum elastoplastic strain
- Volume compression

In addition, if you choose the standard or aggressive criteria, you can prevent isolated soft elements from being removed by choosing to delete only soft elements that have neighboring soft elements.

Abaqus/CAE Usage:

Optimization module:

Task→**Create**, **Name:** *task name*, **Type:** Topology optimization, **Advanced**

Reference:

Abaqus/CAE User's Guide

- “Configuring advanced options” in “Configuring a topology optimization task,” Section 18.6.2, in the HTML version of this guide

3.6 New design responses

Product: Abaqus/CAE

Benefits: You can now create scaled element centroidal von Mises stress and energy stiffness measure design responses.



Description: The energy stiffness measure design response is now available when you create general topology and sizing optimizations. In addition, the scaled, element centroidal, von Mises stress design response is now available when you create a general topology optimization. The energy stiffness measure has no physical meaning but can be used as an objective function or a constraint to optimize the stiffness of a structure that is subjected to both external loading and prescribed displacements.

Abaqus/CAE Usage:

Optimization module:

Design Response→**Create, Name:** *design response name*,

Type: Single-term, **Variable:** **Stress** or **Energy stiffness measure**

Reference:

Abaqus Analysis User's Guide

- “Design responses,” Section 13.2.1

3.7 Specifying the rate at which optimization data are saved

Product: Abaqus/CAE

Benefits: You now have more flexibility when specifying the rate at which the optimization process saves the files created by the Abaqus jobs, such as the status file, the message file, and the output database.

Description: With previous releases of Abaqus/CAE you could choose to save analysis data from the Abaqus job either every design cycle or from the first and last design cycles when creating an optimization process. You can now specify the rate at which optimization data are saved as follows:

- after every design cycle,
- from the initial, first, or last design cycles, or
- after every n design cycles.

Abaqus/CAE Usage:

Job module:

Optimization→**Create, Name:** *optimization process name*, **Optimization:** **Data save**

Reference:

Abaqus/CAE User's Guide

- “Creating and editing optimization processes,” Section 19.12.1, in the HTML version of this guide

3.8 Writing optimization files

Product: Abaqus/CAE



MODELING

Benefits: You now create the files that are needed to run the optimization with an Abaqus execution procedure from the command line.

Description: When you are managing an optimization process, you can create copies of the optimization parameter (**.par**) file and the Abaqus input (**.inp**) file in your working directory. The parameter file contains parameters that are used to execute the optimization and contains information about the input file that is associated with the optimization; the input file, in turn, defines the Abaqus model you are optimizing.

You can run the optimization from the command line with the following command:

```
abaqus optimization -task parameter file -job results folder
```

The parameter file and the input file must be saved in the same directory. During the optimization, Abaqus creates the results folder in which the results of the optimization are stored.

Abaqus/CAE Usage:

Job module:

Optimization→**Write Files**; **Name:** *optimization process name*

Reference:

Abaqus/CAE User's Guide

- “Creating the optimization files,” Section 19.12.2, in the HTML version of this guide

3.9 Switching the context for model instances in Abaqus/CAE

Product: Abaqus/CAE

Benefits: You can now use the Model Tree to switch the context for model instances and their child part instances, making it easier to access the original model or part for editing.

Description: You can select a model instance or a child part instance from the Model Tree and switch the context to the original model or part from which the selected instance was created.

Abaqus/CAE Usage:

Assembly module or Mesh module:

Select model or child part instance from the Model Tree, then click mouse button 3 and select **Switch to part/model context** from the menu.

Reference:

Abaqus/CAE User's Guide

- “Using the Model Tree to switch the context for part or model instances,” Section 13.10.3, in the HTML version of this guide



3.10 Associating geometric faces created from an orphan mesh with the mesh

Product: Abaqus/CAE

Benefits: When you create a geometric face from orphan element faces in a part, Abaqus/CAE now associates the new face with the mesh by default.

Description: You can use the Geometry Edit toolset to create geometry from the faces of orphan mesh elements. By default, the new geometric face is associated with the mesh; you have the option to change this behavior, as shown in Figure 3–1.

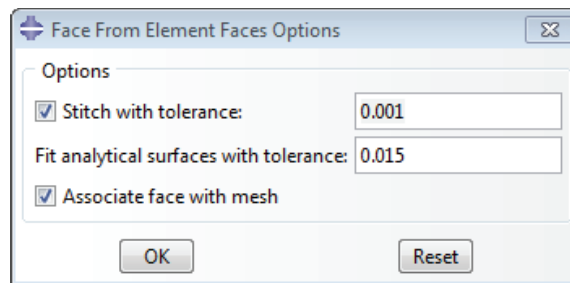


Figure 3–1 Options for creating geometric faces from orphan element faces.

Abaqus/CAE Usage:

Part module:

Tools→**Geometry Edit**; **Face: From element faces**; **Options:** toggle **Associate face with mesh**

Reference:

Abaqus/CAE User's Guide

- “Create face from element faces,” Section 69.7.10, in the HTML version of this guide

3.11 Bidirectional import of parameters using the CATIA V6 Associative Interface

Product: Abaqus/CAE

Benefits: After importing a model using the CATIA V6 Associative Interface, you can modify geometry parameters in the model. Both the Abaqus/CAE model and the original CATIA V6 model are updated based on the modified parameters, which allows you to keep your Abaqus/CAE and CATIA V6 models synchronized while you iterate on a design.



MODELING

Description: The parameter update capability for the CATIA V6 Associative Interface allows you to work exclusively in Abaqus/CAE after importing a model from CATIA V6 while keeping the original CATIA V6 model up to date with any geometric changes. Certain geometry parameters can be modified in Abaqus/CAE, then these modifications are propagated to the original CATIA V6 model (this functionality was previously available for only CATIA V5 and Pro/ENGINEER models).

To use bidirectional import, you must first specify the parameters and their values in the CATIA V6 model that will be imported into Abaqus/CAE. These parameters are used to define dimensions in the CATIA V6 model; for example, a parameter may be used to specify the radius of a hole feature. When you import the model into Abaqus/CAE using the CATIA V6 Associative Interface, the list of specified parameters is also imported. You use the **CAD Parameters** dialog box in Abaqus/CAE to modify the values of each parameter. When you click **Update**, the features associated with the modified dimensions are regenerated, and the model's geometry is updated in Abaqus/CAE and in the CATIA V6 model.

For detailed instructions on using bidirectional import, download the CATIA V6 Associative Interface User's Guide from the Dassault Systèmes Knowledge Base at www.3ds.com/support/knowledge-base.

Abaqus/CAE Usage:

Part module:

Tools→CAD Parameters.

Assembly module:

Tools→CAD Interfaces→CAD Parameters.

Reference:

Abaqus/CAE User's Guide

- "Updating geometry parameters in an imported model," Section 60.2



4. Analysis procedures

This chapter discusses features related to defining an analysis. It provides an overview of the following enhancements:

- “The AMS eigensolver can extract coupled structural-acoustic eigenmodes,” Section 4.1
- “AMS eigensolver performance improvements,” Section 4.2
- “Specifying additional volume in a fluid cavity,” Section 4.3
- “K-epsilon realizable turbulence model in Abaqus/CFD,” Section 4.4
- “Matrix quality check,” Section 4.5
- “Flexible body generation,” Section 4.6

4.1 The AMS eigensolver can extract coupled structural-acoustic eigenmodes

Product: Abaqus/Standard

Benefits: The coupled structural-acoustic eigenmodes can be computed by the AMS eigensolver and stored using the SIM architecture. In addition, the modal methods can utilize these modes for modal superposition.

Description: If the model includes structural-acoustic coupling, the AMS eigensolver can extract coupled modes. Previously, this functionality was available only with the Lanczos eigensolver. This enhancement allows you to take advantage of the superior performance offered by the AMS eigensolver. For example, extracting about 1,000 coupled modes of a 2.5 million degree-of-freedom car body model takes 3 hours and 3 minutes with the Lanczos eigensolver; the AMS eigensolver needs only 32 minutes, which is almost six times faster. This benchmarking was performed using 16 cores on a Sandy Bridge machine with 128 GB of memory.

The coupled structural-acoustic modes computed by the AMS eigensolver are supported in the following modal procedures that use the SIM architecture:

- complex eigenvalue extraction,
- mode-based steady-state dynamic analysis,
- subspace-based steady-state dynamic analysis, and
- substructure generation.

References:

Abaqus Analysis User's Guide

- “Natural frequency extraction,” Section 6.3.5



Abaqus Keywords Reference Guide

- *COMPLEX FREQUENCY
- *FREQUENCY
- *STEADY STATE DYNAMICS
- *SUBSTRUCTURE GENERATE

Abaqus Theory Guide

- “Coupled acoustic-structural medium analysis,” Section 2.9.1

4.2 AMS eigensolver performance improvements

Products: Abaqus/Standard Abaqus/AMS

Benefits: The performance of the AMS eigensolver has been improved for large models that contain many distributing coupling constraints or other features with Lagrange multipliers (for example, fasteners, contact interactions, connectors, or hyperelastic materials).

Description: Enhanced handling of constraint equations in the AMS eigensolver delivers improved performance for large-scale models with many constraints. Table 4–1 illustrates the performance improvement in the AMS eigensolver for one industrial vehicle model that has 18.6 million degrees of freedom and 1.3 million constraint equations. The frequency extraction analysis for the vehicle body model was run on a Linux machine with dual 2.6 GHz Intel Sandy Bridge processors (2×8 cores) and 128 GB of memory. In the table, the ‘Total Standard’ includes the times for the constraint solver, AMS eigensolver, and non-solver part of the AMS eigensolution procedure.

Table 4–1 Performance improvement in the AMS eigensolver.

	Abaqus 6.13 Wall Clock Time (sec.)	Abaqus 6.14 Wall Clock Time (sec.)	Speedup Factor
Constraint Solver	3027	52	58.2
AMS eigensolver	3264	3242	N/A
Total Standard	8362	5311	1.6

**References:****Abaqus Analysis User's Guide**

- “Natural frequency extraction,” Section 6.3.5

Abaqus Keywords Reference Guide

- *FREQUENCY

4.3 Specifying additional volume in a fluid cavity

Product: Abaqus/Standard

Benefits: You can now specify an additional volume for a fluid cavity in an Abaqus/Standard analysis.

Description: This feature, which was previously available only in Abaqus/Explicit, is now available in Abaqus/Standard as well. You can specify an additional volume that will be added to the actual volume of the cavity.

References:**Abaqus Analysis User's Guide**

- “Fluid cavity definition,” Section 11.5.2

Abaqus Keywords Reference Guide

- *FLUID CAVITY

4.4 K-epsilon realizable turbulence model in Abaqus/CFD

Product: Abaqus/CFD

Benefits: You can now apply the k - ϵ realizable turbulence model to fluid flow problems.

Description: The k - ϵ realizable model is a two-equation turbulence model that evolves an equation for the turbulent kinetic energy, k , and the energy dissipation rate, ϵ . The model equations are developed from fundamental physical principles and dimensional analysis; the equation for k is derived using first principles, and the equation for ϵ is postulated using physical insight. This particular version uses realizability constraints, which imposes mathematical consistency in the Reynolds stresses (such as enforcing the positivity of the normal stresses and the Cauchy-Schwarz inequality) to modify the model coefficients and the epsilon equation. These modifications guarantee the physical consistency in the predicted Reynolds stresses. To increase the accuracy of the k - ϵ realizable model in cases where the mesh is not fine enough to resolve the viscous sub-layer in wall-bounded flows, a near-wall model, known as the two-layer model, is implemented.



References:

Abaqus Analysis User's Guide

- “Incompressible fluid dynamic analysis,” Section 6.6.2

Abaqus Keywords Reference Guide

- *TURBULENCE MODEL

4.5 Matrix quality check

Product: Abaqus/Standard

Benefits: You can now check the quality of the generated stiffness and mass matrices in matrix generation or substructure generation analyses.

Description: You can request “rigid body mode” checks for the generated stiffness and mass matrices in the matrix generation or substructure generation procedures. The quality of the generated global assembled matrices is checked in the matrix generation procedure. The quality of the generated condensed substructure matrices is checked in the substructure generation procedure. The matrix check generates six “artificial” rigid body modes and projects the matrices onto the rigid body modal subspace. It is expected that the projected 6×6 stiffness matrix (also known as the rigid body energy matrix) is close to zero in the absence of the boundary conditions and constraints. To perform the stiffness matrix quality check, the boundary conditions and multipoint constraints can be suppressed in the matrix generation procedure. The total inertia statistics for the model are extracted from the projected 6×6 rigid body mass matrix. You can specify the center of rotation for creating the artificial rotational rigid body modes and calculating the global inertia tensor. If the matrix quality check is requested, the check results are printed in the data (.dat) file.

References:

Abaqus Analysis User's Guide

- “Defining substructures,” Section 10.1.2
- “Generating matrices,” Section 10.3.1

Abaqus Keywords Reference Guide

- *MATRIX CHECK
- *MATRIX GENERATE
- *SUBSTRUCTURE GENERATE



4.6 Flexible body generation

Product: Abaqus/Standard

Benefits: You can now generate a flexible body from a substructure. In flexible body dynamic simulations, this capability can eliminate the need to store the full substructure recovery matrix, which is often very large, improving performance and reducing the size of the substructure **.sim** file.

Description: Abaqus/Standard can generate a flexible body from a substructure. The generated flexible body can be used in flexible body dynamic simulations using external solvers. Abaqus/Standard supports generation of several flexible body types that can be used by external flexible body dynamics solvers. The generated flexible body entities are stored in the substructure **.sim** file and can be converted to conventional flexible body representations by postprocessing programs.

References:

Abaqus Analysis User's Guide

- “Defining substructures,” Section 10.1.2

Abaqus Keywords Reference Guide

- *FLEXIBLE BODY
- *SUBSTRUCTURE GENERATE



5. Analysis techniques

This chapter discusses features related to analysis techniques in Abaqus. It provides an overview of the following enhancements:

- “Defining a local coordinate system for materials in Eulerian elements,” Section 5.1
- “Enhancements to the XFEM-based crack propagation capability,” Section 5.2
- “SIMULIA Co-Simulation Engine API availability,” Section 5.3
- “Enhancement to adaptive mesh refinement,” Section 5.4
- “Enhancements to the XFEM-based crack propagation capability in Abaqus/CAE,” Section 5.5
- “Parallel enhancement for DEM analysis,” Section 5.6
- “Enhancements to CZone for Abaqus for composite crush simulation,” Section 5.7
- “Vehicle ride comfort and durability co-simulation,” Section 5.8
- “Logical-Physical co-simulation using Abaqus and Dymola,” Section 5.9
- “Enhancements to the SIMULIA Co-Simulation Engine,” Section 5.10
- “Enhancements for co-simulation,” Section 5.11
- “MpCCI connector to the SIMULIA Co-Simulation Engine,” Section 5.12

5.1 Defining a local coordinate system for materials in Eulerian elements

Product: Abaqus/Explicit

Benefits: You can now define a local coordinate system for materials in Eulerian elements. Therefore, orientation-based materials can be used with Eulerian elements.

Description: In the Eulerian section definition you can now define an orientation for one or more materials in the section. Abaqus/Explicit remaps the material orientation as the material flows through element faces. Anisotropic linear elastic materials, orthotropic linear elastic materials, anisotropic hyperelastic materials, and materials with Hill Plasticity can now be used with Eulerian elements.

References:

Abaqus Analysis User's Guide

- “Eulerian analysis,” Section 14.1.1

Abaqus Keywords Reference Guide

- *EULERIAN SECTION
- *ORIENTATION



5.2 Enhancements to the XFEM-based crack propagation capability

Product: Abaqus/Standard

Benefits: The extended finite element method (XFEM) allows you to model discontinuities, such as cracks, along an arbitrary, solution-dependent path during an analysis. This method can now be extended to include the pore pressure degrees of freedom to account for the discontinuities of pore pressure as well as the fluid flow within the cracked element surfaces. This capability is very useful to accurately assess the hydraulic fracture process in the oil-gas industry as well as fluid flow in the life science and consumer goods industries. To improve the accuracy, you can now use nonlocal averaging stress and strain fields to determine if the damage initiation criterion is satisfied and to determine the crack propagation direction. In addition, output of some quantities on the cracked element surfaces is supported.

Description: XFEM allows you to model crack growth without remeshing the crack surfaces since it does not require the mesh to match the geometry of the crack. The XFEM-based cohesive segments method is extended to model hydraulically driven fracture. In this case additional phantom nodes with pore pressure degrees of freedom are introduced to model the fluid flow within the cracked element surfaces and to represent the discontinuities of displacement and fluid pressure in a cracked element. The fluid flow continuity, which accounts for both tangential and normal flow within and across the cracked element surfaces as well as the rate of opening of the cracked element surfaces, is maintained. The fluid pressure on the cracked element surfaces contributes to the traction-separation behavior of the cohesive segments in the enriched elements, which enables the modeling of hydraulically driven fracture.

In previous releases you could use the local stress and strain fields of an element ahead of the crack tip to determine the crack propagation and if the fracture criterion was satisfied. In the case of coarse and/or unstructured meshes a nonlocal averaging of the stress and strain fields ahead of the crack tip can lead to a more accurate evaluation of those fields, which can improve the accuracy of the computed propagation directions.

You can now output and visualize some surface variables on the cracked element surfaces.

References:

Abaqus Analysis User's Guide

- “Modeling discontinuities as an enriched feature using the extended finite element method,” Section 10.7.1
- “Boundary conditions in Abaqus/Standard and Abaqus/Explicit,” Section 34.3.1
- “Pore fluid flow,” Section 34.4.7

Abaqus Keywords Reference Guide

- *BOUNDARY
- *CFLOW
- *CONTACT OUTPUT



- *DAMAGE INITIATION
- *ENRICHMENT

Abaqus Benchmarks Guide

- “Propagation of hydraulically driven fracture using XFEM,” Section 1.19.5

5.3 SIMULIA Co-Simulation Engine API availability

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CFD

Benefits: The SIMULIA Co-Simulation Engine API, which allows you to couple in-house codes with Abaqus solvers, is now generally available with the Abaqus media.

Description: The SIMULIA Co-Simulation Engine (CSE) API allows you to couple in-house codes with Abaqus solvers. Co-simulation and embedding the CSE API in client code are intended for advanced users who are familiar with Abaqus, their in-house software, the co-simulation technique, and programming. You are encouraged to discuss your co-simulation requirements with your SIMULIA local support office.

The CSE API kit is distributed with the Abaqus media. When you install Abaqus, the CSE API kit is installed in the following location:

install_dir/api/cse/6.14-x_platform_CSE-api.zip

For example, if you install Abaqus 6.14 in **C:\SIMULIA\Abaqus\6.14-1**, the CSE API kit for the Windows 64-bit platform is available in the following location:

C:\SIMULIA\Abaqus\6.14-1\api\cse\6.14-1_win86_64_CSE-api.zip

The **.zip** file contains the CSE API headers, libraries, documentation, and example problems.

References:

Abaqus Analysis User's Guide

- Chapter 17, “Co-simulation”

Abaqus Installation and Licensing Guide

- “The Abaqus parent directory,” Section B.2

5.4 Enhancement to adaptive mesh refinement

Product: Abaqus/Explicit

Benefits: You can now apply the adaptive mesh refinement feature to a subset of elements in an Eulerian section.



Description: This feature benefits models in which the refinement region remains a small part of an Eulerian domain and does not vary much during the analysis. When the job is run with multiple domains, the elements in this subset can be decomposed independently and assigned to different processors. Load balance between processors can be greatly improved with this approach. For models in which the refinement region varies during the analysis, load imbalance can be reduced through the dynamic load balancing scheme.

References:

Abaqus Analysis User's Guide

- “Parallel execution in Abaqus/Explicit,” Section 3.5.3
- “Defining adaptive mesh refinement in the Eulerian domain,” Section 14.1.4

Abaqus Keywords Reference Guide

- *ADAPTIVE MESH REFINEMENT
- *DOMAIN DECOMPOSITION

5.5 Enhancements to the XFEM-based crack propagation capability in Abaqus/CAE

Product: Abaqus/CAE

Benefits: In Abaqus/CAE you can specify if the stress/strain values at the element centroid, at the crack tip, or the combination of both locations are used to measure the crack propagation criterion, which increases the coverage of Abaqus product functionality.

Description: Abaqus/CAE now supports the options for local calculations of the stress and strain fields ahead of the crack tip. You can specify if the stress/strain values at the element centroid, at the crack tip, or the combination of both locations are used to determine if the damage initiation criterion is satisfied and to determine the crack propagation direction (if needed).

Abaqus/CAE Usage:

Property module:

material editor: **Mechanical**→**Damage for Traction Separation Laws:**

Quade Damage, Maxe Damage, Quads Damage, Maxs Damage, Maxpe Damage,
or **Maxps Damage: Position**→**Centroid, Crack tip, or Combined**

Reference:

Abaqus/CAE User's Guide

- “Defining damage,” Section 12.9.3, in the HTML version of this guide



5.6 Parallel enhancement for DEM analysis

Product: Abaqus/Explicit

Benefits: Discrete element method (DEM) simulations run more efficiently due to domain decomposition of the DEM computations.

Description: Computations associated with DEM particle contact are now implemented in domain parallel, enabling better parallel scaling if multiple CPUs are used. Similar to parallel smoothed particle hydrodynamic (SPH) analysis, an evolving domain decomposition is used to avoid large spatial overlap among DEM domains (and, therefore, to maintain good parallel scaling) after large relative motions of DEM particles.

Reference:

Abaqus Analysis User's Guide

- “Discrete element method,” Section 15.1.1

5.7 Enhancements to CZone for Abaqus for composite crush simulation

Product: Abaqus/Explicit

Benefits: The CZone for Abaqus add-on capability has been enhanced to provide a more integrated solution for modeling composite crush in Abaqus/Explicit.

Description: CZone for Abaqus is an add-on product that combines the CZone methodology for composite crushing with the analysis capabilities of Abaqus/Explicit. This combination enables the inclusion of crushing behavior in FEA simulations of composite structures subjected to impact. The localized loading behavior in CZone for Abaqus is modeled using a surface-to-surface contact interaction.

The CZone for Abaqus functionality has been enhanced to provide a more integrated solution for composites crush simulations in Abaqus/Explicit. The enhancements include extended coverage of Abaqus functionality as well as significant improvements to the user interface and the execution process:

- Composite shell sections with orientations defined on individual layers or using distributions are now supported.
- Field variables are no longer needed to define and report the crushing contact state.
- New interface for the definition of the crush stress of the material.
- New interface for the definition of CZone crushing contact interactions.
- New output variable for CZone crushing state.
- Simplified execution process, without the need to use a precompiled Python script (`fileReader.py`) to run a CZone for Abaqus analysis.



For additional information about CZone for Abaqus, see the following page at www.3ds.com/simulia:

www.3ds.com/products-services/simulia/portfolio/abaqus/abaqus-portfolio/abaqus-add-ons/czone-for-abaqus/.

5.8 Vehicle ride comfort and durability co-simulation

Products: Abaqus/Standard Abaqus/Explicit

Benefits: Co-simulation with Abaqus and FTire (from Cosin Scientific Software) is now available for vehicle ride comfort and durability simulation.

Description: The Abaqus co-simulation technique is used with FTire to solve complex comfort and durability simulation of a vehicle maneuvering on an uneven road. Maneuvers such as accelerating, braking, rolling, and skidding can be modeled. The vehicle, its suspension, and wheel rims are modeled in Abaqus. The tire, the road, and the movement of the tire on the road are modeled in FTire (Flexible Ring Tire Model). Abaqus computes displacements and rotations associated with the forces and torques arising from the vehicle moving on the road, which are computed by FTire. Most of the Abaqus features are available for use, including nonlinear materials, nonlinear geometric effects, contact, and connectors. The complex tire phenomena is explained on a strictly mechanical, tribological, and thermodynamical basis in FTire.

Co-simulation with Abaqus and FTire is supported and qualified by SIMULIA.

References:

Abaqus Analysis User's Guide

- “Co-simulation: overview,” Section 17.1.1
- “Preparing an Abaqus analysis for co-simulation,” Section 17.2.1

5.9 Logical-Physical co-simulation using Abaqus and Dymola

Products: Abaqus/Standard Abaqus/Explicit

Benefits: You can now use co-simulation to model the interaction between logical and physical components. You can model logical controls (electronics), electric motors, pneumatics, or hydraulic devices in Dymola and model the mechanical, thermal, and acoustic physics in Abaqus.

Description: You can now use co-simulation between the logical modeling software Dymola and Abaqus to include the modeling of control electronics or electric devices in conjunction with the mechanical, thermal, or acoustical modeling. At any given time, sensor information computed in Abaqus can be exported into Dymola, which processes this information to produce an actuation. The actuation is then imported into Abaqus to drive the model to the desired state. Examples include the following:

- Robotic arms (modeled in Abaqus) actuated by electronically controlled electric motors (modeled in Dymola) to achieve a desired path for the payload



- Automotive ABS systems where tires (modeled in Abaqus) are required to maintain a rolling motion (detected via sensors) while the braking torque (actuation—computed in Dymola) is maximized
- Earth moving machinery such as backhoes (modeled in Abaqus) that require a controlled hydraulic pressure (computed in Dymola) to move their components to the desired effect

References:**Abaqus Analysis User's Guide**

- “Co-simulation: overview,” Section 17.1.1
- “Preparing an Abaqus analysis for co-simulation,” Section 17.2.1
- “Structural-to-logical co-simulation,” Section 17.4.1

Abaqus Keywords Reference Guide

- *CO-SIMULATION REGION

Abaqus Example Problems Guide

- “Analysis of a speaker using Abaqus-Dymola co-simulation,” Section 9.1.3

5.10 Enhancements to the SIMULIA Co-Simulation Engine

Products: Abaqus/Standard Abaqus/Explicit

Benefits: The SIMULIA Co-Simulation Engine is enhanced to allow the definition of multiple co-simulation regions and to provide a configuration file upgrade capability.

Description: The SIMULIA Co-Simulation Engine now supports multiple regions and multiple fields for Abaqus/Standard and Abaqus/Explicit coupled with certain partner products. A client can specify one or more regions as its co-simulation interface. The co-simulation interface region can be a set of discrete points, a surface region, a volume region, or a mixture of these types. Corresponding regions between clients need to be of the same region type, co-located, and have the same region boundaries.

Multiple regions can be used for:

- Transferring different field types across different co-simulation interfaces
- Coupling between points, surface regions, and volume regions
- Assigning different search and mapping tolerances in areas of complex geometry where mapping difficulties are expected

The CSE director process can automatically upgrade the schema for the configuration file from previous releases.



References:

Abaqus Analysis User's Guide

- “SIMULIA Co-Simulation Engine director execution,” Section 3.2.3
- “Co-simulation: overview,” Section 17.1.1

5.11 Enhancements for co-simulation

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CFD

Benefits: Several enhancements, including support for volume coupling, exchanging thermal fields on both faces of shell elements, and configuration file generation for third-party clients, improve the usability of the co-simulation technique.

Description: The following enhancements for co-simulation are available:

- Abaqus/Explicit now supports co-simulation where the interface region is a volume. Volume coupling allows coupling physics where the interface occupies the same space, as in the case of electromagnetic-to-structural coupling.
- In Abaqus/Standard and Abaqus/Explicit thermal fields can now be exchanged on both the SPOS and SNEG faces of shell elements. In previous releases only the SPOS or SNEG face was supported for thermal coupling. If both sides are loaded, you must define two regions, one containing the SPOS faces and the other containing the SNEG faces.
- In the SIMULIA Co-Simulation Engine configuration file, you can now specify the treatment for nodes and elements for which no intersection is found during the spatial mapping. You can specify a default value or specify that the value of the nearest neighbor node or element is to be used.
- The input file preprocessor for Abaqus/Standard and Abaqus/Explicit now writes a model description file that contains the co-simulation definitions (e.g., region names, field and their causality, etc.) This information can be used by a third-party client to generate the co-simulation configuration file without having to parse the Abaqus input file.
- The Run Time Environment (SRE) of the SIMULIA Co-Simulation Engine now supports version control. If incompatible SRE libraries are detected in either the CSE director or any of the clients, the co-simulation is terminated with an appropriate message.

Reference:

Abaqus Analysis User's Guide

- “Preparing an Abaqus analysis for co-simulation,” Section 17.2.1



5.12 MpCCI connector to the SIMULIA Co-Simulation Engine

Products: Abaqus/Standard Abaqus/Explicit

Benefits: The SIMULIA Co-Simulation Engine now supports an interface with MpCCI, expanding the number of third-party solvers that can be coupled with Abaqus.

Description: Prior to the Abaqus 6.14 release, the MpCCI client software was embedded in Abaqus/Standard and Abaqus/Explicit. With the Abaqus 6.14 release, Fraunhofer SCAI has developed a connection between the MpCCI server and the SIMULIA Co-Simulation Engine. This connection extends the co-simulation capability to nonpartner, MpCCI-compliant solvers that can be coupled with Abaqus. This connector supports the latest capabilities available with the SIMULIA Co-Simulation Engine, including implicit iterative coupling.

The connector process is distributed by Fraunhofer SCAI. For more information on MpCCI, see www.mpcci.de. For more information about coupling with third-party software, see the Co-simulation page at www.3ds.com/simulia.

Third-party code coupling with Abaqus via MpCCI is qualified and supported by Fraunhofer SCAI.

Reference:

Abaqus Analysis User's Guide

- “Co-simulation: overview,” Section 17.1.1



6. Materials

This chapter discusses new material models or changes to existing material models. It provides an overview of the following enhancements:

- “Enhancements to parallel rheological framework,” Section 6.1
- “Eulerian elements with anisotropic materials,” Section 6.2
- “Enhancements to creep models,” Section 6.3
- “User-defined frequency domain viscoelastic behavior,” Section 6.4
- “New hybrid formulations for user-defined material behavior,” Section 6.5
- “Different stiffness in tension versus compression for traction-separation elasticity,” Section 6.6
- “Element deletion controlled by state variables,” Section 6.7

6.1 Enhancements to parallel rheological framework

Products: Abaqus/Standard Abaqus/Explicit

Benefits: You can now use the power law creep model or a user-defined creep model that includes pressure and total deformation dependency to define the viscous response in the viscoelastic network. In the equilibrium network, the response that includes plasticity with combined isotropic and kinematic hardening can now be defined. The implementation of nonlinear kinematic hardening will be most beneficial for applications that include cycling loading. In addition, transferring results between various Abaqus analyses is now supported for models defined using the parallel rheological framework.

Description: The viscous response in the viscoelastic network can be defined using a new, power law creep model that includes pressure dependency. In addition, a user-defined creep model can include dependency on pressure and total deformation through appropriate invariants. In Abaqus/Standard the equilibrium network response has been enhanced and can include plasticity with combined isotropic and nonlinear kinematic hardening. The implementation of the nonlinear kinematic hardening is based on the Armstrong-Frederick model and allows multiple backstresses to be specified. Finally, the import capability for material models defined using a parallel rheological framework is now fully supported.

References:

Abaqus Analysis User's Guide

- “Transferring results between Abaqus analyses: overview,” Section 9.2.1
- “Parallel rheological framework,” Section 22.8.2

Abaqus Keywords Reference Guide

- *VISCOELASTIC



MATERIALS

Abaqus User Subroutines Reference Guide

- “UCREEPNETWORK,” Section 1.1.23

Abaqus Verification Guide

- “Transferring results with parallel rheological framework,” Section 3.14.20

6.2 Eulerian elements with anisotropic materials

Product: Abaqus/Explicit

Benefits: You can now perform a coupled Eulerian Lagrangian analysis with anisotropic materials.

Description: For analyses involving extreme deformation, coupled Eulerian-Lagrangian analyses are more effective than traditional Lagrangian analyses because the elements do not deform and materials are allowed to flow through the element faces. This technology can now be used for models involving materials with anisotropic behavior.

References:

Abaqus Analysis User's Guide

- “Eulerian analysis,” Section 14.1.1

Abaqus Keywords Reference Guide

- *ANISOTROPIC HYPERELASTIC
- *EULERIAN SECTION
- *ORIENTATION

6.3 Enhancements to creep models

Product: Abaqus/Standard

Benefits: The classical deviatoric creep behavior has been enhanced by introducing three new creep models: Anand, Darveaux, and double power. These three creep laws are particularly well suited for modeling the behavior of solder alloys used in electronic packaging and have been shown to produce accurate results for a wide range of temperatures and strain rates.

Description: Three new creep models (Anand, Darveaux, and double power) have been implemented in Abaqus/Standard. These models can be used to define metal creep behavior as well as the viscous behavior for a two-layer viscoplastic material.

The double power model consists of two terms that introduce the effects of two mechanisms responsible for creep: climb (low stress) and combined glide/climb (high stress).



The Darveaux model takes into account both primary and secondary creep. The secondary creep is described by a standard hyperbolic sine law, which is modified to incorporate primary creep effects. The temperature dependence is introduced through an Arrhenius-type term.

The Anand model describes creep strain using a hyperbolic sine model, which is modified by introducing a single internal variable, the deformation resistance. This variable represents isotropic resistance to inelastic deformation, and its evolution is defined by a separate rate equation. The temperature dependence is described by an Arrhenius-type term and by making the initial deformation resistance a function of temperature and the hardening/softening parameter a function of temperature and creep strain rate.

References:

Abaqus Analysis User's Guide

- “Rate-dependent plasticity: creep and swelling,” Section 23.2.4
- “Two-layer viscoplasticity,” Section 23.2.11

Abaqus Keywords Reference Guide

- *CREEP
- *VISCOUS

6.4 User-defined frequency domain viscoelastic behavior

Product: Abaqus/Standard

Benefits: You can now define frequency domain viscoelastic behavior through user subroutine **UMAT**.

Description: Viscoelastic behavior in the frequency domain requires specification of a storage modulus (material stiffness) and a loss modulus (material damping). The **UMAT** interface and implementation has been enhanced to support the specification of both material stiffness and material damping. This feature allows you to define viscoelastic response with a complex frequency dependence, which is not supported by the built-in models in Abaqus/Standard.

References:

Abaqus Analysis User's Guide

- “User-defined mechanical material behavior,” Section 26.7.1

Abaqus Keywords Reference Guide

- *USER MATERIAL

Abaqus User Subroutines Reference Guide

- “UMAT,” Section 1.1.41



MATERIALS

Abaqus Verification Guide

- “**UMAT** and **UHYPER**,” Section 4.1.21

6.5 New hybrid formulations for user-defined material behavior

Product: Abaqus/Standard

Benefits: You can now define almost incompressible and fully incompressible nonlinear elastic material behavior with hybrid elements through user subroutine **UMAT**.

Description: The default formulation used by Abaqus when a user material is used to define the response of a hybrid element is suitable for material models that use an incremental formulation (for example, metal plasticity) but is not consistent with the total formulation that is commonly used for hyperelastic materials. In the latter situation the default formulation may lead to convergence problems. Such convergence problems may be observed, for example, when an almost incompressible nonlinear elastic user material is subjected to large deformations. Abaqus/Standard now provides an alternate total formulation for such situations. This formulation is consistent with the native almost incompressible formulation used by Abaqus for hyperelastic materials and works better than the default formulation in these situations. Abaqus/Standard also provides, for the first time, the capability to define a fully incompressible response through user subroutine **UMAT**. The total hybrid formulation can be implemented with minimal changes to an existing **UMAT**. For fully incompressible materials user subroutine **UMAT** needs to define the deviatoric part of the material response only.

References:

Abaqus Analysis User's Guide

- “User-defined mechanical material behavior,” Section 26.7.1

Abaqus Keywords Reference Guide

- *USER MATERIAL

Abaqus User Subroutines Reference Guide

- “**UMAT**,” Section 1.1.41

Abaqus Verification Guide

- “**UMAT** and **UHYPER**,” Section 4.1.21

6.6 Different stiffness in tension versus compression for traction-separation elasticity

Product: Abaqus/Standard



Benefits: You can now define uncoupled traction-separation elastic behavior with different values of stiffness in tension and compression.

Description: In some applications that utilize cohesive elements to model adhesive behavior in both tension and compression, it is useful to be able to specify different values of stiffness in tension versus compression traction-separation responses. This capability affects only the normal traction component.

References:

Abaqus Analysis User's Guide

- “Defining the constitutive response of cohesive elements using a traction-separation description,” Section 32.5.6

Abaqus Keywords Reference Guide

- *ELASTIC

Abaqus Verification Guide

- “Elastic materials,” Section 2.2.1

6.7 Element deletion controlled by state variables

Products: Abaqus/Standard

Benefits: In Abaqus/Standard deletion of solid, shell, membrane, and beam elements can now be controlled using state variables. Deleted elements have no ability to carry stresses and do not contribute to the stiffness of the model. They also do not undergo element quality checks, so an analysis cannot be terminated due to excessive distortion of such elements.

Description: Deletion of elements can now be controlled by state variables in Abaqus/Standard, similar to the functionality previously available in Abaqus/Explicit. The number of the state variable that controls deletion must be specified, and it can be different for each material. The state variable stores the element deletion flag, which can have a value of zero or one. A value of one indicates that the material point is active, while a value of zero indicates that the material point is inactive and the stresses at this point are set to zero. Once a material point is deleted, it cannot be reactivated. The deletion flag can be modified by any of the Abaqus/Standard user subroutines that use state variables and that are called at the material point. If all the material points of an element become inactive, the element is deleted. The status of an element can be monitored by requesting the output variable STATUS.



MATERIALS

References:

Abaqus Analysis User's Guide

- “User subroutines: overview,” Section 18.1.1
- “User-defined mechanical material behavior,” Section 26.7.1

Abaqus Keywords Reference Guide

- *DEPVAR



7. Elements

This chapter discusses elements available in Abaqus. It provides an overview of the following enhancements:

- “Support for coupled temperature–pore pressure elements in Abaqus/CAE,” Section 7.1
- “Linear pyramid acoustic element,” Section 7.2

7.1 Support for coupled temperature–pore pressure elements in Abaqus/CAE

Product: Abaqus/CAE

Benefits: You can now assign element types from the coupled temperature–pore pressure family of elements to your model in Abaqus/CAE.

Description: Coupled temperature–pore pressure elements are used in Abaqus/Standard for modeling fully or partially saturated fluid flow through a deforming porous medium in which the stress, fluid pore pressure, and temperature fields are fully coupled to one another. The **Element Type** dialog box in the Mesh module now enables you to assign element types from the coupled temperature–pore pressure family of elements, which includes element types C3D8PT, C3D8PHT, C3D8RPT, C3D8RPHT, and C3D10MPT.

Abaqus/CAE Usage:

Mesh module

Mesh→Element Type: Family: Coupled Temperature-Pore pressure

References:

Abaqus Analysis User's Guide

- “Three-dimensional solid element library,” Section 28.1.4

Abaqus/CAE User's Guide

- “Element type assignment,” Section 17.5.3

7.2 Linear pyramid acoustic element

Product: Abaqus/Standard

Benefits: The pyramid element is useful to transition between brick elements and tetrahedra elements during mesh generation.

Description: Acoustic elements are provided in Abaqus/Standard for modeling acoustic domain in a purely acoustic analysis or in a coupled acoustic-structural analysis. The new 5-node pyramid element AC3D5



ELEMENTS

supports all of the loadings, material behaviors, and boundary conditions supported by the existing acoustic elements.

References:

Abaqus Analysis User's Guide

- “Three-dimensional solid element library,” Section 28.1.4

Abaqus Verification Guide

- “Eigenvalue extraction for single unconstrained elements,” Section 1.2.1
- “Acoustic modes,” Section 1.2.3
- “Patch test for acoustic elements,” Section 1.5.10
- “Analysis of unbounded acoustic regions,” Section 1.11.8
- “Acoustic submodeling,” Section 3.8.14
- “Volumetric drag,” Section 3.9.1
- “Impedance boundary conditions,” Section 3.9.2
- “Transient acoustic wave propagation,” Section 3.9.3
- “Acoustic model change: steady state,” Section 3.10.8
- “Surface-based tie constraint,” Section 5.1.27



8. Prescribed conditions

This chapter discusses loads, boundary conditions, and predefined fields. It provides an overview of the following enhancements:

- “Initial conditions on damage initiation,” Section 8.1
- “Enhancements for prescribed conditions in Abaqus/CFD,” Section 8.2
- “Acoustic base motion for SIM-based procedures,” Section 8.3

8.1 Initial conditions on damage initiation

Products: Abaqus/Standard Abaqus/Explicit

Benefits: You can now define initial conditions directly on the damage initiation measures for the ductile, shear, and the M₀ forming limit diagram based damage initiation criteria.

Description: This capability is particularly useful in situations where a metal forming operation is carried out in one analysis, which is followed by a separate analysis that subjects the formed metal part to further deformation. The damage initiation measures at the end of the first analysis can be directly specified as initial conditions for the second analysis. An alternate but approximate way of modeling initial conditions on damage initiation is by specifying initial values of the equivalent plastic strain. Abaqus computes damage initiation measures based on the specified initial equivalent plastic strain, assuming a linear strain path between the initial (undeformed) state and the final (deformed) state. This approximation does not work well for deformation paths that deviate significantly from linearity in the strain space.

References:

Abaqus Analysis User's Guide

- “Initial conditions in Abaqus/Standard and Abaqus/Explicit,” Section 34.2.1

Abaqus Keywords Reference Guide

- *INITIAL CONDITIONS

Abaqus Verification Guide

- “Progressive damage and failure of ductile metals,” Section 2.2.21

8.2 Enhancements for prescribed conditions in Abaqus/CFD

Product: Abaqus/CFD



PREScribed CONDITIONS

Benefits: The process to specify initial conditions and boundary conditions for the turbulence model variables is enhanced to be more intuitive and flexible.

Description: You can now specify more intuitive turbulence quantities, such as the turbulence intensity or the turbulent viscosity ratio, and Abaqus/CFD calculates the actual values of the turbulence model variables. Previously, you were required to precompute values for the turbulence variables. The new turbulence quantities can be used to specify initial conditions and boundary conditions for all of the turbulence models. For example, you can prescribe the turbulence intensity (TI) and a characteristic velocity scale (U_∞) to specify the turbulent kinetic energy for the k - ϵ RNG, k - ϵ realizable, and k - ω SST turbulence models as

$$k = \frac{3}{2}(TI \cdot U_\infty)^2.$$

References:

Abaqus Analysis User's Guide

- “Initial conditions in Abaqus/CFD,” Section 34.2.2
- “Boundary conditions in Abaqus/CFD,” Section 34.3.2

Abaqus Keywords Reference Guide

- *FLUID BOUNDARY
- *INITIAL CONDITIONS

8.3 Acoustic base motion for SIM-based procedures

Product: Abaqus/Standard

Benefits: You can now use acoustic media with prescribed pressure values, with different values for different constraint points in the mode-based analysis (similar to the functionality available in a direct-solution steady-state dynamic analysis). This feature is useful in the automotive industry; for example, to analyze mufflers and fuel tanks.

Description: Acoustic pressure base motion is implemented in modal SIM-based transient dynamic and steady-state dynamic (including subspace projection) analyses as “secondary” bases.

To enable nonzero boundary conditions in the modal analysis, Abaqus uses the “big mass” method. Mathematically, this method is a variant of the penalty method. Abaqus uses secondary base motion to support the nonzero boundary conditions in modal procedures. Named secondary bases are defined by specifying boundary conditions in the frequency extraction step. In consecutive modal analysis steps, these boundary conditions can be associated with amplitude curves and applied as base motions. You can now apply the secondary base motion to the acoustic pressure degree of freedom 8; in previous releases, you could apply the secondary base motion only to the mechanical degrees of freedom (1–6). Because acoustic excitation is a scalar value (as opposed to mechanical degrees of freedom), the orientation of the node is irrelevant.



Acoustic pressure base motions are implemented for the following models:

- acoustic media only
- combined acoustic and structural media; uncoupled modes
- combined acoustic and structural media; coupled modes (modal steady-state dynamic, including subspace projection, analyses only)

References:

Abaqus Analysis User's Guide

- “Dynamic analysis procedures: overview,” Section 6.3.1
- “Prescribed motions in modal superposition procedures” in “Transient modal dynamic analysis,” Section 6.3.7

Abaqus Keywords Reference Guide

- *BASE MOTION



9. Constraints

This chapter discusses kinematic constraints. It provides an overview of the following enhancements:

- “New attachment method for mesh-independent fasteners,” Section 9.1
- “Enhanced domain decomposition of thermal ties,” Section 9.2

9.1 New attachment method for mesh-independent fasteners

Products: Abaqus/Standard Abaqus/Explicit

Benefits: You can now preserve the position and direction of the connection for mesh-independent fasteners.

Description: Previously, all the attachment methods for mesh-independent fasteners allowed the fastening point to move to lie on the normal to the first attachment surface. The new attachment method, which must use connector elements for attachment, preserves the position and direction of the connection.

References:

Abaqus Analysis User's Guide

- “Mesh-independent fasteners,” Section 35.3.4

Abaqus Keywords Reference Guide

- *FASTENER

9.2 Enhanced domain decomposition of thermal ties

Product: Abaqus/Explicit

Benefits: Domain decomposition for thermal ties has been improved. The improvement results in a more equitable domain decomposition of thermal ties as well as the elements whose faces are involved in the tie constraints. The performance is especially improved for models that have large thermal ties, and better scaling is observed for execution with a large number of processors.

Description: There are no changes to the keyword interface. Improved domain decomposition is the default behavior.

References:

Abaqus Analysis User's Guide

- “Fully coupled thermal-stress analysis in Abaqus/Explicit” in “Fully coupled thermal-stress analysis,” Section 6.5.3



10. Interactions

This chapter discusses features related to contact and interaction modeling. It provides an overview of the following enhancements:

- “Edge-to-edge contact for general contact in Abaqus/Standard,” Section 10.1
- “Improved contact at complex intersections in Abaqus/Explicit,” Section 10.2
- “Enhancement to the contact detection tool in Abaqus/CAE,” Section 10.3

10.1 Edge-to-edge contact for general contact in Abaqus/Standard

Product: Abaqus/Standard

Benefits: Edge-to-edge contact is extended to include feature edges on solid and shell-like surfaces and shell perimeter edges for more automated and robust resolution of contact between edges. New contact output variables allow visualizing contact enforcement with edge-to-edge contact.

Description: Feature edges on solid and shell-like surfaces and shell perimeter edges can now participate in edge-to-edge contact within the general contact framework. These changes vastly expand the edge-to-edge contact capability compared to the previous release, where only contact between edges on beam surfaces (beam-to-beam contact) was supported. Various robustness and accuracy improvements have also been made. While edge-to-edge contact is the primary formulation for contact between beams, in other cases (for example, between feature edges), it supplements the edge-to-surface and the surface-to-surface contact when those formulations are active.

New surface nodal output variables, CLINELOAD and CPOINTLOAD, are provided to visualize contact loads in active contact regions involving edges. Output variable CLINELOAD has units of force per length and provides a mesh-independent contact load measure where the active contact region is a curve rather than an area. Beam-to-beam contact using the radial formulation and edge-to-surface contact contribute to this output variable. Output variable CPOINTLOAD has units of force and helps visualize contact between non-parallel edges where the active contact region can be idealized as a point. The contact stresses, CSTRESS, reflect contributions from surface-to-surface, edge-to-surface, and edge-to-edge contact with units of force per area.

Figure 10–1 shows initial and deformed configurations of a chain falling under gravity. The individual chain links are meshed with beam, shell, or solid elements such that beam, shell perimeter, and feature edges are activated to participate in edge-to-edge contact. Edge-to-edge contact supplements the edge-to-surface and surface-to-surface contact, which are also active in this model between the individual links and between the chain and the rigid platform. Figure 10–2 shows contact load CPOINTLOAD between the edges of the individual links due to edge-to-edge contact.



INTERACTIONS

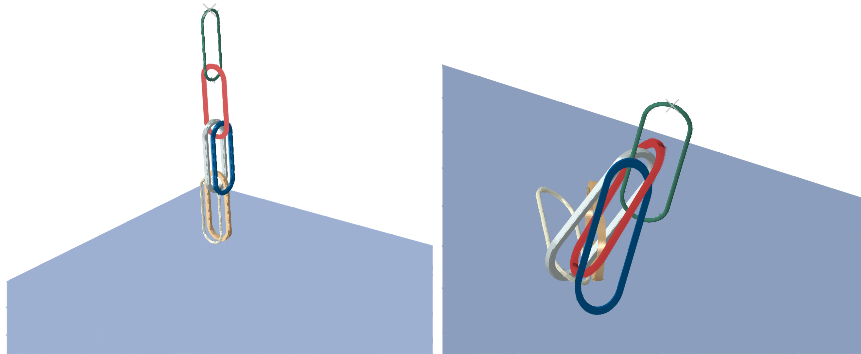


Figure 10–1 Chain falling under gravity onto a rigid platform: initial (left) and deformed (right) configurations.

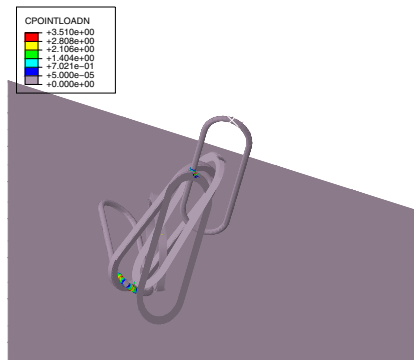


Figure 10–2 Contour plot of contact load CPOINTLOAD.

References:

Abaqus Analysis User's Guide

- “Abaqus/Standard output variable identifiers,” Section 4.2.1
- “Contact interaction analysis: overview,” Section 36.1.1
- “Defining general contact interactions in Abaqus/Standard,” Section 36.2.1
- “Surface properties for general contact in Abaqus/Standard,” Section 36.2.2

Abaqus Keywords Reference Guide

- *CONTACT FORMULATION



- *CONTACT OUTPUT
- *SURFACE PROPERTY ASSIGNMENT

10.2 Improved contact at complex intersections in Abaqus/Explicit

Product: Abaqus/Explicit

Benefits: Contact at complex intersections (edges connected to more than two faces) is improved.

Description: Surface offsets at complex intersections were always set to zero in previous releases of Abaqus/Explicit, even if you specified a nonzero reference surface offset for the underlying elements. This limitation sometimes caused undesirable solution behavior. It is removed in the current release.

Figure 10–3 shows a scenario in which a block slides along shell elements whose reference surface corresponds to the bottom surface of a shell. The bottom surface of the shell happens to correspond to the shell reference surface because a nonzero offset has been specified. The fact that the larger body has a T-junction should not be particularly relevant to this model, but the previous limitation (surface offset set to zero) caused the T-junction to have a large effect.

Shell element thickness profiles are represented in Figure 10–3, but these profiles do not reflect the effect of the previous limitation on the contact surface geometry. The sequence of four deformed-configuration plots at the top of Figure 10–3 shows the effect of the previous limitation as the block nears the T-junction: a gap appears between the bodies (even though significant contact forces are still being transmitted). Solution noise remains in this example even after the block is well past the T-junction. The sequence of plots at the bottom of Figure 10–3 shows the improved (physically realistic) behavior. In this case the block simply slides along the other body as expected, with very little deformation of either part.

This enhancement is in effect by default.

Reference:

Abaqus Analysis User's Guide

- “Surface offset” in “Assigning surface properties for general contact in Abaqus/Explicit,” Section 36.4.2

10.3 Enhancement to the contact detection tool in Abaqus/CAE

Product: Abaqus/CAE

Benefits: You can now include model instances when you search for contact pairs involving different regions on the same instance. This enhancement improves the usability of the contact detection tool in Abaqus/CAE.

Description: The contact detection algorithm has been extended to include model instances when you search for contact pairs involving different regions on the same instance.



INTERACTIONS

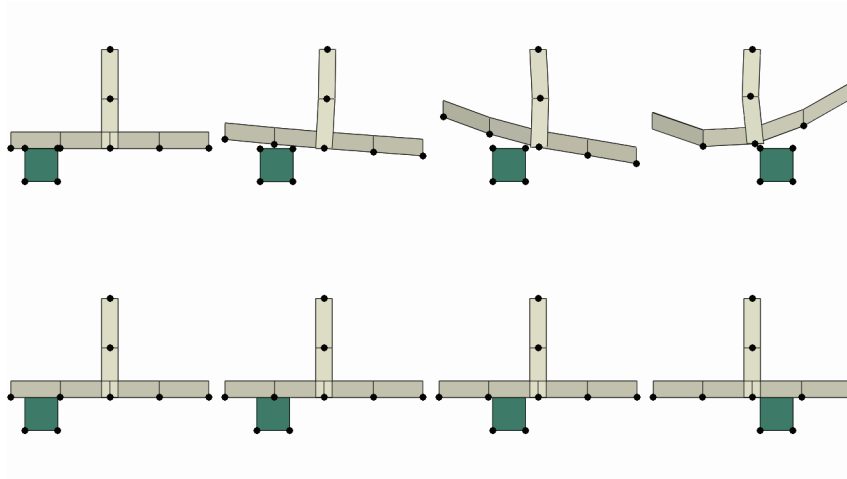


Figure 10–3 Block sliding on a flat portion of a T-junction structure with the previous behavior (top) and current behavior (bottom).

Abaqus/CAE Usage:

Interaction module:

Interaction→**Find contact pairs**: select **Whole model** or select **Instance** and child part instances from the viewport, toggle on **Include pairs with surfaces on the same instance**

References:

Abaqus/CAE User's Guide

- “The contact detection algorithm,” Section 15.6.2
- “Specifying search criteria for contact detection,” Section 15.16.1, in the HTML version of this guide



11. Execution

This chapter discusses commands and utilities for running the Abaqus products. It provides an overview of the following enhancements:

- “GPGPU acceleration of the AMS eigensolver,” Section 11.1
- “GPGPU accelerated direct equation solver,” Section 11.2
- “User control of domain decomposition,” Section 11.3
- “Job execution control enhancements,” Section 11.4
- “SIM database utilities,” Section 11.5
- “Enhancements for translating Abaqus data to modal neutral files,” Section 11.6

11.1 GPGPU acceleration of the AMS eigensolver

Products: Abaqus/Standard Abaqus/AMS

Benefits: The AMS eigensolver can use compute-capable GPGPU cards to reduce the run time for frequency extraction analyses.

Description: GPGPU acceleration of the reduced eigensolution phase of the AMS eigensolver delivers improved performance for large-scale models requiring more than 10,000 eigenmodes. In addition, the parallel scaling of the reduced eigensolution phase is improved compared to Abaqus 6.13 for the same kind of models. Table 11–1 illustrates the performance improvements of the AMS eigensolver due to GPGPU acceleration for two industrial models: Model 1 is a 2.5 million degree-of-freedom vehicle body model, and Model 2 is a 9.4 million degree-of-freedom vehicle body model. Both analyses were run on a Linux machine with dual 2.6 GHz Intel Sandybridge processors (2×8 cores), 128 GB memory, and 2 Nvidia Kepler cards (K20X). In all cases 16 cores are used.

References:

Abaqus Analysis User's Guide

- “Natural frequency extraction,” Section 6.3.5

Abaqus Keywords Reference Guide

- *FREQUENCY

**Table 11–1** Performance improvements due to GPGPU acceleration of the AMS eigensolver.

Model	Degrees of Freedom	Number of Modes	Abaqus 6.13	Abaqus 6.14	Abaqus 6.14 with 2 GPUs	Abaqus 6.14 GPU Acceleration Speedup Factor
Model 1	2.5 million	16,926	3816 s	3433 s	2662 s	1.29
Model 2	9.4 million	10,743	1293 s	1159 s	837 s	1.38

11.2 GPGPU accelerated direct equation solver

Product: Abaqus/Standard

Benefits: Performance of the GPGPU accelerated direct equation solver has been improved, and GPGPU memory requirements have been reduced.

Description: For large models where the solution time is dominated by the direct equation solver, use of GPGPU can reduce the overall time of the analysis significantly, as shown in Figure 11–1. These types of models use predominantly solid elements and have more than 1 million equations.

Reference:

Abaqus Analysis User's Guide

- “Parallel execution in Abaqus/Standard,” Section 3.5.2

11.3 User control of domain decomposition

Product: Abaqus/Explicit

Benefits: Improved parallel scalability can be achieved when localized regions are computationally intensive.

Description: You can influence the domain decomposition by specifying a region to be independently decomposed or by specifying that all elements of a particular element set should be constrained to the same parallel domain. This option is not recommended for widespread use. It can be beneficial to parallel scalability for some cases in which a localized region requires more extensive contact or other computations than other regions, but this option may degrade performance. For example, specifying the impact region in a bird strike model as an independent domain decomposition region may improve parallel scalability. Coupled Eulerian-Lagrangian simulations with localized adaptive mesh refinement (where elements are refined adding to the computational cost) may also benefit from defining a local domain decomposition region. With this

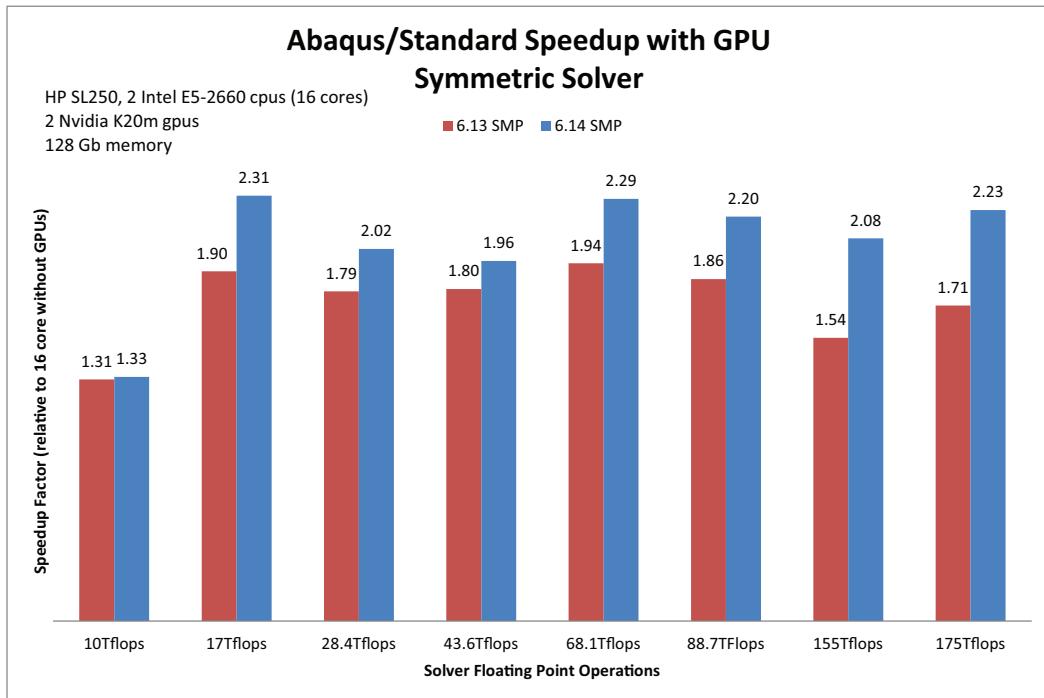


Figure 11–1 GPGPU accelerated direct solver performance improvement.

approach, each processor will process multiple domains: for example, one domain associated with a localized region and one domain associated with the rest of the model.

Multiple domain decomposition regions can be specified. The number of domains in total is the total number of domain decomposition regions times the user-specified number of domains per region. The number of domain decomposition regions is often one more than the number of user-defined domain decomposition regions because elements not explicitly defined to be in a domain decomposition region will form a separate domain decomposition region.

References:

Abaqus Analysis User's Guide

- “Parallel execution in Abaqus/Explicit,” Section 3.5.3

Abaqus Keywords Reference Guide

- *DOMAIN DECOMPOSITION



EXECUTION

11.4 Job execution control enhancements

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CFD

Benefits: You can now control job execution remotely; previously, you were required to execute job control commands on the machine running the analysis. In addition, two new command line options are available for network connections.

Description: The execution procedures for job execution control include **abaqus suspend**, **abaqus resume**, and **abaqus terminate**. These utilities are used to suspend, resume, and terminate Abaqus analysis jobs. You can now execute these commands remotely; you are not required to execute these commands on the machine running the analysis.

The **host** and **port** options are used to specify the host name and port number, respectively, for the connection running the analysis job. You can specify either the **job** option or the **host** and **port** options to suspend, resume, or terminate Abaqus analysis jobs.

Reference:

Abaqus Analysis User's Guide

- “Job execution control,” Section 3.2.39

11.5 SIM database utilities

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CFD

Benefits: The new SIM database utilities provide tools that allow you to convert SIM database files to make them compatible with other releases.

Description: You can use the **abaqus sim_version** utilities to convert a SIM database file from one release to another, to query a SIM database file for its SIM release level, or to determine the SIM release level of the current code you are using. For more information on the SIM database file, see “Writing results from an Abaqus analysis to a SIM database,” Section 12.1.

Reference:

Abaqus Analysis User's Guide

- “SIM database utilities,” Section 3.2.19

11.6 Enhancements for translating Abaqus data to modal neutral files

Product: Abaqus/Standard



Benefits: The translation of Abaqus results to a modal neutral file is significantly faster and the size of the modal neutral file can be significantly smaller than in previous releases.

Description: Two enhancements to the **abaqus adams** translator are available. The translator now reads inertia invariants computed during substructure generation rather than computing them, which makes translating the results in an Abaqus substructure SIM database file to an MSC.ADAMS modal neutral (**.mnf**) file significantly faster. Now, the translator only needs to process an “interesting” subset of the displacement modes, which can result in a significantly smaller modal neutral file.

Reference:

Abaqus Analysis User's Guide

- “Translating Abaqus data to MSC.ADAMS modal neutral files,” Section 3.2.36



12. Output and visualization

This chapter discusses obtaining, postprocessing, and visualizing results from Abaqus analyses. It provides an overview of the following enhancements:

- “Writing results from an Abaqus analysis to a SIM database,” Section 12.1
- “Output of mode mix ratios during mixed mode delamination,” Section 12.2
- “Output of last converged increment,” Section 12.3
- “Output of contact-related energies,” Section 12.4
- “Magnitude output of vector nodal variables,” Section 12.5
- “Equivalent plastic strain rate,” Section 12.6
- “Principal stresses with bounding values for output field filtering,” Section 12.7
- “Requesting reference sink temperature and film coefficient field output in Abaqus/CAE,” Section 12.8
- “Generating tabular reports for free body cuts from multiple frames,” Section 12.9
- “Selecting the area centroid as the summation point for free body cuts,” Section 12.10
- “Reading X – Y data from free bodies defined on view cuts,” Section 12.11
- “Performing display group Boolean operations on linked viewports,” Section 12.12

12.1 Writing results from an Abaqus analysis to a SIM database

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CFD Abaqus/CAE

Benefits: You can now write results to a SIM database (**.sim**) file, which allows you to import the results of an Abaqus analysis into the Physics Results Explorer app on the 3DEXPERIENCE Platform for high-performance postprocessing.

Description: The new SIM database (*job-name.sim*) is the file format used to exchange bulky FEA data between 3DEXPERIENCE Platform apps and computationally intensive components like implicit and explicit solvers. For example, the Physics Results Explorer app uses the SIM database for high-performance postprocessing of analysis results.

The Abaqus output database is now available in two formats, ODB and SIM. By default, the results output is created in ODB format. Only results in SIM format can be imported into the 3DEXPERIENCE Platform for high-performance postprocessing. For Abaqus/Standard and Abaqus/Explicit analyses the new **resultsformat** command line option provides control over the output format for Abaqus results, including writing results in both formats. For Abaqus/CFD analyses you use the **field** and **history** command line options to write results in SIM format.

Your choice of output format depends on your level of experience with high-performance visualization, the Physics Results Explorer app, and your postprocessing needs.



OUTPUT AND VISUALIZATION

- If you are still learning to use high-performance visualization and you want to compare your results with Abaqus/Viewer, write results in both formats.
- If the model is large and you need the improved performance of the Physics Results Explorer app, as well as the capabilities of Abaqus/Viewer, write results in both formats.
- If you are confident that the high-performance visualization features in the Physics Results Explorer app provide all the capabilities you need, write results in SIM format.

A subset of options in Abaqus/Standard and Abaqus/Explicit are not supported for analyses that produce results in SIM format; Abaqus/CFD has no restrictions on output in SIM format. If you include one or more of these options or parameters in your analysis and write results in SIM format or both formats, the analysis will either terminate with errors or produce limited results. For more information, see “Limitations when writing results in SIM format” in “Output,” Section 4.1.1 of the Abaqus Analysis User’s Guide.

Abaqus/CAE Usage:

Job module:

Job→**Edit: General** tabbed page: **Results Format: ODB, SIM, or Both** (Abaqus/Standard and Abaqus/Explicit only)

References:

Abaqus Analysis User’s Guide

- “Abaqus/Standard, Abaqus/Explicit, and Abaqus/CFD execution,” Section 3.2.2
- “Output,” Section 4.1.1

Abaqus/CAE User’s Guide

- “The job editor,” Section 19.2.4

12.2 Output of mode mix ratios during mixed mode delamination

Product: Abaqus/Standard

Benefits: For delamination behavior modeled with cohesive elements, you can now output the mode mix ratios both at the initiation of damage and during damage evolution.

Description: Delamination is a common failure mode for laminated composite materials, and the energy required for delamination is generally a strong function of the mode mix ratio in the delamination zone. The mode mix ratio refers to the relative proportion of tensile to shear deformation. In most practical situations, the delamination process is strongly mixed mode in nature, and the mode mix ratio often varies significantly during the delamination process. The new output variables, MMIXDMI and MMIXDME, measure the mode mix ratios at integration points in cohesive elements at damage initiation and during damage evolution, respectively. These variables provides an accurate measure of the mode of failure (tensile versus shear) during mixed mode delamination.

**References:****Abaqus Analysis User's Guide**

- “Defining the constitutive response of cohesive elements using a traction-separation description,” Section 32.5.6

Abaqus Keywords Reference Guide

- *OUTPUT

Abaqus Verification Guide

- “Transferring results from one Abaqus/Standard analysis to another Abaqus/Standard analysis,” Section 3.14.2

12.3 Output of last converged increment

Product: Abaqus/Standard

Benefits: If a general step in an Abaqus/Standard analysis fails to converge, the results from the last converged increment are written to the output database. This can be helpful in diagnosing why convergence failed.

Description: The default output behavior has been modified to output the last converged increment when convergence is not achieved. The output is written only if there are convergence difficulties beyond the first increment. This default behavior is available for all general steps in Abaqus/Standard.

Reference:**Abaqus Analysis User's Guide**

- “Output to the output database,” Section 4.1.3

12.4 Output of contact-related energies

Product: Abaqus/Standard

Benefits: New contact-related whole model energies have been introduced, and energy accounting (ETOTAL) has been improved.

Description: New energy output variables are available to avoid drifting of the total energy balance ETOTAL and provide insight to simulation behavior. These new output variables account for elastic contact energy, dissipative energies associated with contact stabilization and contact damping, and energy associated with contact constraint “discontinuity work,” as described in Table 12–1.



Table 12–1 New output variables for contact-related energy.

Description		Output variable
Elastic contact energy	Energy stored among all penalty springs and “softened” contact constraints associated with normal contact constraints	ALLCCEN
	Energy stored among all penalty springs associated with tangential contact constraints	ALLCCET
	Energy stored among all penalty springs and “softened” contact constraints associated with normal and tangential contact constraints (equal to the sum of ALLCCEN and ALLCCET)	ALLCCE
Energy dissipation associated with contact stabilization and contact damping	Normal contact direction for the whole model	ALLCCSDN
	Tangential contact direction for the whole model	ALLCCSDT
	Whole model (equal to the sum of ALLCCSDN and ALLCCSDT)	ALLCCSD
Energy associated with contact constraint “discontinuity work”	Accounts for the portion of the work done by contact forces when contact conditions change that is not accounted for by other contact energy variables	ALLCCDW

The existing output variables ALLSD and ALLVD continue to account for dissipative energies associated with contact stabilization and contact damping as they have in previous releases of Abaqus.

The elastic contact energies and dissipative energies associated with contact stabilization and contact damping are associated with numerical effects that would be zero in idealized situations, such as infinite penalty stiffness or zero stabilization stiffness. Significant values of these output variables compared to other physically based energies in a model, such as internal energy (ALLIE), are sometimes indicative of solution inaccuracy. The contact constraint discontinuity work will tend to zero as the time increment size becomes very small. However, as discussed in “Energy computations in a contact analysis,” Section 1.1.25 of the Abaqus Example Problems Guide, it is quite common for ALLCCDW to have a significant value without causing solution inaccuracy.

References:

Abaqus Analysis User’s Guide

- “Output to the output database,” Section 4.1.3
- “Abaqus/Standard output variable identifiers,” Section 4.2.1



- “Whole model contact-related energy variables” in “Defining general contact interactions in Abaqus/Standard,” Section 36.2.1

Abaqus Keywords Reference Guide

- *ENERGY OUTPUT
- *ENERGY PRINT

Abaqus Example Problems Guide

- “Energy computations in a contact analysis,” Section 1.1.25

12.5 Magnitude output of vector nodal variables

Product: Abaqus/Explicit

Benefits: You can now output the magnitude of translational vector nodal quantities for displacement, velocity, acceleration, and reaction force.

Description: Four new nodal output variables have been introduced: spatial displacement magnitude UMAG, spatial velocity magnitude VMAG, spatial acceleration magnitude AMAG, and reaction force magnitude RFMAG. These new output variables are available for history output requests.

References:**Abaqus Analysis User's Guide**

- “Output to the output database,” Section 4.1.3
- “Abaqus/Explicit output variable identifiers,” Section 4.2.2

Abaqus Keywords Reference Guide

- *NODE OUTPUT

12.6 Equivalent plastic strain rate

Product: Abaqus/Explicit

Benefits: You can now output the equivalent plastic strain rate for both history and field output element output requests.

Description: A new element output variable, PEEQR, representing the equivalent plastic rate that is used for the evaluation of strain-dependent materials has been introduced. This quantity accounts for the filtering factor used to alleviate nonphysical high-frequency oscillations. This output quantity is available for both field and history element output requests.



References:

Abaqus Analysis User's Guide

- “Output to the output database,” Section 4.1.3
- “Abaqus/Explicit output variable identifiers,” Section 4.2.2
- “Material data definition,” Section 21.1.2

Abaqus Keywords Reference Guide

- *ELEMENT OUTPUT

12.7 Principal stresses with bounding values for output field filtering

Product: Abaqus/Explicit

Benefits: You can now apply a filter with a bounding value for maximum, intermediate, or minimum principal stress field output.

Description: When applying field filtering to the output, each component of a tensor or vector quantity is filtered individually and the maximum, minimum, or absolute maximum value and the limiting values are reported separately for each component. You can, however, apply a filter directly to an invariant. Three new values for the invariant parameter on field filtering are introduced: MAXP representing the maximum principal stress, INTERMP representing the intermediate principal stress, and MINP representing the minimum principal stress.

References:

Abaqus Analysis User's Guide

- “Output to the output database,” Section 4.1.3

Abaqus Keywords Reference Guide

- *ELEMENT OUTPUT
- *FILTER

12.8 Requesting reference sink temperature and film coefficient field output in Abaqus/CAE

Product: Abaqus/CAE

Benefits: You can now request reference sink temperatures and reference film coefficient values in Abaqus/CAE, which increases the coverage of Abaqus product functionality.



Description: For coupled thermal-stress, coupled thermal-electrical, coupled thermal-electrical-structural, and heat transfer procedures, the following field output variables are now available in Abaqus/CAE:

- SINKTEMP: Reference sink temperature
- FILMCOEF: Reference film coefficient value

Abaqus/CAE Usage:

Step module:

Field output request editor: **Output Variables: Thermal: SINKTEMP and FILMCOEF**

Reference:

Abaqus/CAE User's Guide

- “Creating and modifying output requests,” Section 14.4.5

12.9 Generating tabular reports for free body cuts from multiple frames

Product: Abaqus/CAE

Benefits: You can now create tabular reports of the resultant forces and moments in free body cuts from multiple frames. This enhancement is also available for free body cuts based on view cuts.

Description: When selecting options for free body cut reports, you can choose the frames from which Abaqus/CAE reads data, as shown in Figure 12–1. You can choose to use the current step/frame, all active steps/frames, or selected active steps/frames. For multiple frames, the time associated with each frame is included in the report.

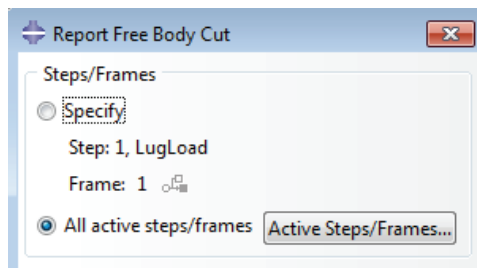


Figure 12–1 Selecting multiple steps/frames for free body cut reports.

Abaqus/CAE Usage:

Visualization module:

Report→Free Body Cut; All active steps/frames



Reference:

Abaqus/CAE User's Guide

- “Selecting options and thresholds for free body cut reports,” Section 54.3.4, in the HTML version of this guide

12.10 Selecting the area centroid as the summation point for free body cuts

Product: Abaqus/CAE

Benefits: A new option is available for specifying the summation point for free body cuts.

Description: For the summation point in free body calculations, you can now specify that the area centroid or the nodal average be used.

In previous releases the **Centroid of cut** option in the **Edit Free Body Cut** dialog box placed the summation point at the average of the nodal coordinates of the selected elements of the free body cross-section. That behavior is now obtained by choosing the **Nodal average** option.

Now when you choose the **Centroid of cut** option for the summation point, the area centroid of the polygon formed by the external boundary edges of the selected faces is used in the free body calculations. Figure 12–2 shows the available options for the summation point in free body cuts.

Abaqus/CAE Usage:

Visualization module:

Tools→**Free Body Cut**→**Create**; **Summation Point**: **Centroid of cut** or **Nodal average**

Reference:

Abaqus/CAE User's Guide

- “Creating or editing a free body cut,” Section 67.2, in the HTML version of this guide

12.11 Reading X – Y data from free bodies defined on view cuts

Product: Abaqus/CAE

Benefits: Abaqus/CAE can now read X – Y data from free body cuts defined on view cuts; previously, only X – Y data from free body cuts that were defined using nodes and elements could be used to create X – Y data sets.

Description: Abaqus/CAE reads X – Y data from all the active free body cuts defined in your session. If more than one slice is specified for free bodies based on view cuts, an X – Y data set is created for each slice.

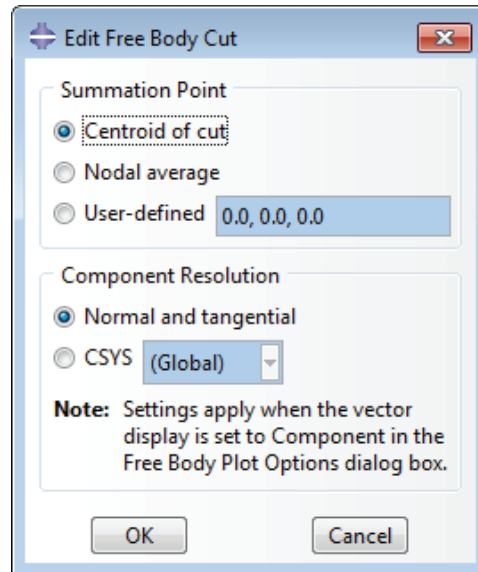


Figure 12–2 Summation point options.

Abaqus/CAE Usage:

Visualization module:

Tools→XY Data→Create; Free body

Reference:

Abaqus/CAE User's Guide

- “Reading *X–Y* data from all active free body cuts,” Section 47.2.4

12.12 Performing display group Boolean operations on linked viewports

Product: Abaqus/CAE

Benefits: In the Visualization module you can now perform selected display group operations simultaneously in all linked viewports. You can also display a list of items common between linked viewports when you create display groups.

Description: You can now perform selected display group operations simultaneously in all linked viewports. If you select specific items for the display group by names or labels, the same display group operation is performed in all linked viewports that share the items specified. This option applies only in the Visualization module.



OUTPUT AND VISUALIZATION

When you are creating a display group, you now have the option to display items common to all linked viewports in the **Method** list. For **Part instances**, viewports containing model databases or output databases are considered; for other model components, only viewports containing output databases are considered.

Abaqus/CAE Usage:

Visualization module:

Viewport→**Linked Viewports**: toggle **Link viewports**, toggle **Display groups**

Tools→**Display group**→**Create**: toggle **Show common list between linked viewports**

References:

Abaqus/CAE User's Guide

- “Linking viewports,” Section 4.5.2, in the HTML version of this guide
- “Creating or editing a display group,” Section 78.2.1, in the HTML version of this guide



13. User subroutines, utilities, and plug-ins

This chapter discusses additional user programs that can be run with Abaqus. It provides an overview of the following enhancements:

- “**VEXTERNALDB**: User subroutine that gives control to the user at key moments of the analysis,” Section 13.1
- “**VOCHARLENGTH**: User subroutine to define the characteristic element length at a material point,” Section 13.2
- “Enhancements to user subroutine **UMAT**,” Section 13.3
- “Allocatable arrays,” Section 13.4
- “Parallel user subroutines,” Section 13.5

13.1 VEXTERNALDB: User subroutine that gives control to the user at key moments of the analysis

Product: Abaqus/Explicit

Benefits: You can now use user subroutine **VEXTERNALDB** to dynamically exchange data among Abaqus user subroutines and with external programs or files.

Description: User subroutine **VEXTERNALDB** is called automatically once at the beginning of the analysis, at the beginning of each step, before each increment, at the start of each increment, at the end of each increment, at the end of each step, and finally at the end of the analysis; thus ceding control to you at key moments. You can modify the suggested time increment and trigger output of restart data, if desired. You can also instruct Abaqus to skip the remainder of the current step or to terminate the whole analysis. You can use the new allocatable local array utilities to coordinate data between various other user subroutines on a given process. Further, you can deploy the allocatable global arrays and the MPI mechanism within **VEXTERNALDB** to synchronize your data among the processes in domain parallel analyses.

References:

Abaqus User Subroutines Reference Guide

- “**VEXTERNALDB**,” Section 1.2.3
- “Obtaining parallel processes information,” Section 2.1.4
- “Allocatable arrays,” Section 2.1.23



13.2 VUCHARLENGTH: User subroutine to define the characteristic element length at a material point

Product: Abaqus/Explicit

Benefits: You can now use user subroutine **VUCHARLENGTH** to define the characteristic element length as a function of element topology, nodal and material point coordinates, and material orientation.

Description: The characteristic element length defined in user subroutine **VUCHARLENGTH** is used by Abaqus in regularization schemes needed to mitigate mesh dependency in constitutive models that include strain softening. It can be used with the built-in Abaqus material models as well as with user subroutine–based material models.

References:

Abaqus Keywords Reference Guide

- *CHARACTERISTIC LENGTH

Abaqus User Subroutines Reference Guide

- “VUCHARLENGTH,” Section 1.2.11

Abaqus Verification Guide

- “VUCHARLENGTH,” Section 4.1.32

13.3 Enhancements to user subroutine UMAT

Product: Abaqus/Explicit

Benefits: You can now use user subroutine **UMAT** to define the damping part of the material response for frequency-domain viscoelastic material behavior, to define almost incompressible nonlinear elastic response for hybrid elements using a total hybrid formulation, and to fully incompressible nonlinear elastic response for hybrid elements.

Description: Viscoelastic behavior in the frequency domain requires specification of a storage modulus (material stiffness) and a loss modulus (material damping). The **UMAT** interface and implementation has been enhanced to support the specification of both material stiffness and material damping. This feature allows you to define viscoelastic response with a complex frequency dependence that is not supported by the built-in models in Abaqus/Standard.

The default formulation used by Abaqus when a user material is used to define the response of a hybrid element is suitable for material models that use an incremental formulation (for example, metal plasticity) but is not consistent with the total formulation that is commonly used for hyperelastic materials. In the latter situation the default formulation may lead to convergence problems. Such convergence problems may be observed, for example, when an almost incompressible nonlinear elastic user material is subjected to



large deformations. Abaqus/Standard now provides an alternate total formulation for such situations. This formulation is consistent with the native almost incompressible formulation used by Abaqus for hyperelastic materials and works better than the default formulation in these situations. Abaqus/Standard also provides, for the first time, the capability to define a fully incompressible response (for hybrid elements) through user subroutine **UMAT**. The total hybrid formulation can be implemented with minimal changes to an existing **UMAT**. For fully incompressible materials, the **UMAT** needs to define the deviatoric part of the material response only.

References:

Abaqus Keywords Reference Guide

- *USER MATERIAL

Abaqus User Subroutines Reference Guide

- “UMAT,” Section 1.1.41

13.4 Allocatable arrays

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CFD

Benefits: To facilitate data accumulation and transfer between user subroutines, Abaqus provides utility functions for users to create and manage their own dynamic storage in the form of allocatable arrays.

Description: Two kinds of arrays are supported: global and thread-local.

Global arrays are visible and accessible from all threads in an executable. To prevent race conditions, write access to these arrays from multiple threads must be protected by mutexes. Using mutexes on every access will incur a performance penalty. In some situations it is possible to avoid this by restricting all threads to operate on non-intersecting ranges of a global array. Another alternative is to use thread-local arrays.

Thread-local arrays are local to a thread. Each thread holds its own private copy of an array. They are deliberately not shared, and, thus, do not need to be locked or protected from competing threads. In fact, one thread cannot cross-reference a local array of another thread. However, all user subroutines running in one thread can access all arrays of that thread. Without the need for locking, thread-local arrays can be accessed concurrently and, thus, are faster than global arrays.

These arrays are meant as a thread-safe replacement for common blocks. You can create any number of global or thread local arrays, in either Fortran or C++. At the time of creation, you assign an integer ID to each. Later, in other user subroutines, you can reference these arrays simply by their IDs and use them as native C++ or Fortran arrays.

Reference:

Abaqus User Subroutines Reference Guide

- “Allocatable arrays,” Section 2.1.23



13.5 Parallel user subroutines

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CFD

Benefits: Parallel user subroutines are supported in all parallel modes: thread-parallel and MPI-parallel.

Description: Infrastructural and procedural changes have been put in place to support parallel execution of Abaqus User Subroutines.

For threads, a number of parallel primitives has been exposed in Fortran that allow you to write safe multithreaded code in Fortran. These primitives have also been exposed in the C++, although this language has its own facilities for writing multithreaded code.

For MPI, full native support has been enabled and provided for both Fortran and C++. All MPI calls are available from within Abaqus user subroutines.

Special arrangement have been made to provide thread-shared and thread-local storage for users' needs.

References:

Abaqus Analysis User's Guide

- “Parallel execution: overview,” Section 3.5.1

Abaqus User Subroutines Reference Guide

- “Allocatable arrays,” Section 2.1.23



14. Abaqus Scripting Interface

This chapter discusses using the Abaqus Scripting Interface to write user scripts. Abaqus makes every attempt to be backward compatible and can execute most Abaqus Scripting Interface scripts from previous releases of Abaqus. However, backward compatibility is not guaranteed beyond several releases of Abaqus, and it is recommended that you upgrade your commands to the most recent release. A complete list of Abaqus Scripting Interface commands that have changed is included in “Summary of Abaqus Scripting Interface changes between Abaqus 6.13 and Abaqus 6.14” in the Abaqus Scripting Reference Guide. This chapter provides an overview of the following enhancements:

- “Enhancements to the *addData* method,” Section 14.1
- “Enhancement to the *nodes* member of the OdbSet object,” Section 14.2

14.1 Enhancements to the *addData* method

Product: Abaqus/CAE

Benefits: The *addData* method is enhanced for the C++ ODB API.

Description: For the C++ ODB API, the *addData* method is enhanced to support adding double precision nodal data. In addition, the performance of the *addData* method is improved for adding large amounts of data.

Reference:

Abaqus Scripting Reference Guide

- “addData,” Section 61.7.10

14.2 Enhancement to the *nodes* member of the OdbSet object

Product: Abaqus/CAE

Benefits: The *nodes* member of the OdbSet object is enhanced for the Python ODB API and C++ ODB API.

Description: For the Python ODB API and C++ ODB API, the *nodes* member of the OdbSet object is enhanced to return valid nodes that constitute the surface for element-based surface sets in addition to node-based surface sets.



Abaqus SCRIPTING INTERFACE

References:

Abaqus Scripting Reference Guide

- “OdbSet object,” Section 34.23
- “OdbSet object,” Section 61.24



15. Summary of changes

This section summarizes the changes and the additions that have been made to the items that define an Abaqus model, including elements, keywords, user subroutines, and output variables. For more information on these modifications, refer to the preceding chapters.

The following identifiers are used:

- new** New in 6.14.
- mod** Existed in 6.13 but has been modified or enhanced in 6.14.
- rem** Existed in 6.13 but has been removed in 6.14.
- (S) New, modified, or removed in Abaqus/Standard.
- (E) New, modified, or removed in Abaqus/Explicit.
- (C) New, modified, or removed in Abaqus/CFD.

15.1 Changes in Abaqus elements

This section summarizes the changes and the additions that have been made to the elements that can be used in an Abaqus model.

- new (S)** AC3D5
5-node linear acoustic pyramid element.

15.2 Changes in Abaqus options

This section summarizes the changes and the additions that have been made to the options that define an Abaqus model.

- mod (E)** *ADAPTIVE MESH REFINEMENT
Use the new COARSENING parameter to specify if refinement can be removed once the refinement criteria are no longer met.
- mod (S)(E)(C)** *AMPLITUDE
The DEFINITION parameter can now be used to define the amplitude via co-simulation with a logical modeling program.
- new (S)** *BEAM SECTION OFFSET
Define an offset for the beam cross-section origin.



SUMMARY OF CHANGES

- mod (S)** ***BOUNDARY**
The PHANTOM parameter can now be set equal to EDGE to apply boundary conditions to a phantom node located at an element edge in enriched elements.
- mod (S)** ***CFLOW**
The PHANTOM parameter is now available to apply concentrated flows to a phantom node located at an element edge or located coincident with a specified real node in enriched elements.
- new (E)** ***CHARACTERISTIC LENGTH**
Define characteristic element length at a material point.
- mod (S)** ***CONTACT OUTPUT**
The SURFACE parameter is now available in Abaqus/Standard for small-sliding contact between cracked surfaces in enriched elements.
- mod (S)(E)** ***CO-SIMULATION**
The PROGRAM parameter is no longer set to a value of LOGICAL for co-simulation with Dymola. The parameter setting PROGRAM=MULTIPHYSICS is now used for Dymola co-simulation.
- mod (S)** ***CREEP**
The LAW parameter can now be set to choose an Anand law, a Darveaux law, or a double power law.
- mod (S)(E)** ***DAMAGE INITIATION**
The R CRACK DIRECTION parameter is available for calculating the averaged stress/strain used to obtain the crack propagation direction.
- new (E)** ***DOMAIN DECOMPOSITION**
Define a region for domain decomposition and/or define constraints on the domain decomposition.
- mod (S)(E)** ***ELASTIC**
The COMPRESSION FACTOR parameter is now available for defining different elastic moduli in tension and compression for uncoupled traction-separation elastic behavior.
- mod (S)(E)** ***FASTENER**
The ATTACHMENT METHOD can now take the value CONNECTORDIRECTION to establish fastening points on the respective surfaces based on the axial direction of the connector element associated with the fastener.



- mod (E)** ***FILTER**
The INVARIANT parameter can now take the values MAXP, INTERMP, and MINP to apply the filtering to the maximum principal stress, the intermediate principal stress, or the minimum principal stress, respectively.
- new (S)** ***FLEXIBLE BODY**
Generate a flexible body from a substructure.
- mod (S)** ***FLUID CAVITY**
The ADDED VOLUME parameter is now available in Abaqus/Standard.
- mod (S)(E)** ***INITIAL CONDITIONS**
The TYPE parameter now supports a new value, DAMAGE INITIATION. A new parameter, CRITERION may be set to DUCTILE, SHEAR, or MSFLD to specify the damage initiation criterion for which the initial measure is being provided.
- new (S)** ***MATRIX CHECK**
Check the quality of the generated stiffness and mass matrices.
- mod (S)** ***MATRIX GENERATE**
Use the new optional FIELD, MPC, and SOURCE parameters to indicate that matrices are generated for the structural or acoustic parts of the model, to apply the multipoint constraints during the matrix generation, and to generate matrices including contributions from the finite elements or from the matrix input, respectively.
- mod (C)** ***TURBULENCE MODEL**
The TYPE parameter can now be set to specify the realizable k - ε turbulence model.
- mod (S)** ***USER MATERIAL**
The new HYBRID FORMULATION parameter is now available in Abaqus/Standard. It can be set to TOTAL or to INCOMPRESSIBLE to define almost-incompressible or incompressible nonlinear elastic user material response.
- mod (S)(E)** ***VISCOELASTIC**
The LAW parameter can now be set to choose a power law model.
- mod (S)** ***VISCOUS**
The LAW parameter can be set to choose an Anand law, a Darveaux law, or a double power law.

15.3 Changes in Abaqus user subroutines

This section summarizes the changes and the additions that have been made to user subroutines that can be used in an Abaqus model.



SUMMARY OF CHANGES

mod (S)

UCREEPNETWORK

Four new variables can be defined:

- **outputData(io_creep_deqcreepinc_dinv1)**,
the derivative: $\partial\Delta\bar{\epsilon}^{cr}/\partial\bar{I}_1$
- **outputData(io_creep_deqcreepinc_dinv2)**,
the derivative: $\partial\Delta\bar{\epsilon}^{cr}/\partial\bar{I}_2$
- **outputData(io_creep_deqcreepinc_ddetf)**,
the derivative: $\partial\Delta\bar{\epsilon}^{cr}/\partial J$
- **outputData(io_creep_deqcreepinc_dpress)**,
the derivative: $\partial\Delta\bar{\epsilon}^{cr}/\partial p$

Four variables can now be passed in for information:

- **r_array(ir_creep_inv1)**,
the first invariant, $\bar{I}_1$, of the left Cauchy-Green strain tensor, $\bar{\mathbf{B}}$
- **r_array(ir_creep_inv2)**,
the second invariant, $\bar{I}_2$, of the left Cauchy-Green strain tensor, $\bar{\mathbf{B}}$
- **r_array(ir_creep_detf)**,
the determinant of the deformation gradient, $\mathbf{F}$
- **r_array(ir_creep_press)**,
Kirchhoff pressure

mod (S)

UEXTERNALDB

The **LOP** variable to be passed in for information can take new values. **LOP=5** indicates that the user subroutine is being called at the start of a step. **LOP=6** indicates that the user subroutine is being called at the end of a step.

mod (S)

UMAT

Variable **JSTEP(1)** can now be passed in to identify the step number.

new (S)

UXFEMNONLOCALWEIGHT

User subroutine to define the weight function used to compute the average stress/strain to determine the crack propagation direction.

new (E)

VEXTERNALDB

User subroutine that gives control to the user at key moments of the analysis so that data can be exchanged dynamically among Abaqus user subroutines and with external programs or files.

new (E)

VUCHARLENGTH

User subroutine to define characteristic element length at a material point.



15.4 Changes in Abaqus output variable identifiers

This section summarizes the changes and the additions that have been made to output variable identifiers used in Abaqus.

Element integration point variables

new (S)(E)	MMIXDME Mode mix ratio during damage evolution.
new (S)(E)	MMIXDMI Mode mix ratio at damage initiation.
new (E)	PEEQR Equivalent plastic strain rate.

Whole element variables

new (S)	ALEAKVRBXFEM Accumulated leak-off flow volume at the bottom cracked surface of the enriched element.
new (S)	ALEAKVRTXFEM Accumulated leak-off flow volume at the top cracked surface of the enriched element.
new (S)	GFVRXFEM Gap fluid volume rate of the enriched element.
new (S)	LEAKVRBXFEM Leak-off flow rate at the bottom cracked surface of the enriched element.
new (S)	LEAKVRTXFEM Leak-off flow rate at the top cracked surface of the enriched element.
new (S)	PFOPENXFEM Fracture opening of the enriched element.
new (S)	PORPRES Fluid pressure of the enriched element.

Nodal variables

new (E)	AMAG Magnitude of translational accelerations.
----------------	---



SUMMARY OF CHANGES

new (S)	CLINELOAD Contact load due to line contact from edge-to-surface and radial edge-to-edge constraints in units of force per length. The normal (CLINELOADN) and frictional shear (CLINELOADT) components are available only for general contact to the .odb file.
new (S)	CPOINTLOAD Contact load in units of force due to point contact from edge-to-edge constraints using the cross formulation. The normal (CPOINTLOADN) and frictional shear (CPOINTLOADT) components are available only for general contact to the .odb file.
new (E)	RFMAG Magnitude of reaction forces.
new (E)	UMAG Magnitude of translational displacements.
new (E)	VMAG Magnitude of translational velocities.

Surface variables-nodal quantities

new (S)	CRKDISP Crack opening and relative tangential motions on cracked surfaces in enriched elements.
mod (S)	CSDMG Damage variable for cracked surfaces in enriched elements.

Total energy output quantities

new (S)	ALLCCDW Contact constraint discontinuity work.
new (S)	ALLCCE The sum of ALLCCEN and ALLCCET.
new (S)	ALLCCEN Contact constraint elastic energy in normal direction due to penalty constraint enforcement.
new (S)	ALLCCET Contact constraint elastic energy in tangential direction due to friction penalty constraint enforcement.



- new (S)** ALLCCSD
The sum of ALLCCSDN and ALLCCSDT.
- new (S)** ALLCCSDN
Contact constraint stabilization dissipation in normal direction.
- new (S)** ALLCCSDT
Contact constraint stabilization dissipation in tangential direction.

State and field variables

- rem (C)** ENSTROPHY
Enstrophy per unit mass.
- rem (C)** HELICITY
Dot product of vorticity and velocity.
- new (C)** QCRIT
Coherent structure visualizator, known as Qcriteria.

Turbulence variables

- new (C)** TURBOMEGA
Specific turbulent energy dissipation rate.
- new (C)** TURBVISCOSITYRATIO
Eddy to molecular viscosity ratio.



I. Product Index

Abaqus/Standard

Section 4.1	The AMS eigensolver can extract coupled structural-acoustic eigenmodes
Section 4.2	AMS eigensolver performance improvements
Section 4.3	Specifying additional volume in a fluid cavity
Section 4.5	Matrix quality check
Section 4.6	Flexible body generation
Section 5.2	Enhancements to the XFEM-based crack propagation capability
Section 5.3	SIMULIA Co-Simulation Engine API availability
Section 5.8	Vehicle ride comfort and durability co-simulation
Section 5.9	Logical-Physical co-simulation using Abaqus and Dymola
Section 5.10	Enhancements to the SIMULIA Co-Simulation Engine
Section 5.11	Enhancements for co-simulation
Section 5.12	MpCCI connector to the SIMULIA Co-Simulation Engine
Section 6.1	Enhancements to parallel rheological framework
Section 6.3	Enhancements to creep models
Section 6.4	User-defined frequency domain viscoelastic behavior
Section 6.5	New hybrid formulations for user-defined material behavior
Section 6.6	Different stiffness in tension versus compression for traction-separation elasticity
Section 6.7	Element deletion controlled by state variables
Section 7.2	Linear pyramid acoustic element
Section 8.1	Initial conditions on damage initiation
Section 8.3	Acoustic base motion for SIM-based procedures
Section 9.1	New attachment method for mesh-independent fasteners
Section 10.1	Edge-to-edge contact for general contact in Abaqus/Standard
Section 11.1	GPGPU acceleration of the AMS eigensolver
Section 11.2	GPGPU accelerated direct equation solver
Section 11.4	Job execution control enhancements
Section 11.5	SIM database utilities
Section 11.6	Enhancements for translating Abaqus data to modal neutral files
Section 12.1	Writing results from an Abaqus analysis to a SIM database
Section 12.2	Output of mode mix ratios during mixed mode delamination
Section 12.3	Output of last converged increment
Section 12.4	Output of contact-related energies
Section 13.4	Allocatable arrays
Section 13.5	Parallel user subroutines



Abaqus/Explicit

- Section 5.1 Defining a local coordinate system for materials in Eulerian elements
- Section 5.3 SIMULIA Co-Simulation Engine API availability
- Section 5.4 Enhancement to adaptive mesh refinement
- Section 5.6 Parallel enhancement for DEM analysis
- Section 5.7 Enhancements to CZone for Abaqus for composite crush simulation
- Section 5.8 Vehicle ride comfort and durability co-simulation
- Section 5.9 Logical-Physical co-simulation using Abaqus and Dymola
- Section 5.10 Enhancements to the SIMULIA Co-Simulation Engine
- Section 5.11 Enhancements for co-simulation
- Section 5.12 MpCCI connector to the SIMULIA Co-Simulation Engine
- Section 6.1 Enhancements to parallel rheological framework
- Section 6.2 Eulerian elements with anisotropic materials
- Section 8.1 Initial conditions on damage initiation
- Section 9.1 New attachment method for mesh-independent fasteners
- Section 9.2 Enhanced domain decomposition of thermal ties
- Section 10.2 Improved contact at complex intersections in Abaqus/Explicit
- Section 11.3 User control of domain decomposition
- Section 11.4 Job execution control enhancements
- Section 11.5 SIM database utilities
- Section 12.1 Writing results from an Abaqus analysis to a SIM database
- Section 12.5 Magnitude output of vector nodal variables
- Section 12.6 Equivalent plastic strain rate
- Section 12.7 Principal stresses with bounding values for output field filtering
- Section 13.1 **VEXTERNALDB**: User subroutine that gives control to the user at key moments of the analysis
- Section 13.2 **VUCHARLENGTH**: User subroutine to define the characteristic element length at a material point
- Section 13.3 Enhancements to user subroutine **UMAT**
- Section 13.4 Allocatable arrays
- Section 13.5 Parallel user subroutines

Abaqus/CFD

- Section 4.4 K-epsilon realizable turbulence model in Abaqus/CFD
- Section 5.3 SIMULIA Co-Simulation Engine API availability
- Section 5.11 Enhancements for co-simulation
- Section 8.2 Enhancements for prescribed conditions in Abaqus/CFD
- Section 11.4 Job execution control enhancements
- Section 11.5 SIM database utilities
- Section 12.1 Writing results from an Abaqus analysis to a SIM database



- Section 13.4 Allocatable arrays
- Section 13.5 Parallel user subroutines

Abaqus/CAE

- Section 2.1 Accessing the Learning Community from Abaqus/CAE
- Section 3.1 Sizing optimization
- Section 3.2 Optimization across multiple models
- Section 3.3 Viewing the progression of an optimization by combining output database files
- Section 3.4 Monitoring an optimization process
- Section 3.5 Additional criteria for creating soft elements
- Section 3.6 New design responses
- Section 3.7 Specifying the rate at which optimization data are saved
- Section 3.8 Writing optimization files
- Section 3.9 Switching the context for model instances in Abaqus/CAE
- Section 3.10 Associating geometric faces created from an orphan mesh with the mesh
- Section 3.11 Bidirectional import of parameters using the CATIA V6 Associative Interface
- Section 5.5 Enhancements to the XFEM-based crack propagation capability in Abaqus/CAE
- Section 7.1 Support for coupled temperature–pore pressure elements in Abaqus/CAE
- Section 10.3 Enhancement to the contact detection tool in Abaqus/CAE
- Section 12.1 Writing results from an Abaqus analysis to a SIM database
- Section 12.8 Requesting reference sink temperature and film coefficient field output in Abaqus/CAE
- Section 12.9 Generating tabular reports for free body cuts from multiple frames
- Section 12.10 Selecting the area centroid as the summation point for free body cuts
- Section 12.11 Reading X – Y data from free bodies defined on view cuts
- Section 12.12 Performing display group Boolean operations on linked viewports
- Section 14.1 Enhancements to the *addData* method
- Section 14.2 Enhancement to the *nodes* member of the OdbSet object

Abaqus/AMS

- Section 4.2 AMS eigensolver performance improvements
- Section 11.1 GPGPU acceleration of the AMS eigensolver

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' 3DEXPERIENCE 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 3DEXPERIENCE® Platform powers our brand applications, serving 12 industries, and provides a rich portfolio of industry solution experiences.

Dassault Systèmes, the 3DEXPERIENCE 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.

