

Knowledge Base

Information

QA Article: QA00000008241e
 Applicable Level: 6.9
 1)
 Last Update Date: 29.10.2020
 Rating:
 Views: 864



Abaqus/CAE Plug-in Application to Automatically Generate Point-to-Point Springs, Dashpots, MPC constraints, or Wires for Connectors

Portfolio / Domain: SIMULIA Abaqus Unified FEA / SIMULIA Abaqus Unified FEA
Product: SIMULIA Abaqus/CAE

QUESTION

How can I generate multiple spring/dashpot elements, wires for connectors, or MPC constraints between proximal nodes of neighboring parts in Abaqus/CAE?

ANSWER

(The following applies to Release 6.9 and higher)

An Abaqus/CAE plug-in application for this purpose is attached below. Given two node sets, for each node in the first set, this plug-in will locate the closest node in the second set and generate either an MPC constraint, a wire for connectors as part of a wire feature, or a spring element and/or a dashpot element. This allows for the automatic generation of numerous wires, springs, dashpots, or MPCs between the nodes of neighboring components.

Installation

To install the plug-in, save the attached archive file to one of the following directories:

`abaqus_dir\abaqus_plugins` where `abaqus_dir` is the Abaqus parent directory

`home_dir\abaqus_plugins` where `home_dir` is your home directory

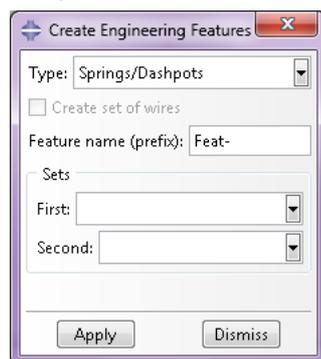
`current_dir\abaqus_plugins` where `current_dir` is the current directory

Note that if the `abaqus_plugins` directory does not exist in the desired path, it must be created. The `plugin_dir` directory can also be used, where `plugin_dir` is a directory specified in the `abaqus_v6.env` file by the environment variable `plugin_central_dir`. You can store plug-ins in a central location that can be accessed by all users at your site if the directory to which `plugin_central_dir` refers is mounted on a file system that all users can access. For example, `plugin_central_dir = r'\\fileServer\sharedDirectory'`

On Windows platforms, right click on the archive file and select **WinZip** → **Extract to here**. On Linux platforms, type **unzip engFeatures.zip** at the command prompt. A folder named `abq_EngFeatures` and a file named `engFeatures_plugin.py` will be extracted. Note that the plug-in will not function properly if this procedure is not followed.

Usage:

1. Create assembly level node sets containing the nodes from each part instance for which spring/dashpot elements, wires for connectors, or MPCs are to be generated. The sets can be created from nodes, elements, or geometry.
2. From the **Interaction** module, select **Plug-ins** → **Create Engineering Features....** The following dialog box is displayed:

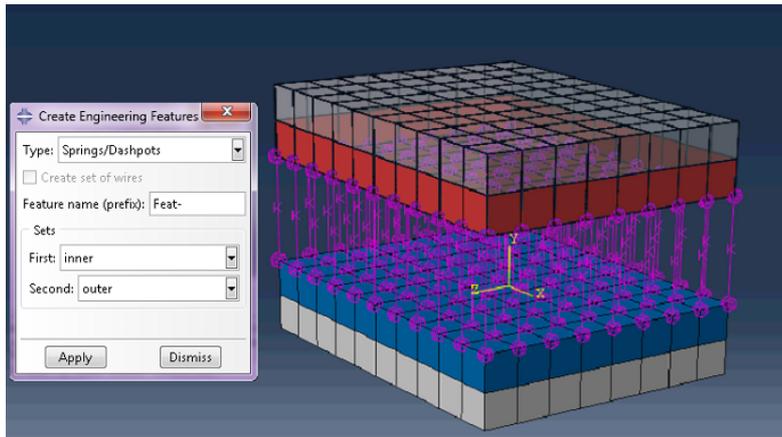


3. Specify the feature type under **Type**.
4. If **Type** is **Wires for Connectors**, the **Create set of wires** check button will be enabled. Select it if you want a wire set to be created in addition to the wire feature containing the indicated wires.
5. Specify the feature name under **Feature name (prefix)**. For springs/dashpots and for wires, a single feature having a unique name and encompassing multiple entities will be created. For MPC features, the prefix name will be used and any existing features with the same name will be overwritten. For wire features the **Feature name (prefix)** text field is disabled and ignored because newly-created wire feature names are always of the form **Wire- <N >**, where **<N >** is the next available wire ID.
6. Select the desired node sets under **First** and **Second** and click **Apply**.

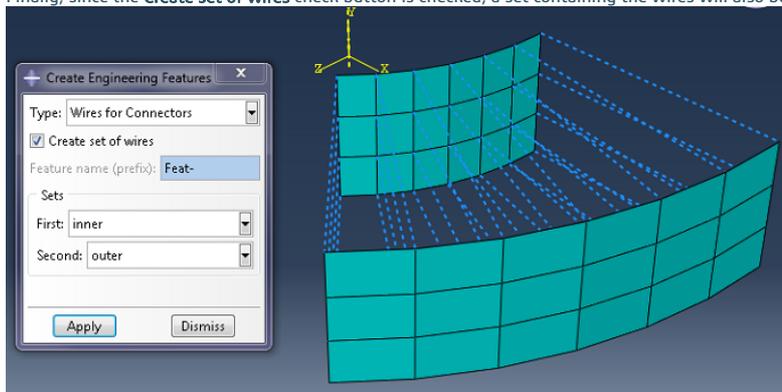
Notes

1. The node sets specified can be of either geometry, node, or element type but must be defined at the assembly level.
2. For spring/dashpot elements, an assembly level Engineering Feature is generated containing all the elements. By default, spring (dashpot) elements with a spring coefficient of 1 are generated; this can easily be modified by editing the feature from the Model Tree. Note that spring elements are created by default; the feature must be edited to include dashpot elements and to exclude spring elements. In addition, non-linear spring or dashpot definitions can be included while editing the feature.
3. For MPCs, individual MPC constraints are generated with the prefix given followed by the node label in the first set.

4. The plug-in will generate one spring/dashpot or one MPC or one wire for each node in the first set given. The second node will be the closest node in the second set. Therefore, after the springs or MPCs or wires have been generated, it is your responsibility to check that the connectivity is appropriate, especially for highly curved surfaces. Note also that the processing time of the plug-in will increase with the number of nodes in the sets.
5. A sample model is shown below. The node set 'Set-1' is on the upper surface of the bottom part instance. The node set 'Set-2' is on the lower surface of the top part instance. For all nodes in set 'Set-1', spring elements will be generated between each node and the closest node in set 'Set-2'. Since the two instances specified have matching meshes, a grid of spring elements is generated.



6. The procedure for generating a large number of connectors between two assembly-level node sets is:
 1. Use this plug-in to create an assembly-level wire feature containing the many individual wires that will each correspond to a new connector.
 2. Create the Connector Section for the connector elements.
 3. Create a Connector Section Assignment that will map the Connector Section to the wire feature.
7. Another sample model is shown below. The node set 'inner' contains the nodes on the inner surface (the surface closest to the origin) and node set 'outer' contains the nodes on the outer surface. For all nodes in set 'inner', a wire for connectors will be generated between each node and the closest node in set 'outer'. In addition, new reference points will be generated at the ends of each wire. They will not be displayed because reference point visibility is automatically disabled to minimize graphical clutter. (If you want to see them, you can go to View->Assembly Display Options->Datum tab to turn their visibility back on.) Furthermore, a new feature that contains all of these wires will also be generated. Since the two sets specified have matching meshes, a grid of wires is generated. Finally, since the **Create set of wires** check button is checked, a set containing the wires will also be created.



Revision History

05 May 10	Release of Version 1.1-1
26 Sep 2017	Release of Version 1.2-1. Added support for "Wires for Connectors" option

Disclaimer

The attachments to this article are subject to certain usage conditions. Please click here for details.

KEYWORDS **plug-in, plugin, spring, dashpot, mpc, wire, connector constraint**

ATTACHMENT

- [answer_4321_fig2.png](#)
- [answer_4321_fig1.png](#)
- [rtestModel.rpy.out](#)
- [QA8241_fig3.png](#)
- [rtestModel.rpy](#)
- [QA8241_fig2.png](#)
- [engFeatures.zip](#)
- [rtestModel.guilog](#)
- [QA8241_fig1.png](#)

SUBSCRIBE TO CHANGES

RATING **On a scale of 1-5, how would you rate the technical content of the article?**

Please rate this article...

LET US KNOW
IF THIS ARTICLE
NEEDS TO BE
ENHANCED

UNCLEAR

MISSING INFO

DUPLICATE

OUT OF DATE

ERROR DETECTED

[See Comments \(2\)](#)

MY FAVORITE CONTENT