

Knowledge Base

Information



Plug-in utility for creating slide line definitions

Portfolio / Domain: SIMULIA Abaqus Unified FEA / SIMULIA Abaqus Unified FEA
Product: n/a

QA Article: QA0000008810e
Last Update Date: 24.09.2020
Rating: 5.0
Views: 377

QUESTION

I use axisymmetric solid elements with asymmetric deformation (the CAXA family) in contact analyses. How can I make use of Abaqus/CAE to generate the necessary ISL elements and slide line definitions?

ANSWER

(The following applies to Versions 6.5 and higher.)

Modeling contact with CAXA elements requires slide line-based, rather than surface-based, interaction definitions. For each slide line, a list of spatially sequential node numbers is required along the line. Additionally, ISL-type elements are required on the opposing body. Neither slidelines or ISL elements are directly supported in Abaqus/CAE.

An Abaqus/CAE plug-in utility to speed the process of generating both the slide lines and the ISL element definitions is attached to this Answer. The plug-in automatically defines the ISL elements and slide lines based on surfaces you select in Abaqus/CAE.

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 SlideLine_Plugin.zip** at the command prompt. Folder named `abq_SlideLine` a file named `slnSlideLine_plugin.py` will be extracted. Note that the plug-in will not function properly if this procedure is not followed.

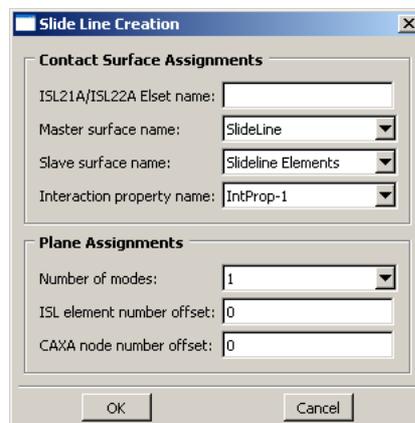
Usage

Abaqus/CAE does not currently support the CAXA element family; it does however, support the CAX family. Create your model (geometry, sections, materials, assembly, mesh, etc.) using regular axisymmetric (CAX family) elements. The slide lines and ISL elements can then be included in your model using the plug-in.

The plug-in uses surface definitions to create the ISL elements and the associated slide line. Within the context of a "master-slave" approach, the slide line serves as the "master" and the ISL elements as the "slave."

Move to the Interaction module and complete the following steps:

1. Create the surfaces on which the ISL elements and the slide line will reside. If there will be multiple ISL-slide line pairings, create all the necessary surfaces.
2. Next, create the contact interaction property; multiple properties can be created if needed for multiple slide line definitions. Note that contact pairs are *not* needed, just the interaction properties. Select **Interaction** → **Property** → **Create...** and complete the definition.
3. Select **Plug-ins** → **Tools** → **Slideline Creation...**. The following dialog box will appear:



4. In the **Contact Surface Assignments** panel, enter a name for the ISL element set. Select the "master" surface to locate the slide line, the "slave" surface to locate the ISL elements, and the interaction property name.

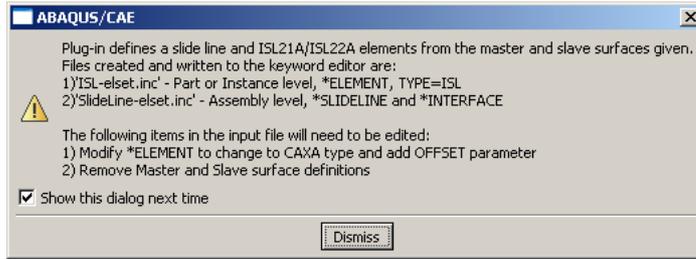
5. In the **Plane Assignments** panel, enter the **Number of modes** that will be used for the CAXA elements.

Next, enter the **ISL element number offset**; this is the number by which the ISL elements on each plane of the CAXA elements will be offset. For example, if the last ISL element on the 0° plane is number 1050, the CAXA element has one mode, and the offset is set to 1000, the corresponding element on the 180° degree plane will be number 2050. Note that the plug-in will automatically determine an appropriate starting number for the ISL elements.

Finally, enter the **CAXA node number offset**. This is the number by which the additional nodes on the planes of the CAXA elements will be offset. It must have the same value as the **OFFSET** parameter on the associated ***ELEMENT** definition, which we will manually set later.

6. Select **OK**.

The plug-in will execute and issue the following dialog:



As stated in the dialog, the plug-in creates two files:

- `ISL-ElsetName.inc`: This file will contain the keywords that define the ISL elements, where `ElsetName` is the name specified in the **Slide Line Creation** dialog box.
- `SlideLine-ElsetName.inc`: This file will contain the keywords that define the slide lines, where `ElsetName` is the name specified in the **Slide Line Creation** dialog box.

The plug-in will write two ***INCLUDE** keyword options (visible in the keyword editor; select **Model** → **Edit Keywords**) into the input file so that the aforementioned files are automatically included in the model definition.

7. Repeat steps 3-6 for each surface pairing that requires ISL element and slide line definitions. Note that multiple ***INCLUDE** options will be written to the input file as necessary- two for each ISL element-slide line pairing.

8. Move to the Job module. Create the job and write the input file.

9. The input file must then be edited as follows:

- The element type must be changed from CAX to CAXA on the ***ELEMENT** keyword. The **OFFSET** parameter on ***ELEMENT** must be set to the value chosen for the **CAXA node number offset** when the plug-in was executed.
- The surface definitions specified as "master" and "slave" when the plug-in was executed must be either commented or removed. They are not used with CAXA contact interactions and will cause errors in the analysis if left in the model definition.

Notes

The plug-in allows for the creation of multiple ISL-slide line pairings. The plug-in must be executed multiple times and separate ***INCLUDE** keyword options will be generated each time.

The plug-in requires input files that contain Parts and Assemblies.

If the plug-in is executed multiple times for a given slide line definition, multiple ***INCLUDE** options will still be written to the input file; that is, the previous definition is not overwritten. The last ***INCLUDE** option associated with the given ISL element set will be the most current and the previous ***INCLUDE** options should be commented or deleted. In addition, parts and assembly must be used otherwise ***INCLUDE** will not be written.

Revision History

Mar 27 2009	Release of plug-in version 1.3-1.
Nov 18 2010	Release of plug-in version 1.3-2. Remove keyword block deletion.
Mar 15 2011	Release of plug-in version 1.3-3. Fix error with odd/even division.
Mar 17 2011	Release of plug-in version 1.3-4. Merge changes of version 1.3-2 and 1.3-3.
April 14 2011	Release of plug-in version 1.3-5. Fix corner elements that has more than 2 nodes .

Disclaimer

The attachments on this article are subject to certain usage conditions. Please [click here](#) for details.

KEYWORDS

plugin, plug-in, python, custom, customize, customization, customapp, custom application, module, cu

ATTACHMENT

- [Answer_3077_Figure2.png](#)
- [Answer_3077_Figure1.png](#)
- [SlideLine_Plugin.zip](#)

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 \(1\)](#)

MY FAVORITE CONTENT