NTNU – NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF ENGINEERING
DEPARTMENT OF STRUCTURAL ENGINEERING

TKT4142 Finite Element Methods in Structural Engineering

Workshop 4

This workshop will briefly guide you through Case Study 4. Case Study 4 revisits the wooden beam from Case Study 1, but now with an additional hole in the center of the beam.

Section 1 introduces 3D solid parts in Abaqus by modelling a cantilever beam before we increase the complexity of the model to also include the hole at the steel rod support and the hole in the center in Section 2. Section 3 give a brief introduction to the Python scripting interface in Abaqus by using the model created in Section 1.

1. Modelling the Cantilever Beam Using 3D Solid Elements

The cantilever beam with dimensions and applied load is shown in Figure 1-1.

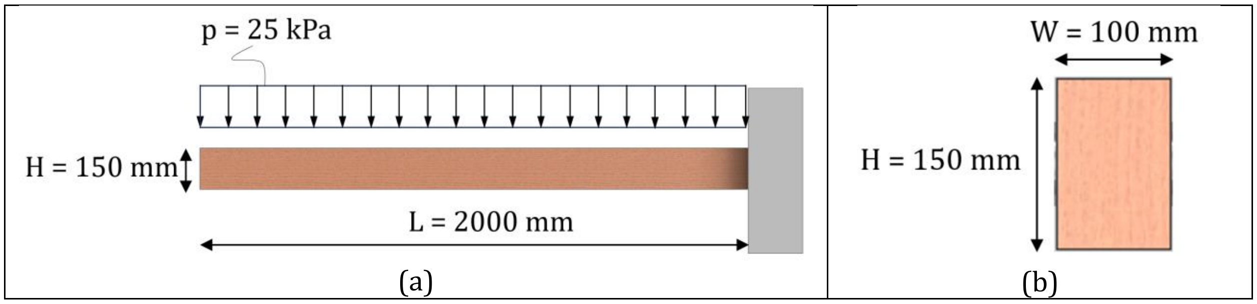


Figure 1-1 – Cantilever beam.

Load: $p = -0.025\ \text{N/mm}^2$ (Downwards)
Material data: $E = 10\,000\ \text{N/mm}^2, \nu = 0.30\ , \rho = 500\ \text{kg/m}^3, \sigma_y = 20\ \text{N/mm}^2$

Make sure your work directory is properly set.

Change the model's name by right click on the model (named Model-1) in the **Model Tree** to access the **Models menu.** Select **Rename…** and enter **BEAM** in the **Rename Model** dialog box. Select **OK.**

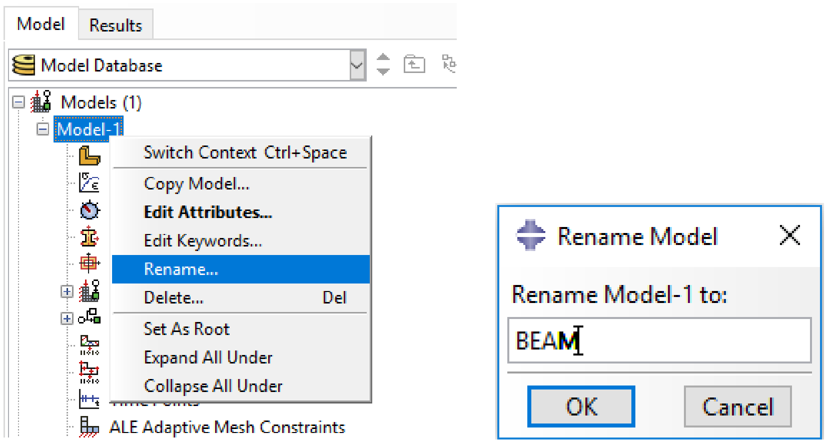
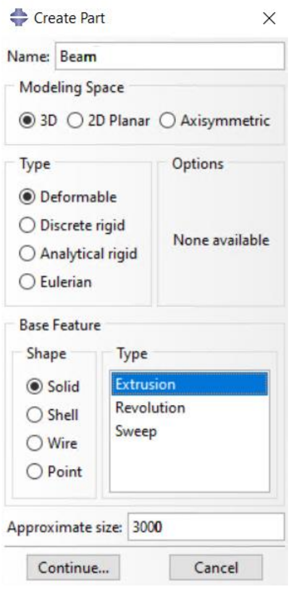


Figure 1-2 – Rename the model.

1.1. Creating a Part

In this section, you will create a three-dimensional, deformable solid body by sketching the two-dimensional profile of the beam (a rectangle) and extruding it.

1. In the Model Tree, double-click on **Parts** to create a new part in the model **BEAM**.
   The **Create Part** dialog box appears, as shown in Figure 1-3.
2. In the **Create Part** dialog box, name the part **Beam**, and specify an approximate size of **3000**. Accept the default settings of a three-dimensional, deformable body with a solid, extruded base feature. Click **Continue**. This takes you into the **sketch environment.**



Create Part

Name: Beam

Modeling Space: [✓ 3D] 2D Planar Axisymmetric

Type: [✓ Deformable] Discrete rigid Analytical rigid Eulerian

Options: None available

Base Feature

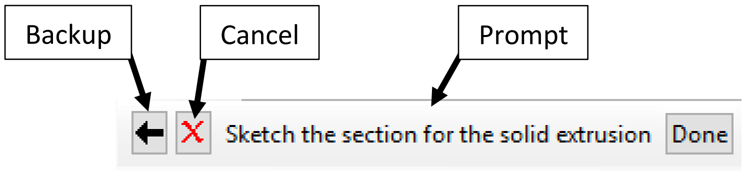
Shape: [✓ Solid] Shell Wire Point

Type: [✓ Extrusion] Revolution Sweep

Approximate size: 3000

Figure 1-3 – Create Part dialog box.

Abaqus/CAE displays text in the **prompt area** near the bottom of the window to guide you through the procedure, as shown in Figure 1-4. Click the **cancel** button to cancel the current task; click the **backup** button to cancel the current step in the task and return to the previous step.



Sketch the section for the solid extrusion

Figure 1-4 – Prompt area.

3. To sketch the profile of the cantilever beam, you need to select the **rectangle drawing** tool, as shown in Figure 1-5.

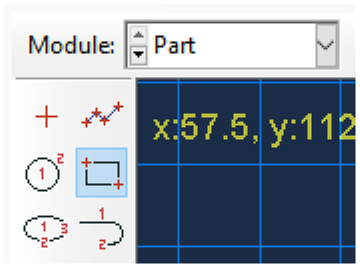


Figure 1-5 – Rectangle sketch tool.

4. In the viewport, sketch the rectangle using the following steps:
   a. You will first sketch a rough approximation of the beam and then use constraints and dimensions to refine the sketch. Select any two points as the opposite corners of the rectangle.
   b. Click mouse **Esc** or **Cancel** (✕) in the prompt area to exit the rectangle tool.
   c. The Sketcher automatically adds constraints to the sketch (in this case the four corners of the rectangle are assigned perpendicular constraints and one edge is designated as horizontal).
   d. Use the dimension tool ( ) to dimension the top and left edges of the rectangle. The top edge should have a horizontal dimension of **2000** mm, and the left edge should have a vertical dimension of **150** mm. See Figure 1-1. When dimensioning each edge, simply select the line, Left-click to position the dimension text, and then enter the new dimension in the prompt area.
   e. The final sketch is shown in Figure 1-6.

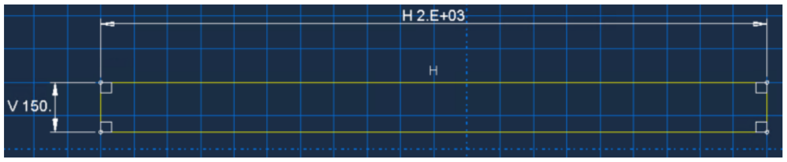
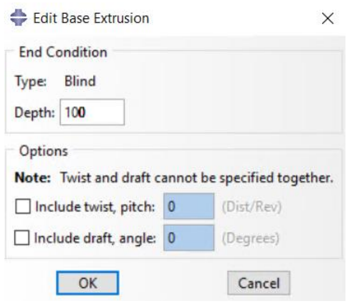


Figure 1-6 – Sketch of the rectangle.

5. Click **Done** in the prompt area to exit the sketcher.
6. Enter an extrusion depth of **100** in the **Edit Base Extrusion** dialog box, and click **OK**.



Type: Blind
Depth: 100
Note: Twist and draft cannot be specified together.
☐ Include twist, pitch: 0 (Dist/Rev)
☐ Include draft, angle: 0 (Degrees)

Figure 1-7 – Edit Base Extrusion dialog box.

7. Abaqus/CAE displays an isometric view of the new part, as shown in Figure 1-8(a). The Part **Beam** is now added to the **Model Tree** in Figure 1-8(b). If you want to do any changes to the sketch, double-click on **Section Sketch** (See Figure 1-8(b)).

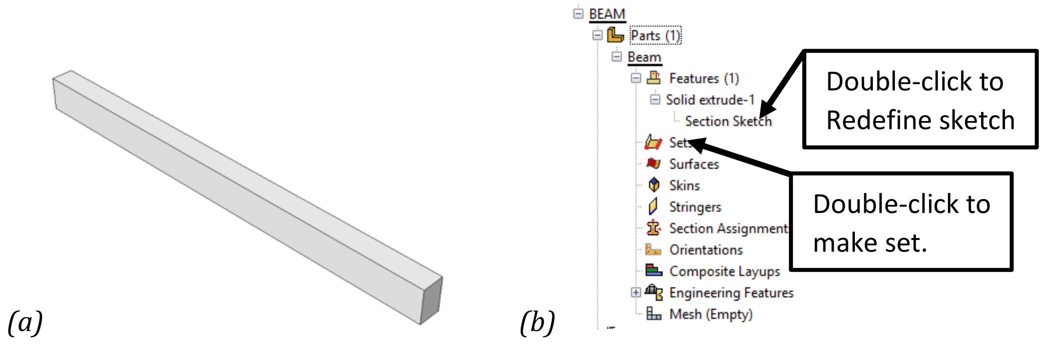
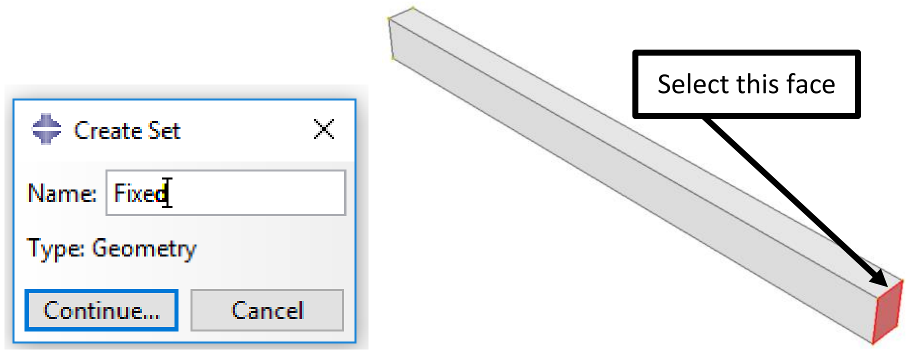


Figure 1-8 – Extruded part and Model Tree.

8. It is often convenient to define particular regions of the model to different sets. Loads and BC can then be added to these sets. It is also possible to extract values (Force/Displacement/ect.) from a set using History Output.
   In the **Model Tree** under **BEAM** → **Parts** → **Beam**, double-click **Sets** (See Figure 1-8(b)). Name this set **Fixed** and click **Continue...** in the **Create Set** dialog box. Select the left face of the model, as shown in Figure 1-9. Use the **Rotate View** tool ( ) in order to rotate the model.


Create Set

Name: Fixed

Type: Geometry

Figure 1-9 – Create Set dialog box and selected face for the set "Fixed".

1.2. Assigning Properties (Material and Section) to the Model

We will create a single elastic material with Young's modulus of 209×10^3 MPa and Poisson's ratio of 0.3. Next, we will create a homogenous solid section and assign it to the beam.

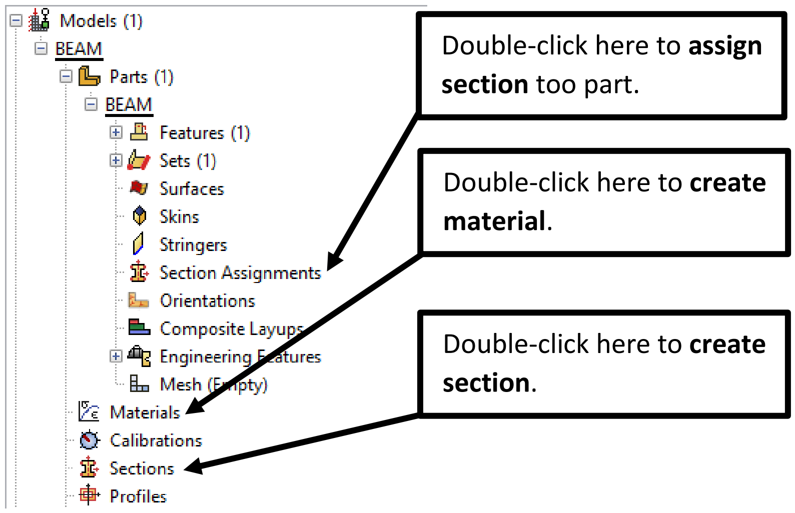
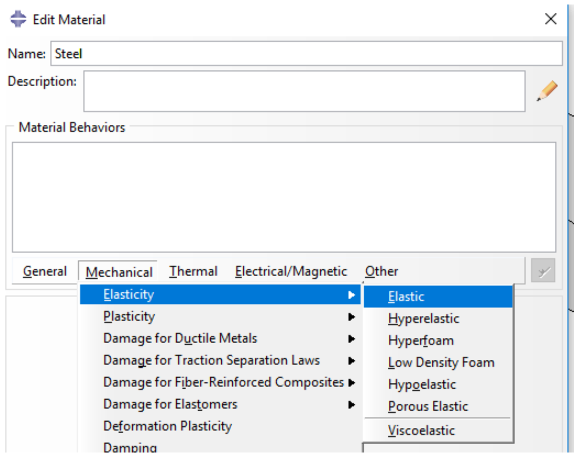
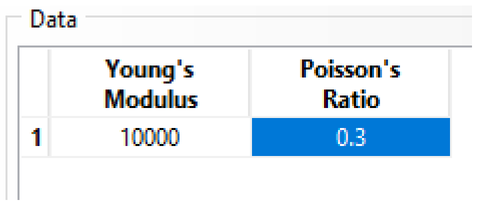


Figure 1-10 – The Model Tree.

Creating a Material

1. In the **Model Tree**, double-click **Materials** to create a new material in the model BEAM. See Figure 1-10. Notice that Abaqus/CAE switches to the Property module and the **Edit Material** dialog box appears.
2. In the **Edit Material** dialog box, name the material **Wood**. Notice the various options available in this dialog box.
3. From the material editor's menu bar, select **Mechanical** → **Elasticity** → **Elastic**, as shown in Figure 1-11(a).
   Abaqus/CAE displays the **Elastic** data form.
4. Enter a value of **10000** for **Young's Modulus** and a value of **0.3** for **Poisson's Ratio** in the respective fields, as shown in Figure 1-11(b).
5. Click **OK** to exit the material editor.

(a)


(b)


	Young's Modulus	Poisson's Ratio
1	10000	0.3

Figure 1-11 – (a) The Edit Material dialog box and (b) material editor.

Creating a Homogeneous Solid Section

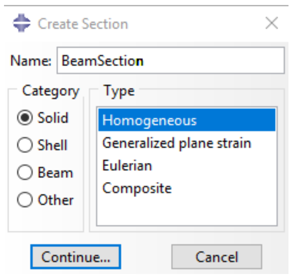
1. In the Model Tree, double-click **Sections** to create a new section in the model BEAM. See Figure 1-10.

The **Create Section** dialog box appears, shown in Figure 1-11(a).

2. In the **Create Section** dialog box:
    a. Name the section **BeamSection**.
    b. Accept the default category **Solid** and the default type **Homogeneous**.
    c. Select **Continue**.

   The **Edit Section** dialog box appears, shown in Figure 1-11(b).
3. In the **Edit Section** dialog box:
    a. Accept the default selection of **Wood** for the **Material** associated with the section.
    b. Accept the default value of **1** for **Plane stress/strain thickness**.
    Note: For three-dimensional solid geometry, this value is not used. It is only relevant for two-dimensional geometry.
    c. Click **OK**.

(a)


Create Section
Name: BeamSection
Category: (● Solid) ○ Shell ○ Beam ○ Other
Type: Homogeneous (selected), Generalized plane strain, Eulerian, Composite

(b)

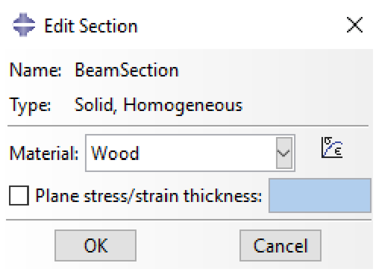

Edit Section
Name: BeamSection
Type: Solid, Homogeneous
Material: Wood
☐ Plane stress/strain thickness:

Figure 1-12 – (a) the Create Section dialog box and (b) the Edit Section dialog box.

Assign the Section to the Part

1. In the Model Tree, expand the branch for the part **Beam**. Double-click **Section Assignments** to assign a section to the part Beam, as shown in Figure 1-10. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. Click anywhere on the beam to select the entire part as the region to which the section will be assigned, as shown in Figure 1-13(a).
3. In the prompts, check **Create set.** See Figure 1-13(b). Call this set **Whole** and click **Done** in the prompt area to accept the selected geometry. This will make a new set called **Whole** and contains the whole part. This set can be found in the **Model Tree** together with the set made in Section 1.1.
   The **Edit Section Assignment** dialog box appears, as shown in Figure 1-14(a).
4. In the **Edit Section Assignment** dialog box, accept the default selection of **BeamSection** as the section definition, and click **OK**.
   Abaqus/CAE colors the beam green to indicate that the section has been assigned, as shown in Figure 1-14(b).

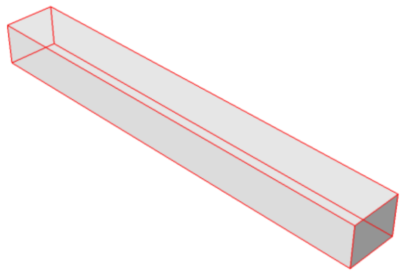
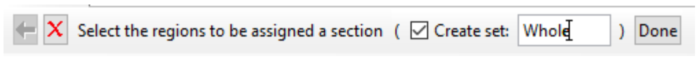
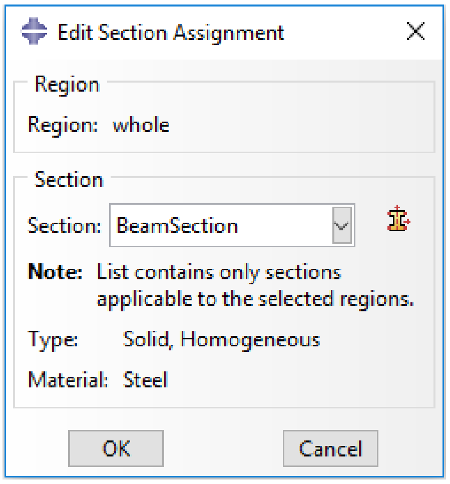
(a)

(b)

Select the regions to be assigned a section (☑ Create set: Whole)

Figure 1-13 – (a) The selected geometry and (b) the prompt area.

(a)

Edit Section Assignment

Region

Region: whole

Section

Section: BeamSection

Note: List contains only sections applicable to the selected regions.

Type: Solid, Homogeneous

Material: Steel

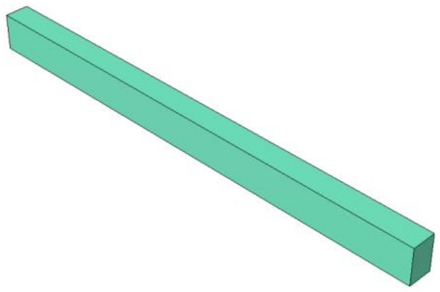
(b)


Figure 1-14 – (a) The Edit Section Assignment dialog box and (b) the part after a section is assigned.

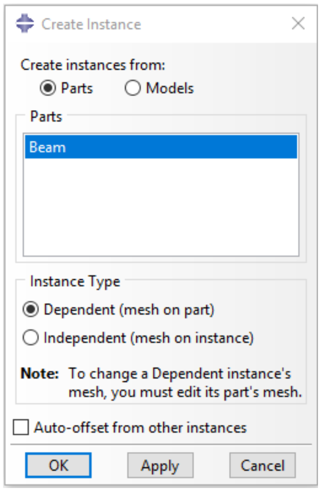
1.3. Create an Assembly

The assembly for this analysis consists of a single instance of the part **Beam**.

Assemble the Model:

1. In the **Model Tree**, expand the branch for the **Assembly** of the model **BEAM** and double-click **Instances** to create a new part instance as shown in Figure 1-15(b).
   Abaqus/CAE switches to the Assembly module, and the **Create Instance** dialog box appears. See Figure 1-15(a).
2. In the **Create Instance** dialog box, select **Beam**. Choose the instance type **Dependent (mesh on part)** and click **OK**.
   Abaqus/CAE displays the new part instance in the viewport.

(a)



Create Instance

Create instances from: [✓ Parts] Models

Parts: Beam

Instance Type: [✓ Dependent (mesh on part)] Independent (mesh on instance)

Note: To change a Dependent instance's mesh, you must edit its part's mesh.

☐ Auto-offset from other instances

(b)

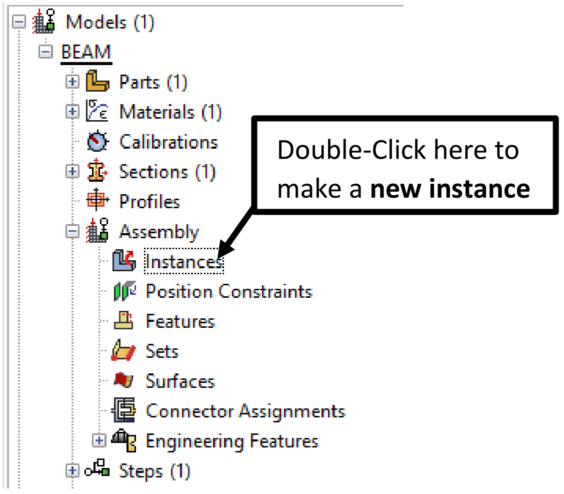


Figure 1-15 – (a) The Create Instance dialog box and (b) The Model Tree.

1.4. Defining the Step and Output data

In this simulation, we are interested in the static response of the cantilever beam to a pressure load applied over the beam top. This is a single event, so only a single analysis step is needed for the simulation. Consequently, this model will consist of two steps:

- An initial step, in which you will apply a boundary condition that constrains one end of the cantilever beam.
- A general, static analysis step, in which you will apply a pressure load to the top face of the beam.

Abaqus/CAE generates the initial step automatically, but you must create the analysis step yourself.

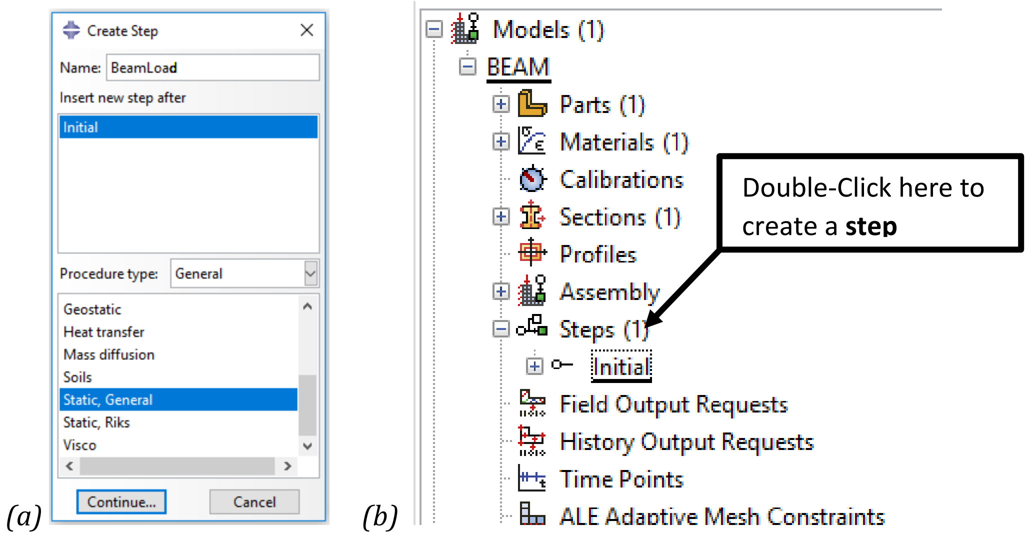
Create a General, Static Analysis Step

1. In the **Model Tree**, double-click **Steps** to create a new step in the model **BEAM**. See Figure 1-16 (b).
   Abaqus/CAE switches to the Step module, and the **Create Step** dialog box appears, shown in Figure 1-16(a).

2. In the **Create Step** dialog box:
   a. Name the step **BeamLoad**.
   b. From the list of available general procedures in the **Create Step** dialog box, select **Static**, **General** if it is not already selected.
   c. Click **Continue**.

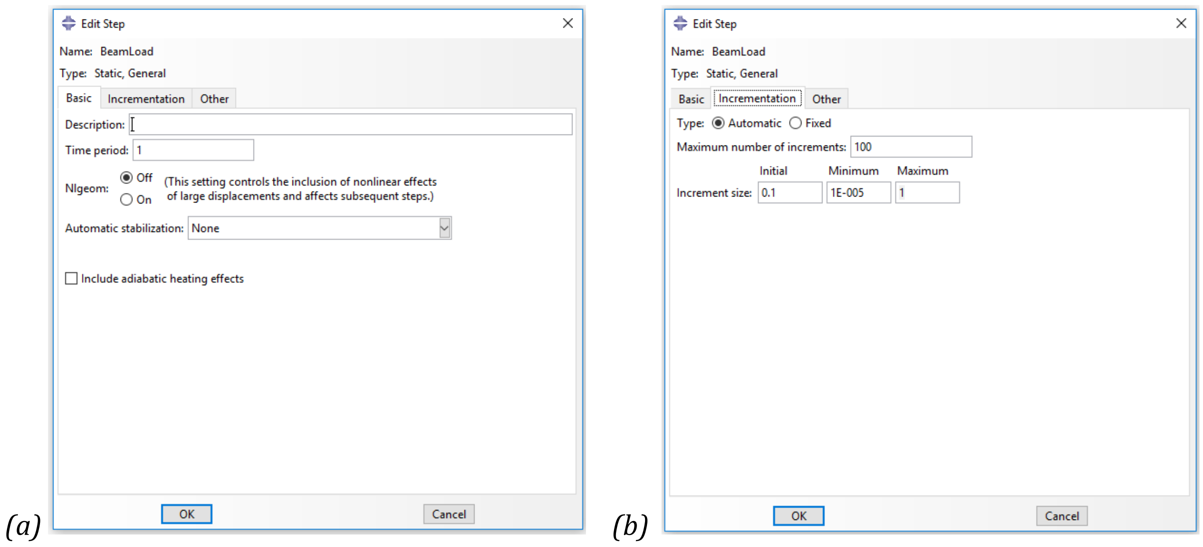
   The **Edit Step** dialog box appears. See Figure 1-17.
3. In the **Edit Step** dialog box:
   a. In the **Time period** field of the **Basic** tab page, enter **1**. Let **Nlgeom** (Non-linear geometry) be turned off. See Figure 1-17(a).
   b. Click the **Incrementation** tab, and delete the value of **1** that appears in the **Initial** text field. Type a value of **0.1** for the initial increment size. See Figure 1-17(b).
   c. You can accept the default values provided in the **Other** tab.

d. Click **OK** to create the step and to exit the step editor.



Create Step
Name: BeamLoad
Insert new step after: Initial
Procedure type: General
Geostatic
Heat transfer
Mass diffusion
Soils
Static, General
Static, Riks
Visco

Figure 1-16 – (a) The Create Step dialog box and (b) the Model Tree.



(a) Edit Step
Name: BeamLoad
Type: Static, General
Basic | Incrementation | Other
Description:
Time period: 1
Nlgeom: ◉ Off ○ On (This setting controls the inclusion of nonlinear effects of large displacements and affects subsequent steps.)
Automatic stabilization: None
☐ Include adiabatic heating effects

(b) Edit Step
Name: BeamLoad
Type: Static, General
Basic | Incrementation | Other
Type: ◉ Automatic ○ Fixed
Maximum number of increments: 100

	Initial	Minimum	Maximum
Increment size:	0.1	1E-005	1

Figure 1-17 – The Edit Step dialog box. (a) Basic tab and (b) increment tab.

Output Requests

The **Field Output Requests** and the **History Output Requests** in the model tree defines which data (i.e. stress, strain, displacement, temperature, etc.) that should be included in our result file. Abaqus/CAE automatically creates two output requests, called **F-Output-1** and **H-Output-1,** once a step is defined. These two requests contain the most basic information. They may be modified, but we will keep them as they are. In general:

- **Field Output Requests** – The output-data defined here will be included for every point of your model. This allows us to output the data as a contour plot for instance. However, for very large models with many time steps, a field output may become large, resulting in a large output file. In addition, writing the data to the output file can be time-consuming. Thus, only essential data should be included as a field output in large models.

- **History Output Requests –** If we want to output data for only a few points, say the displacement at the end of the model, a history output is normally used. A history output request consumes much less space.

1.5. Applying Load and Boundary Condition to the Model

Next, you will define the boundary condition and loading that will be active during the **BeamLoad** step.

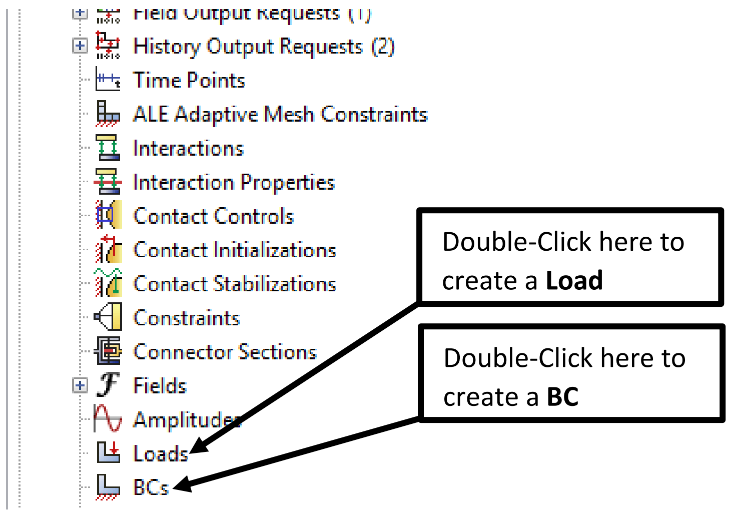


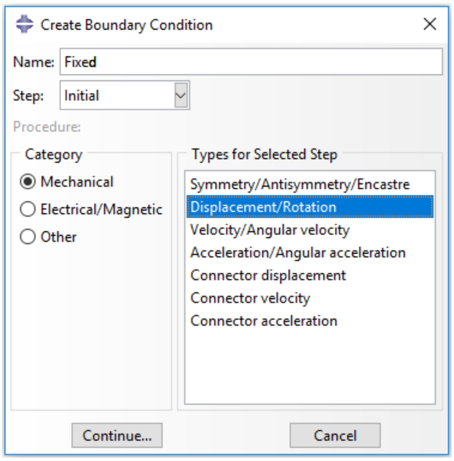
Figure 1-18 – The Model Tree.

Apply Boundary Condition to One End of the Cantilever Beam

1. In the **Model Tree**, double-click **BCs** to create a new boundary condition in the model **BEAM**. Abaqus/CAE switches to the Load module, and the **Create Boundary Condition** dialog box appears, see Figure 1-19(a).
2. In the **Create Boundary Condition** dialog box:
   a. Name the boundary condition **Fixed**.
   b. Select **Initial** as the step in which the boundary condition will be activated.
   c. In the **Category** list, accept the default category selection **Mechanical**.
   d. In the **Types for Selected Step** list, select **Displacement/Rotation** as the type.
   e. Click **Continue**.

   The **Region Selection** dialog box will appear, Figure 1-19(b). If not, click on **Sets** (Sets...) in the prompt area to access this dialog box.
3. Select the set **Beam-1.Fixed** in the **Region Selection** dialog box. This set is the same as the set created in Section 1.1. Click **Continue**.
   The **Edit Boundary Condition** dialog box appears, as shown in Figure 1-20.
4. In the **Edit Boundary Condition** dialog box:
   a. Toggle on **U1**, **U2**, and **U3**, since only the translational degrees of freedom need to be constrained (the beam will be meshed with solid elements later, which has no rotational degrees of freedom at its nodes.). This will fix the displacement components at the end of the beam.
   b. Click **OK** to create the boundary condition definition and to exit the editor.

Abaqus/CAE displays arrows at each corner and midpoint on the selected face to indicate the constrained degrees of freedom.

(a) 

Create Boundary Condition

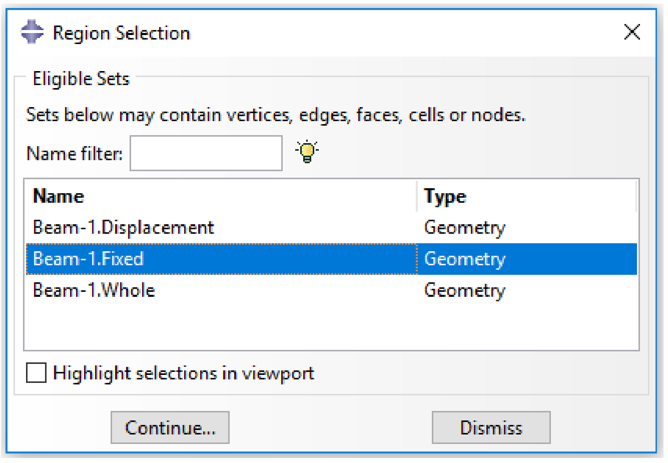
Name: Fixed

Step: Initial

Procedure:

Category: [✓ Mechanical] Electrical/Magnetic Other

Types for Selected Step: Symmetry/Antisymmetry/Encastre, [✓ Displacement/Rotation], Velocity/Angular velocity, Acceleration/Angular acceleration, Connector displacement, Connector velocity, Connector acceleration

(b) 

Region Selection

Eligible Sets

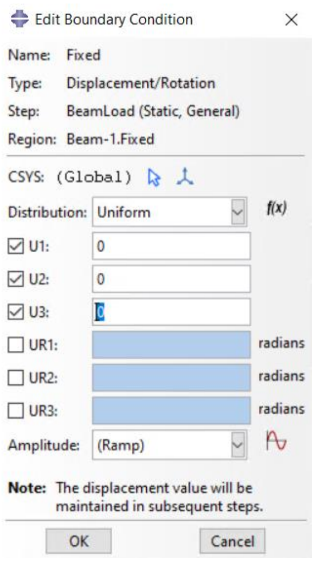
Sets below may contain vertices, edges, faces, cells or nodes.

Name filter:

Name	Type
Beam-1.Displacement	Geometry
Beam-1.Fixed	Geometry
Beam-1.Whole	Geometry

Highlight selections in viewport

Figure 1-19 – (a) The Create Boundary Condition dialog box and (b) the Region Selection dialog box.



Edit Boundary Condition

Name: Fixed

Type: Displacement/Rotation

Step: BeamLoad (Static, General)

Region: Beam-1.Fixed

CSYS: (Global)

Distribution: Uniform

[✓] U1: 0

[✓] U2: 0

[✓] U3: 0

UR1: radians

UR2: radians

UR3: radians

Amplitude: (Ramp)

Note: The displacement value will be maintained in subsequent steps.

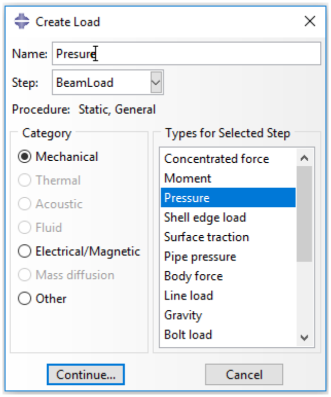
Figure 1-20 – The Edit Boundary Condition dialog box.

Apply a Load to the Top of the Cantilever Beam

1. In the **Model Tree**, double-click **Loads** to create a new load in the model **BEAM**. The **Create Load** dialog box appears, Figure 1-21(a).
2. In the **Create Load** dialog box:
    a. Name the load **Pressure**.
    b. Select **BeamLoad** as the step in which the load will be applied.
    c. In the **Category** list, accept the default category selection **Mechanical**.
    d. In the **Types for Selected Step** list, select **Pressure**.
    e. Click **Continue**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
3. In the viewport, select the top face of the beam as the surface to which the load will be applied. The desired face is highlighted in Figure 1-21(b). Check the **Create Surface** and

name it **BeamLoad**. Click **Done** in the prompt area to indicate that you have finished selecting regions.

(a)

Create Load

Name: Presure

Step: BeamLoad

Procedure: Static, General

Category: (● Mechanical) Thermal, Acoustic, Fluid, Electrical/Magnetic, Mass diffusion, Other

Types for Selected Step: Concentrated force, Moment, [✓ Pressure], Shell edge load, Surface traction, Pipe pressure, Body force, Line load, Gravity, Bolt load

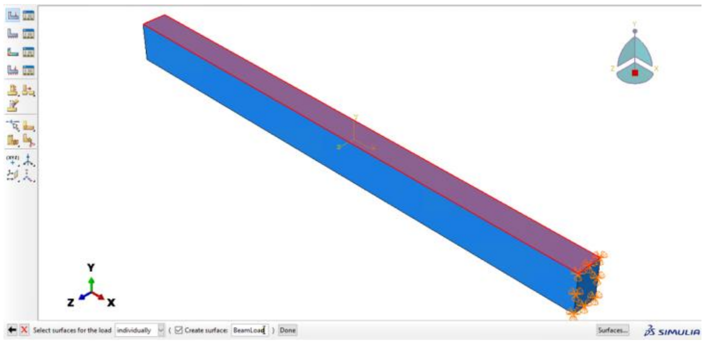
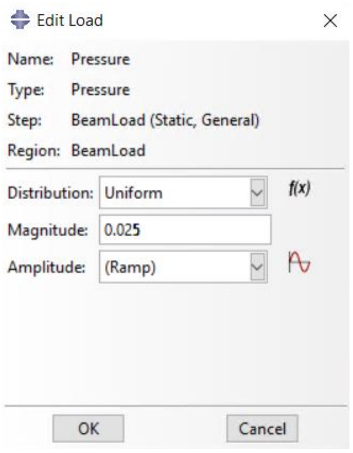
(b)

Select surfaces for the load: individually (☑ Create surface: BeamLoad) Done

Figure 1-21 – (a) The Create Load dialog box and (b) the prompt area.

The **Edit Load** dialog box appears, Figure 1-22.

4. In the **Edit Load** dialog box:
   a. Enter a magnitude of **0.025** for the load.
   b. Accept the default **Amplitude** selection (**Ramp**) and the default **Distribution** (**Uniform**).
   c. Click **OK** to create the load definition and to exit the editor.


Edit Load

Name: Pressure

Type: Pressure

Step: BeamLoad (Static, General)

Region: BeamLoad

Distribution: Uniform

Magnitude: 0.025

Amplitude: (Ramp)

Figure 1-22 – The Edit Load dialog box.

1.6. Meshing the Model

You use the Mesh module to generate the finite element mesh. You can choose the meshing technique that Abaqus/CAE will use to create the mesh, the element shape, and the element type. Abaqus/CAE uses a number of different meshing techniques. The default meshing technique assigned to the model is indicated by the color of the model when you enter the Mesh module; if Abaqus/CAE displays the model in orange, it cannot be meshed without assistance from the user.

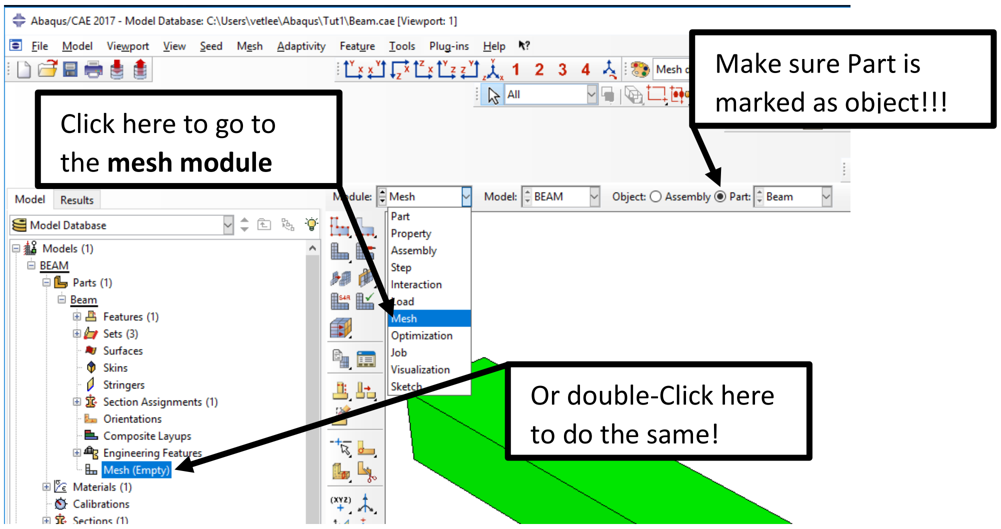
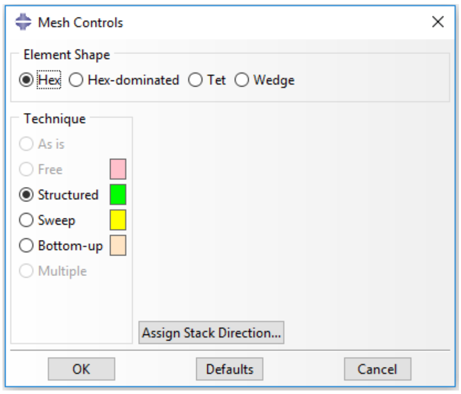


Figure 1-23 – How to access the meshing environment.

Assign the Mesh Controls:

1. In the Module drop-down menu, choose **Mesh** to access the meshing environment. You may alternately double-click on **Mesh** in the branch for the part **Beam** in the **Model Tree**. See Figure 1-23.
   Abaqus/CAE switches to the Mesh module and displays the part **Beam**. **Make sure Part is selected as Object!**
2. From the tools on the left ribbon, choose the **Mesh Controls**. See (1) in Figure 1-24(b).
3. In the **Mesh Controls** dialog box, accept **Hex** as the default **Element Shape** selection.
4. Accept **Structured** as the default **Technique** selection.
5. Click **OK** to assign the mesh controls and to close the dialog box.

(a)


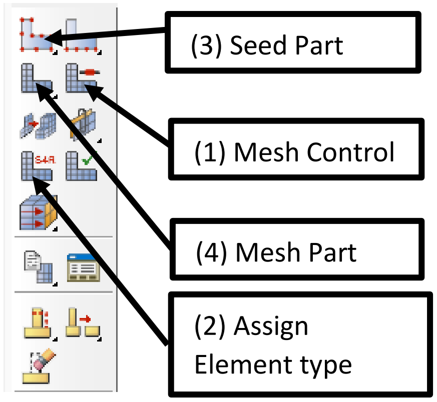
(b)


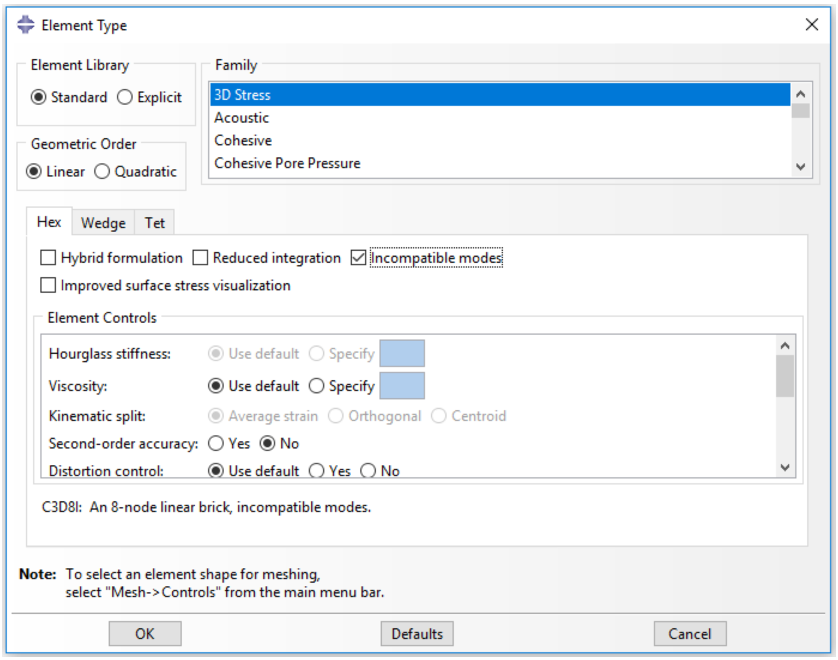
Figure 1-24 – (a) The Mesh Control dialog box and (b) the different tools for meshing a part in this workshop.

Assign an Abaqus Element Type:

1. From the tools in the left ribbon, choose the **Assign Element type**. See (2) in Figure 1-24(b).
2. In the **Element Type** dialog box, as shown in Figure 1-25. Accept the following default selections that control the elements that are available for selection:
   - **Standard** is the default **Element Library** selection.

- **Linear** is the default **Geometric Order**.
- **3D Stress** is the default **Family** of elements.

3. In the lower portion of the dialog box, examine the element shape options. A brief description of the default element selection is available at the bottom of each tabbed page.
4. In the **Hex** tabbed page, select **Reduced integration**. No modification is necessary in the **Wedge** and **Tet** tab.
   A description of the element type C3D8R appears at the bottom of the dialog box. Abaqus/CAE will now mesh the part with C3D8R elements.
5. Click **OK** to assign the element type and to close the dialog box.


Element Type

Element Library: [✓ Standard] Explicit

Geometric Order: [✓ Linear] Quadratic

Family: [✓ 3D Stress], Acoustic, Cohesive, Cohesive Pore Pressure

Hex | Wedge | Tet

☐ Hybrid formulation ☐ Reduced integration ☑ Incompatible modes
☐ Improved surface stress visualization

Element Controls

Hourglass stiffness:	[✓ Use default] Specify
Viscosity:	[✓ Use default] Specify
Kinematic split:	[✓ Average strain] Orthogonal Centroid
Second-order accuracy:	Yes [✓ No]
Distortion control:	[✓ Use default] Yes No

C3D8I: An 8-node linear brick, incompatible modes.

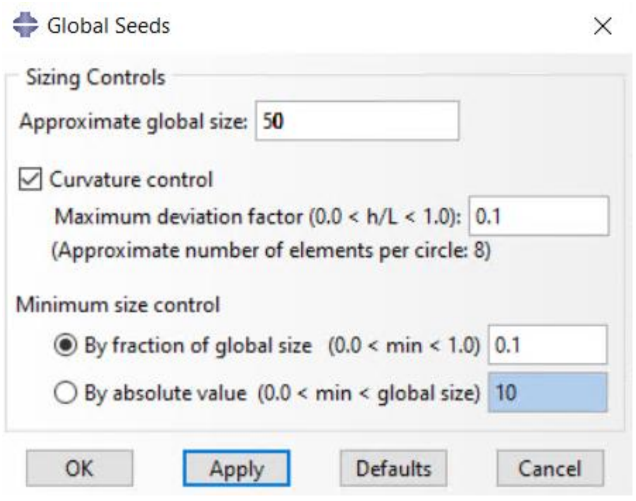
Note: To select an element shape for meshing, select "Mesh->Controls" from the main menu bar.

OK Defaults Cancel

Figure 1-25 – The Element Type dialog box.

Mesh the Model:

1. From the tools in the left ribbon, choose the **Seed Part**. See (3) in Figure 1-24(b).
2. The **Global Seeds** dialog box appears, Figure 1-26. The default global element size is based on the size of the part.
3. In the **Global Seeds** dialog box, enter an approximate global size of **50** (mm) and click **OK**. Abaqus/CAE applies the seeds to the part, as shown in Figure 1-27(a).
4. From the tools in the left ribbon, choose the **Mesh Part**. See (4) in Figure 1-24(b).
5. Click **Yes** in the prompt area to confirm that you want to mesh the part instance. Abaqus/CAE meshes the part instance and displays the resulting mesh, as shown in Figure 1-27(b).



Global Seeds

Sizing Controls

Approximate global size: 50

☑ Curvature control

Maximum deviation factor (0.0 < h/L < 1.0): 0.1

(Approximate number of elements per circle: 8)

Minimum size control

● By fraction of global size (0.0 < min < 1.0) 0.1

○ By absolute value (0.0 < min < global size) 10

Figure 1-26 – The Global Seeds dialog box.

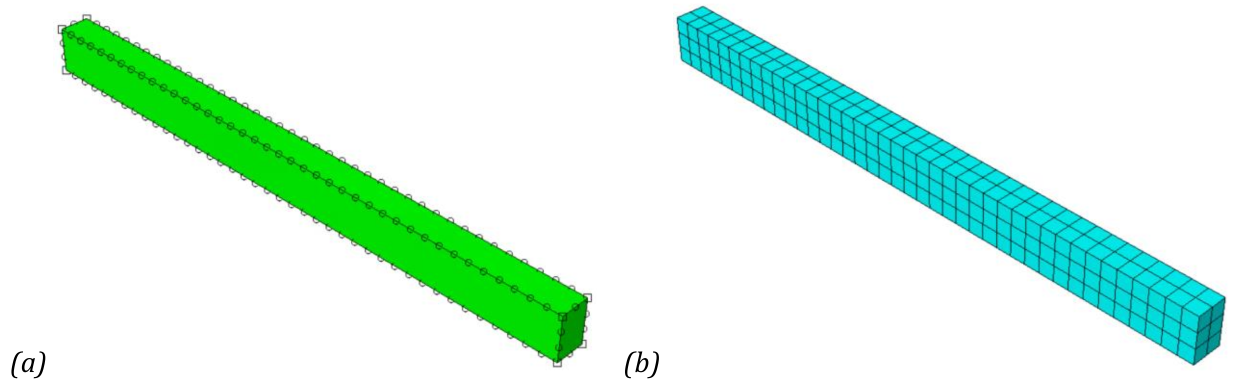


Figure 1-27 – (a) The Seeded part instance and (b) part instance showing the resulting mesh.

1.7. Creating and Submitting an Analysis

The definition of the model **BEAM** is now complete. Next, you will create and submit an analysis job to analyze the model. Before submitting a job, **ALWAYS check if the work directory is set**.

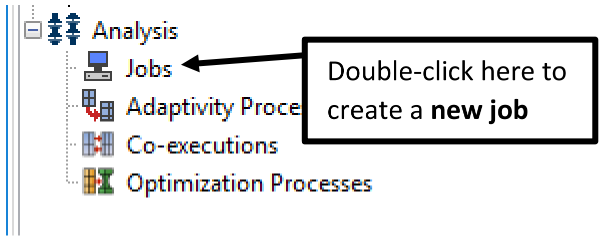
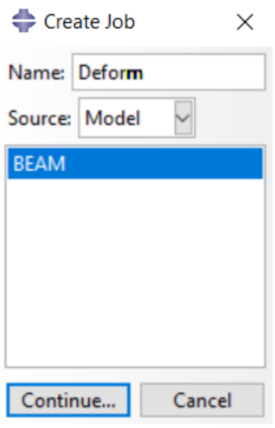


Figure 1-28 – The analysis part of the Model Tree

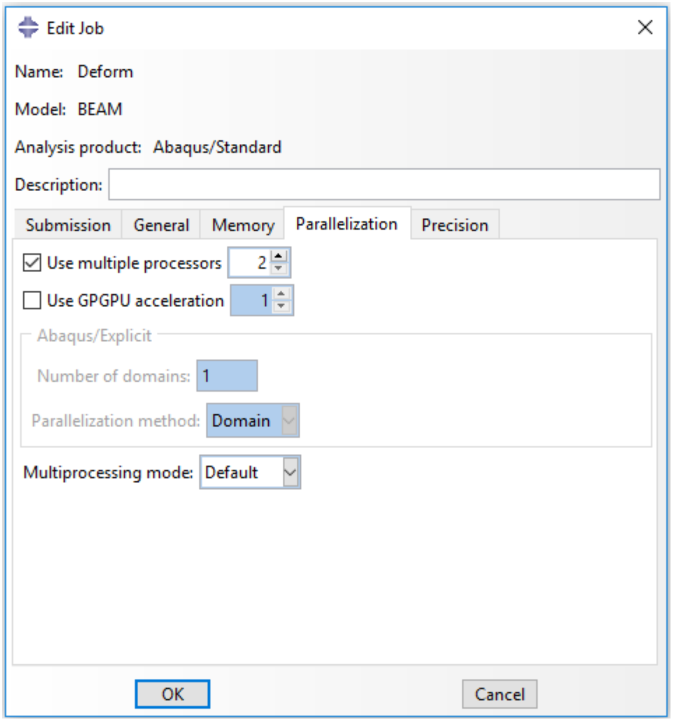
Create and Submit an Analysis Job

1. In the Model Tree, double-click **Jobs** to create a new analysis job.
   Abaqus/CAE switches to the Job module, and the **Create Job** dialog box appears, see Figure 1-29(a).

2. In the **Create Job** dialog box, name the job **Deform** and select the model **BEAM**. Click **Continue**.
   The **Edit Job** dialog box appears.
3. In the **Parallelization** tab, Figure 1-29(b), check the **Use multiple processors** in order to use more than one CPU, allowing faster simulations. Click **OK** to accept the default job settings.

(a) 

Create Job
Name: Deform
Source: Model
BEAM

(b) 

Edit Job
Name: Deform
Model: BEAM
Analysis product: Abaqus/Standard
Description:

☑ Use multiple processors 2
☐ Use GPGPU acceleration 1
Abaqus/Explicit
Number of domains: 1
Parallelization method: Domain
Multiprocessing mode: Default

Figure 1-29 – (a) The Create Job dialog box and (b) the Edit Job dialog box.

4. In the **Model Tree**, expand the **Jobs** container and right-click on the job **Deform**.
   The menu for the job **Deform** appears, as shown in Figure 1-30(a).
5. From the job menu, select **Submit**. Abaqus/CAE will now send the job to a solver, which performs the analysis.
   The icon for the job will change to indicate the status of the job in parenthesis after the job name. As the job runs the status **Running** will be shown in the Model Tree. When the job completes successfully, the status will change to **Completed**, as shown in Figure 1-30(b). Notice that Abaqus has generated many new files to your work directory. The **.odb** file contains the results from your analysis.

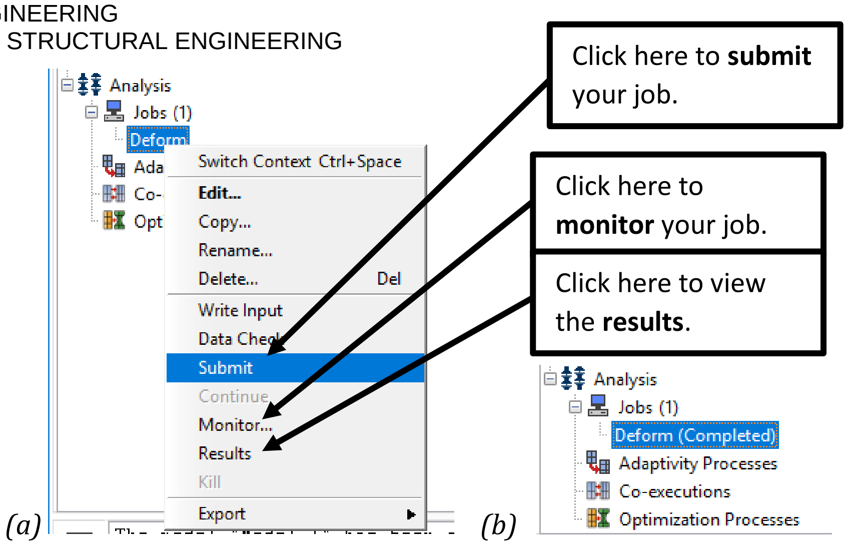
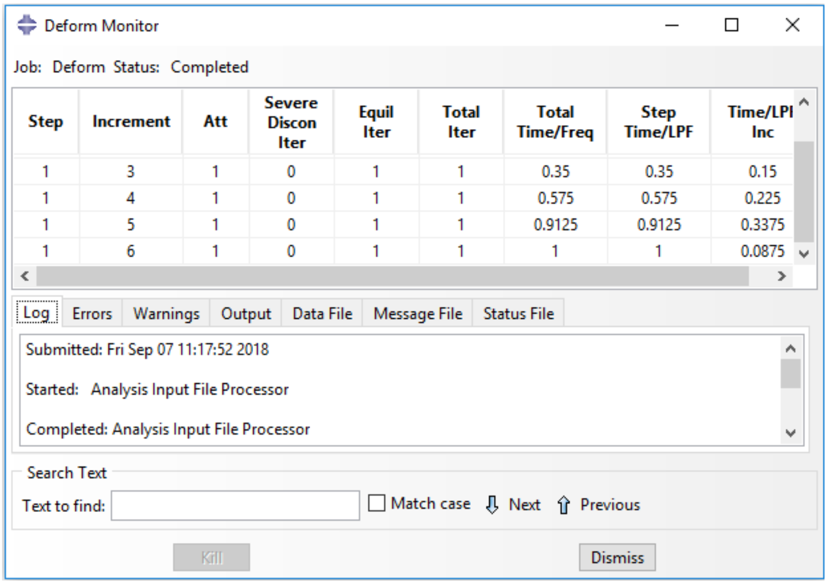


Figure 1-30 – (a) The menu for the Deform job and (b) status when the analysis has completed.

6. In the menu for the job **Deform**, Figure 1-30(a), click on **Monitor** to access the **Monitor** dialog box. Here the analysis status may be checked for **Errors** and **Warnings**. **This should be checked for every analysis**. The same information can be found in the **.msg** and **.sta** files in your working directory.



Deform Monitor

Job: Deform Status: Completed

Step	Increment	Att	Severe Discon Iter	Equil Iter	Total Iter	Total Time/Freq	Step Time/LPF	Time/LPF Inc
1	3	1	0	1	1	0.35	0.35	0.15
1	4	1	0	1	1	0.575	0.575	0.225
1	5	1	0	1	1	0.9125	0.9125	0.3375
1	6	1	0	1	1	1	1	0.0875

```
Submitted: Fri Sep 07 11:17:52 2018

Started:   Analysis Input File Processor

Completed: Analysis Input File Processor
```

Figure 1-31 – The Monitor dialog box.

1.8. Post-processing the Results

Open a Result (.odb) File

Abaqus stores the requested output data in an ODB-file (ODB = Output Database). This file is saved in your work directory. You can open this file two ways

1. Right-click on your job-file in the **Model Tree** and choose **Results**. See Figure 1-30.
2. In the main menu bar, select **File** → **Open**. The **Open Database** dialog box appears. Make sure to set the **File Filter** to **Output Database (*.odb)**.

Abaqus/CAE switches to the Visualization module, opens the output database created by the job (**Deform.odb**), and displays the undeformed shape of the model, as shown in Figure 1-32.

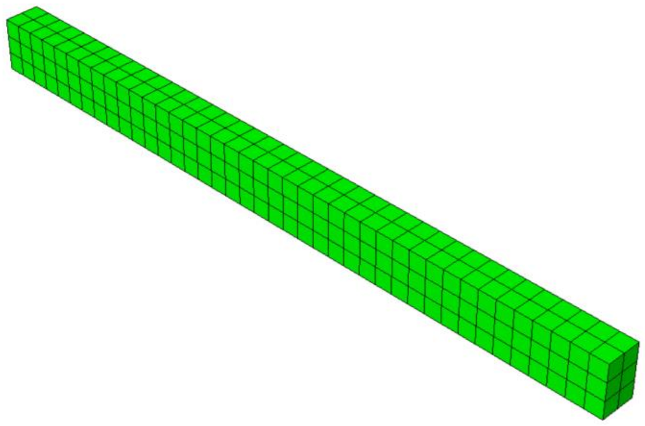


Figure 1-32 – Un-deformed model shape.

Plotting Deformation and Contour Plots

The data saved in the **Field Output Requests** can be viewed as a deformation and contour plot in the viewport.

1. In the toolbox, click (or select **Plot** → **Deformed Shape** from the main menu bar) to view a deformed model shape plot, as shown in Figure 1-33.
   In order to plot the un-deformed beam in the same plot, go to **Plot** → **Allow Multiple Plot State**.

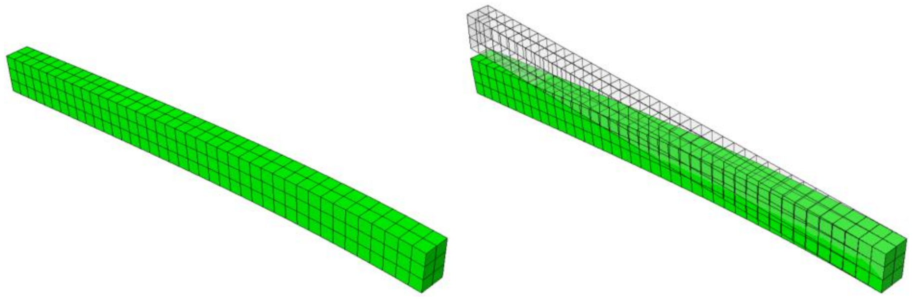


Figure 1-33 – Deformed model shape.

   You may need to use the **Auto-Fit View** tool to rescale the figure in the viewport.

2. In the toolbox, click (or select **Plot** → **Contours** → **on Deformed Shape** from the main menu bar) to view a contour plot of the von Mises stress, as shown in Figure 1-34.

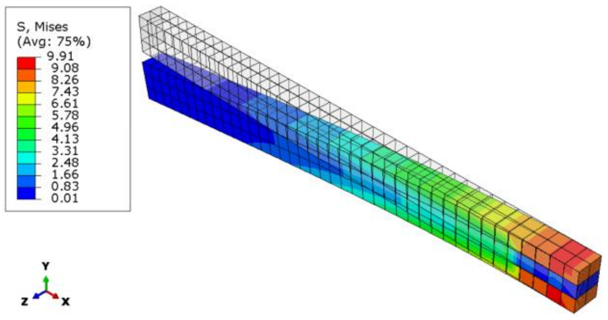


Figure 1-34 – von Mises contour plot.

3. You can change the contour plot from the **Field Output** menu, as shown in Figure 1-35. You may choose from the output data defined in the **History Output Request**, see Section 1.4.

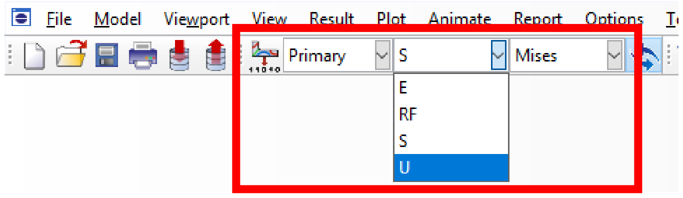
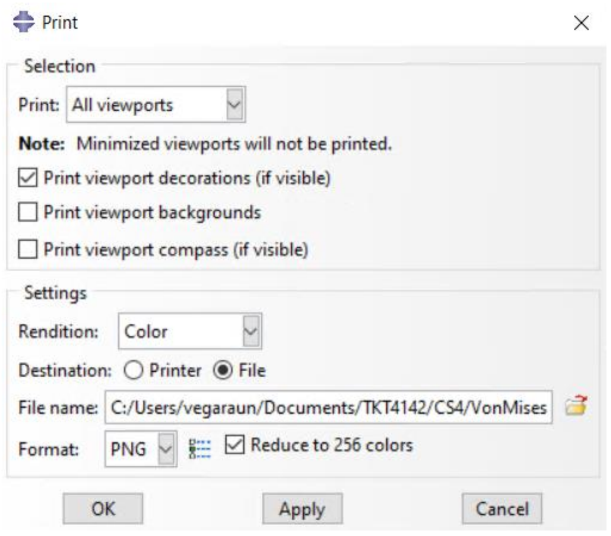


Figure 1-35 – The Field Output menu.

4. You can change the color in the background of the viewport by selecting **View** → **Graphics Options** from the main menu bar. Here, change the **Viewport Background** option.
5. The annotations can be removed by selecting **Viewport** → **Viewport Annotation Options** from the main menu bar. Uncheck **Show compass** and **show triad**. Select **OK**.
6. Export a picture of the deformation plot from **File** → **Print** in the main menu bar**.** Change the **Destination** to **File** in the **Print** dialog box and set your destination. Use PNG as format. See Figure 1-36.



Print

Selection

Print: All viewports

Note: Minimized viewports will not be printed.

☑ Print viewport decorations (if visible)

☐ Print viewport backgrounds

☐ Print viewport compass (if visible)

Settings

Rendition: Color

Destination: Printer [✓ File]

File name: C:/Users/vegaraun/Documents/TKT4142/CS4/VonMises

Format: PNG ☑ Reduce to 256 colors

OK Apply Cancel

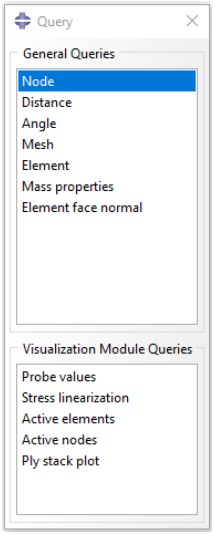
Figure 1-36 –The Print dialog box

Obtain Numerical Values from the Field Output

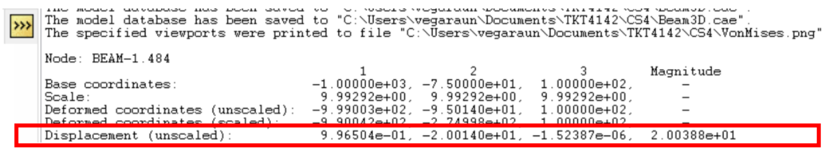
You can obtain numerical values from the Field Output. The simplest way is to use the **Query information** tool.

1. In the upper toolbox, select use the **Query information** tool . The **Query** dialog box opens.
2. In the **Query** dialog box, use **Node** as the **General Queries**. Input the node in the prompt area or select a node in the viewport. The numerical value for the displacement is displayed in the **Kernel Command Line Interface**, see Figure 1-37(b).

(a)



(b)

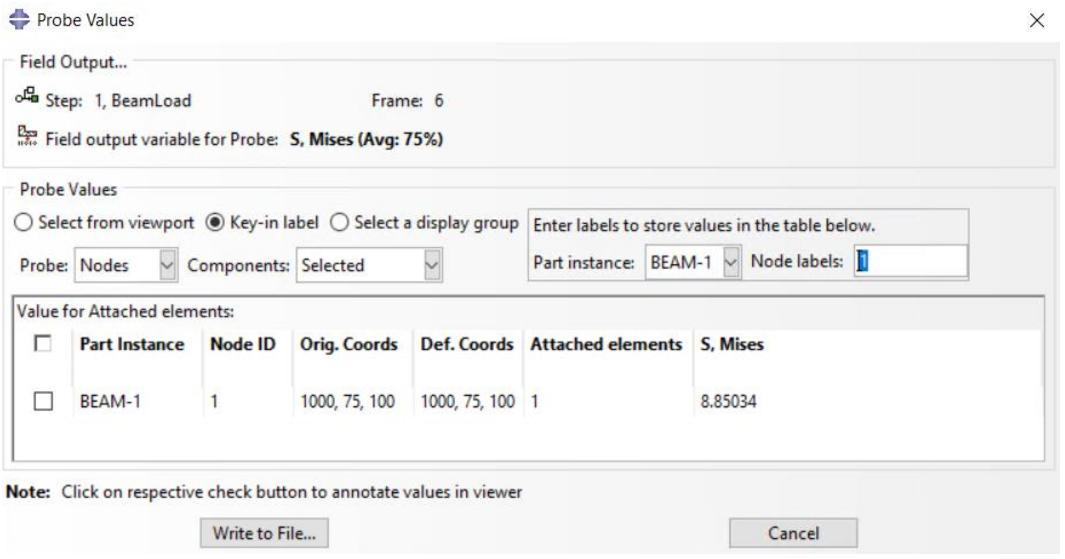


```
The model database has been saved to "C:\Users\vegaraun\Documents\TKT4142\CS4\Beam3D.cae".
The specified viewports were printed to file "C:\Users\vegaraun\Documents\TKT4142\CS4\VonMises.png"

Node: BEAM-1.484
                                        1              2              3          Magnitude
Base coordinates:                -1.00000e+03,  -7.50000e+01,   1.00000e+02,       -
Scale:                            9.99292e+00,   9.99292e+00,   9.99292e+00,       -
Deformed coordinates (unscaled): -9.99003e+02,  -9.50140e+01,   1.00000e+02,       -
Deformed coordinates (scaled):   -9.90042e+02,  -2.74998e+02,   1.00000e+02,       -
Displacement (unscaled):          9.96504e-01,  -2.00140e+01,  -1.52387e-06,   2.00388e+01
```

Figure 1-37 – (a) The Query dialog box and (b) the Kernel Command Line Interface.

You can also use the **Probe values** tool in **Query** dialog box. This opens the **Probe Values** dialog box. Select the last **Frame** and **Field output variable for Probe**. In the **Probe Values**, select **Key-in label** and set the **Probe** to **Nodes**. See Figure Figure 1-38. Input the **Node labels**.



Step: 1, BeamLoad   Frame: 6
Field output variable for Probe: S, Mises (Avg: 75%)

( ) Select from viewport [✓ Key-in label] ( ) Select a display group
Enter labels to store values in the table below.
Probe: Nodes   Components: Selected
Part instance: BEAM-1   Node labels: 1

Value for Attached elements:

☐	Part Instance	Node ID	Orig. Coords	Def. Coords	Attached elements	S, Mises
☐	BEAM-1	1	1000, 75, 100	1000, 75, 100	1	8.85034

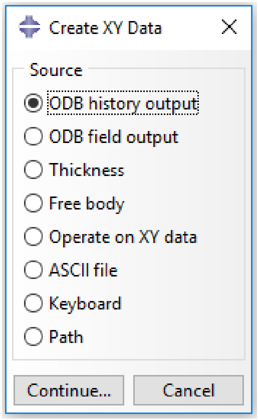
Note: Click on respective check button to annotate values in viewer

Figure 1-38 – The Probe Values dialog box.

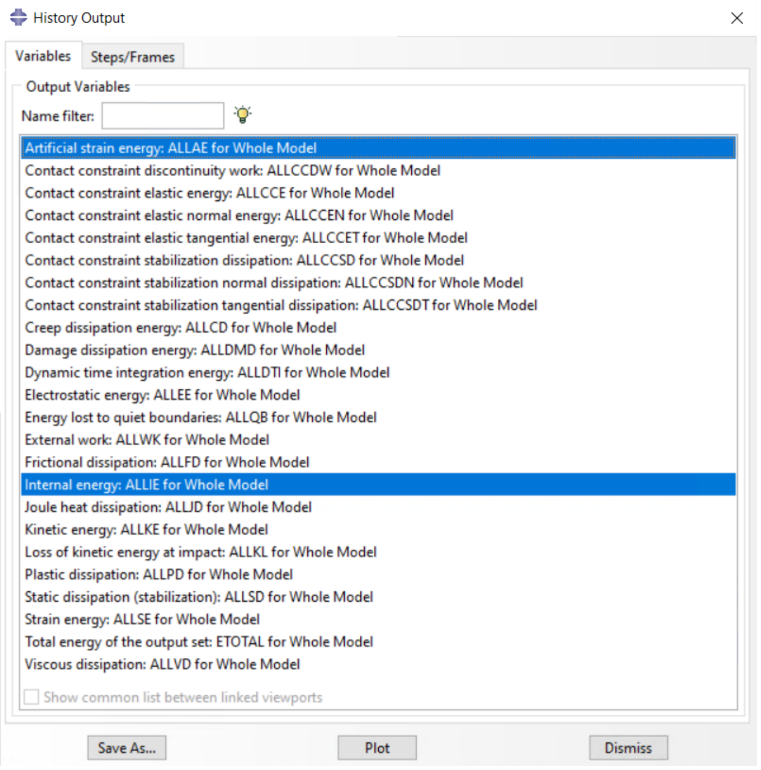
Obtain Energies From the History Output

We want to find the **internal energy** of the model. First, open your results.

1. Use the **Create XY Data** tool (or **Tools** → **XY Data** → **Create** in the main menu bar). This opens the **Create XY Data** dialog box, see Figure 2-39(a). As **Source**, use **ODB history output**. Select **Continue**.
2. In the **History Output** dialog box, click on both **Artificial strain energy: ALLAE for Whole Model** and **Internal energy: ALLIE for Whole Model.** See Figure 2-39(b). Select **Plot** to plot the data in the viewport.
3. You want to read the value at time 1. This can be done using the **Probe values** in the **Query** tool
   a. First, click on to access the **Query tool**.
   b. Then select **Probe values** from the **Query** dialog box.
   c. Read of the value for your last increment.

(a)

Create XY Data

Source
- [✓ ODB history output]
- ODB field output
- Thickness
- Free body
- Operate on XY data
- ASCII file
- Keyboard
- Path

(b)

History Output

Output Variables

Name filter:

- [✓ Artificial strain energy: ALLAE for Whole Model]
- Contact constraint discontinuity work: ALLCCDW for Whole Model
- Contact constraint elastic energy: ALLCCE for Whole Model
- Contact constraint elastic normal energy: ALLCCEN for Whole Model
- Contact constraint elastic tangential energy: ALLCCET for Whole Model
- Contact constraint stabilization dissipation: ALLCCSD for Whole Model
- Contact constraint stabilization normal dissipation: ALLCCSDN for Whole Model
- Contact constraint stabilization tangential dissipation: ALLCCSDT for Whole Model
- Creep dissipation energy: ALLCD for Whole Model
- Damage dissipation energy: ALLDMD for Whole Model
- Dynamic time integration energy: ALLDTI for Whole Model
- Electrostatic energy: ALLEE for Whole Model
- Energy lost to quiet boundaries: ALLQB for Whole Model
- External work: ALLWK for Whole Model
- Frictional dissipation: ALLFD for Whole Model
- [✓ Internal energy: ALLIE for Whole Model]
- Joule heat dissipation: ALLJD for Whole Model
- Kinetic energy: ALLKE for Whole Model
- Loss of kinetic energy at impact: ALLKL for Whole Model
- Plastic dissipation: ALLPD for Whole Model
- Static dissipation (stabilization): ALLSD for Whole Model
- Strain energy: ALLSE for Whole Model
- Total energy of the output set: ETOTAL for Whole Model
- Viscous dissipation: ALLVD for Whole Model

Show common list between linked viewports

Figure 2-39 – (a) The Create XY Data dialog box and (b) The History Output dialog box.

Abaqus uses hourglass control algorithms to prevent large deformation when reduced elements are used. In a simple linear analysis, these mechanisms are easy to observe from the deformation pattern. However, for large models and non-linear simulations, mechanisms are not as prominent in the deformation pattern.

One can check the occurrence, and the degree of occurrence, of mechanisms by comparing the Artificial Strain Energy (called ALLAE) to the Internal Strain Energy (ALLIE). As stated in the Abaqus documentation; "Artificial strain energy associated with constraints used to remove singular modes (such as hourglass control)…". Thus, to prevent large deformations, Abaqus constraints the element by introducing artificial stiffness. Hence, to monitor the energy stored in hourglass resistances one should evaluate the artificial strain energy (ALLAE). Large values of artificial strain energy indicate that mesh refinement or other changes to the mesh are necessary.

One way to evaluate the importance of artificial strain energy is by comparison to the internal energy (ALLIE) component. The ratio ALLAE/ALLIE can be used to help evaluate whether an analysis is yielding an appropriate response. The internal energy is the sum of the recoverable elastic strain energy and the artificial strain energy.

The ratio ALLAE/ALLIE should not exceed 2-3% in a simple linear analysis. Figure 2-40 shows the plot of these energies for the reduced quad elements in this workshop. Probing the values at time = 1.0 shows that ALLIE = 20000 and ALLAE = 45, resulting in a ratio of 0,23 %. The ALLAE is below the recommended threshold of 2-3 %. This should be checked every time you run an analysis using reduced elements.

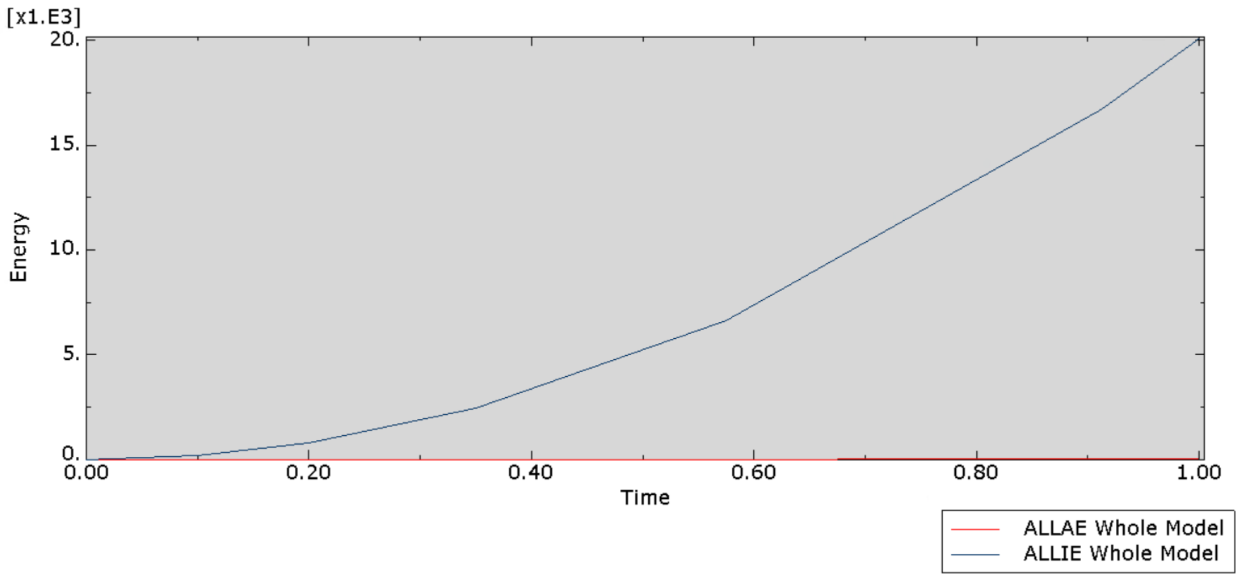


Figure 2-40 – Plot of the Artificial strain energy (ALLAE) and Total strain energy (ALLIE).

2. Modelling the Beam with Holes Using 3D Solid Elements

We will now increase the complexity of the model to also include the hole at the steel rod support. See Figure 3-1. The process of modeling the beam with a hole is rather similar as done in Section 1. This section will only highlight the main differences. The reader is referred to Section 1 for a more in-depth description of the modeling process.

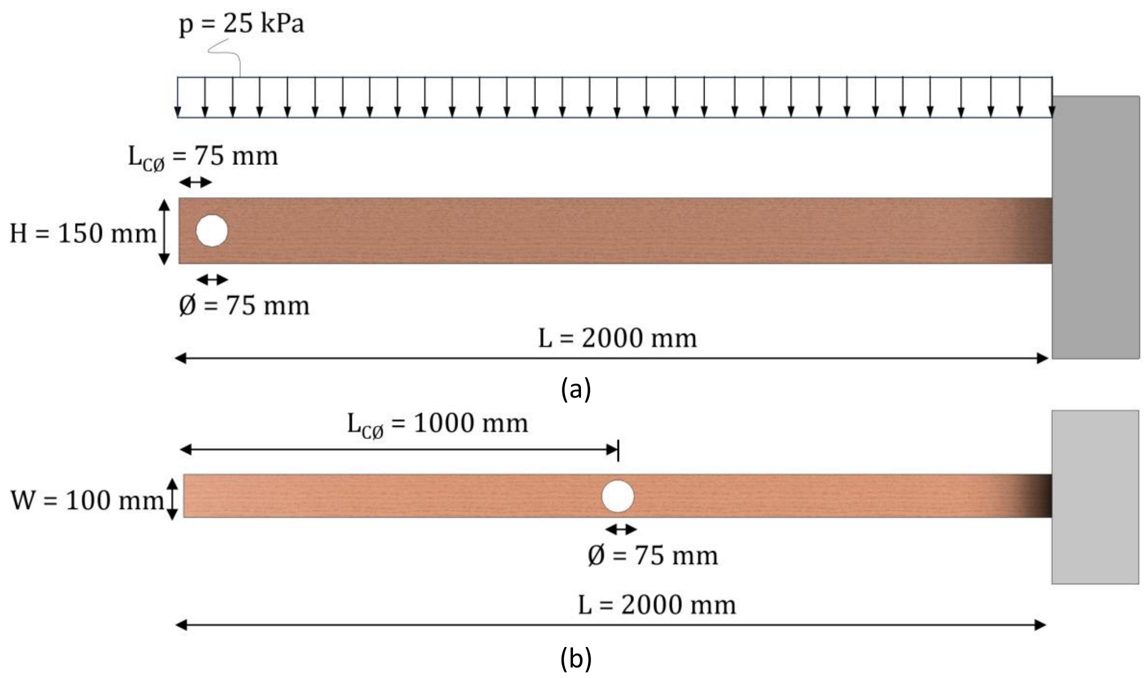
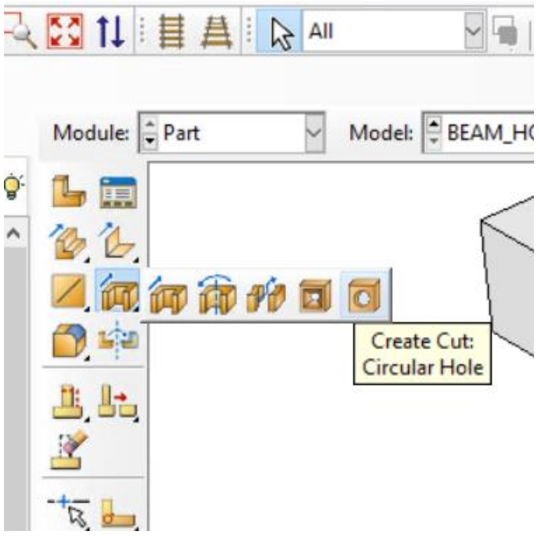
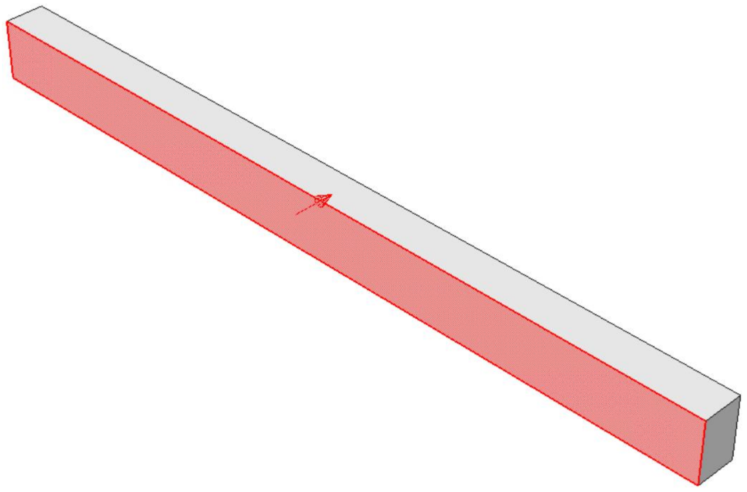


Figure 3-1 – (a) Side view of the beam including the hole at the end of the beam and (b) top view of the beam illustrating the hole at the center of the beam.

1. First, create a new model by right-click on the previous **BEAM** model in the **Model Tree**. Select **Copy Model** to create a new model. Name the model **BEAM_HOLES**. Click **OK**. A new model is added to work **Model Tree.**
2. Go to the **Part** module and find **Create Cut: Circular Hole.** Select plane for the hole. Click OK.

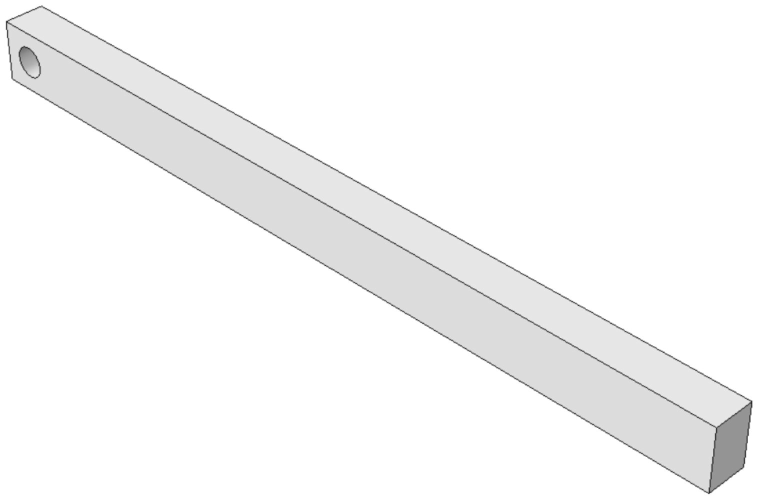




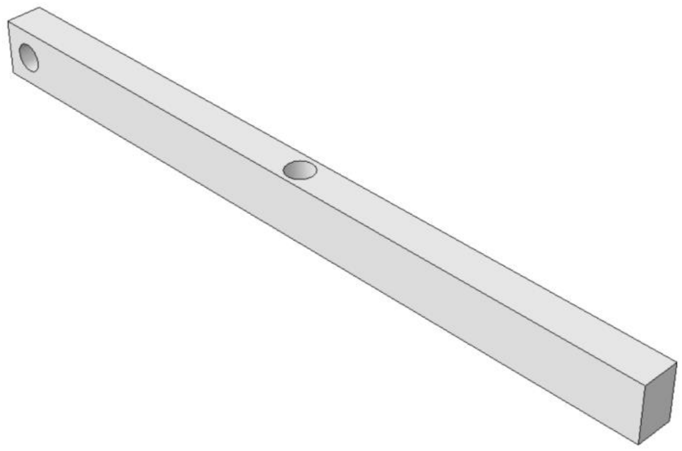
Select the left edge as the reference to locate the hole. Then **75** is the distance from this edge to the hole center.

Select the lower horizontal edge as the second edge as the second reference. Again, **75** is the distance from the lower horizontal edge to the center of the hole. Finally, type **75** for the diameter of the hole and press Enter.

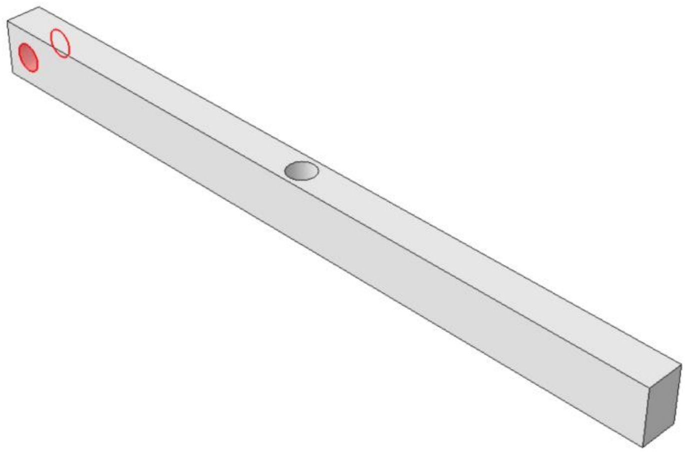
Voilla, we now have the first hole in the beam.



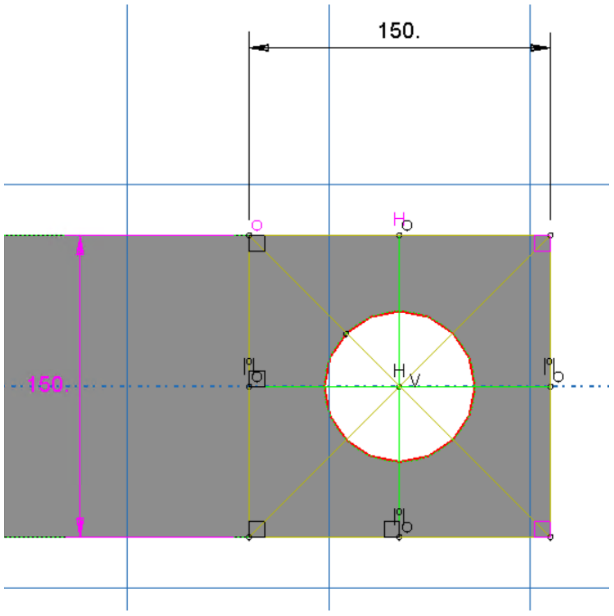
Repeat this process to generate the second hole with position and orientation as shown in Figure 3-1(b). The final geometry of the part will then look like this:



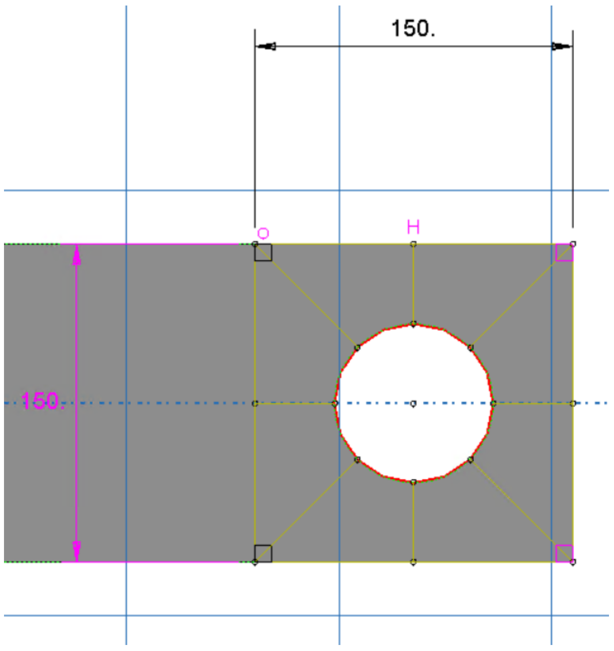
Partition face: Sketch ( ) on the hole around the steel rod to define boundary conditions similar to that in Case Study 1.



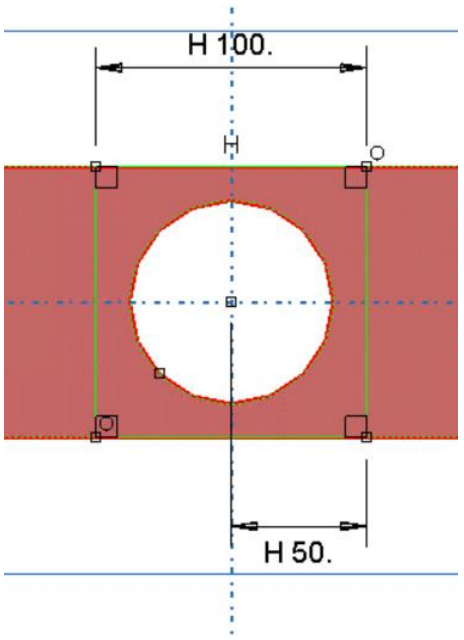
Sketch partition around the hole:



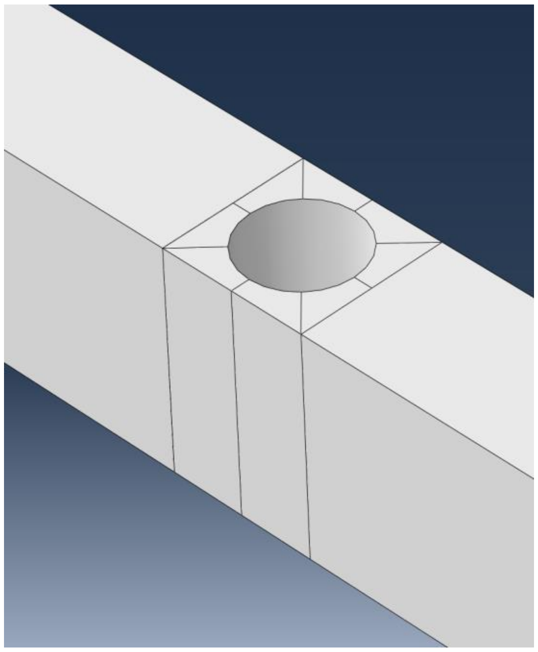
If you get an error message in Abaqus, try to delete the dimensions and/or trim the lines overlapping with the hole.



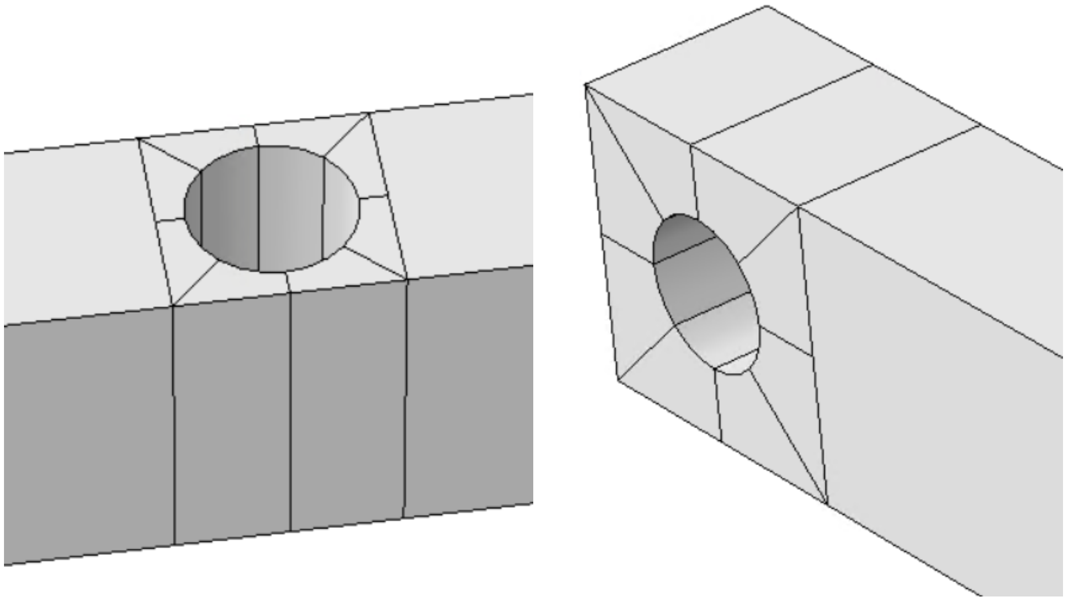
Similarly, partition the areas around the center hole to be able to conduct the meshing:



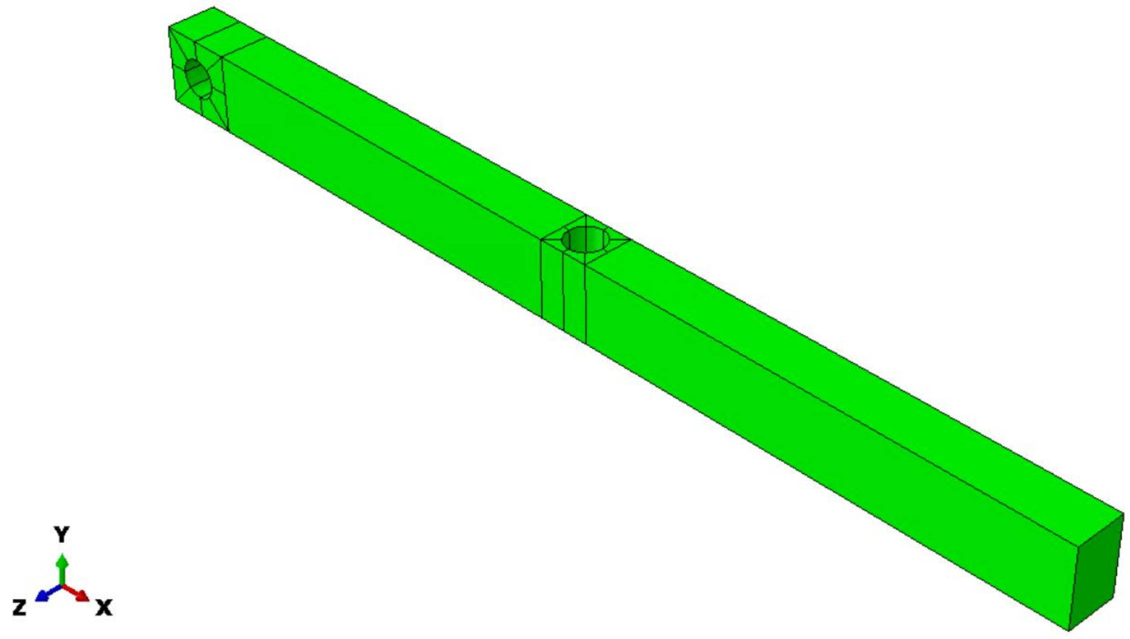
Now, we have defined the partitions on the surface. For the centre hole, this will look something like the figure above. Note that the partition lines do not go through the thickness.



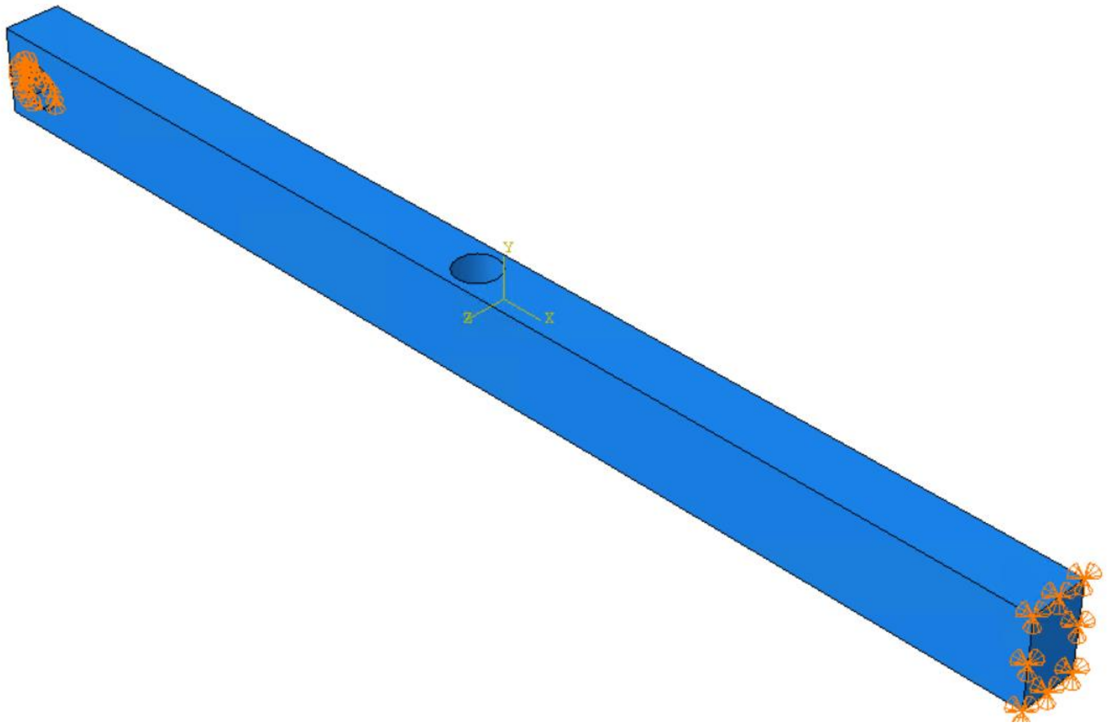
To make the partition go through the surface, we can use **Partition Cell: Define Cutting Plane** ( ). Click **Partition Cell: Define Cutting Plane** and select the region you want to partition. Click 3 Points and select three points that are on the surface you want to cut through the region. Click **Create Partition.** Do this for all the sections around the hole to create the partitions. Repeat for the other hole.

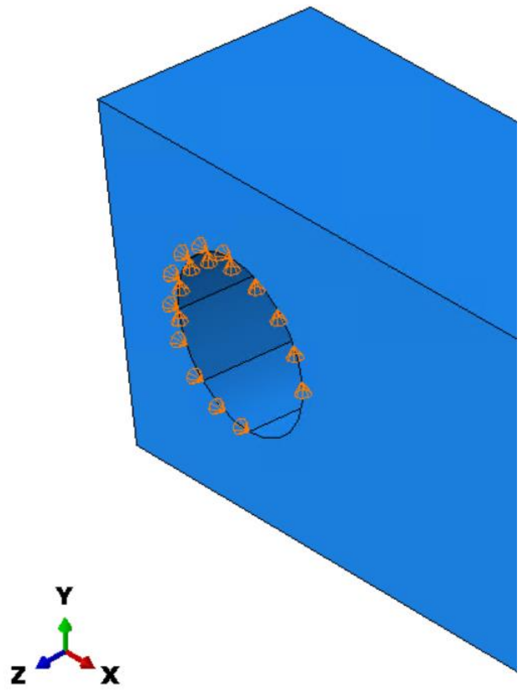


The model turns green in the mesh module when the partitioning of the geometry allows for a structured mesh:

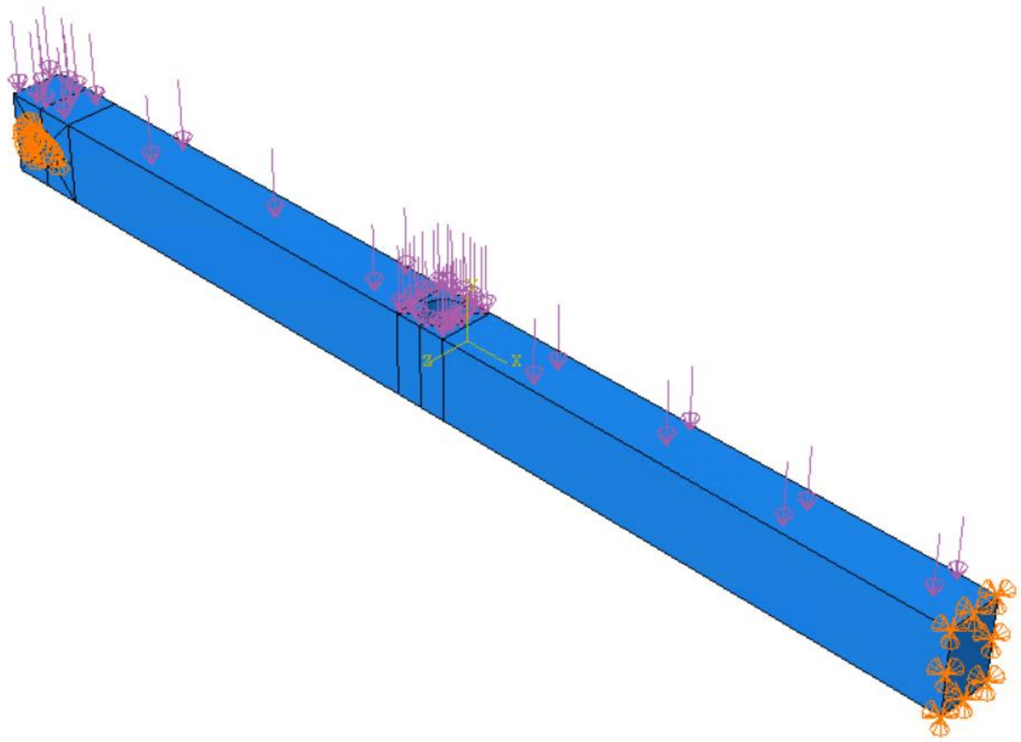


Assign boundary conditions mimicking the steel rod support in the load module:





In the load module, update the surface for the applied pressure:



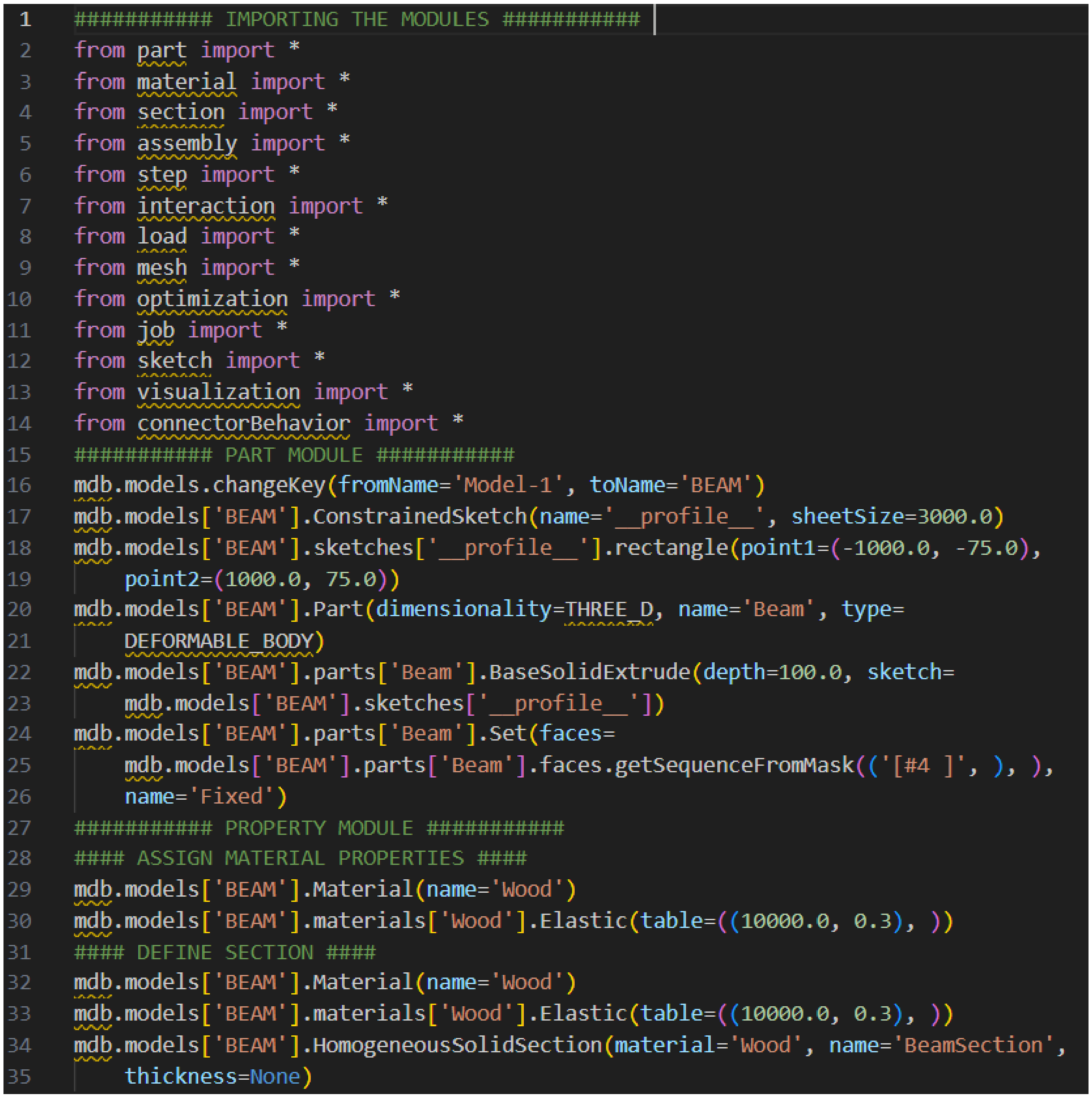
Mesh the model, create, and submit a job and post-process the results (see Sections 1.6, 1.7 and 1.8).

3. An introduction to the Abaqus Python scripting interface

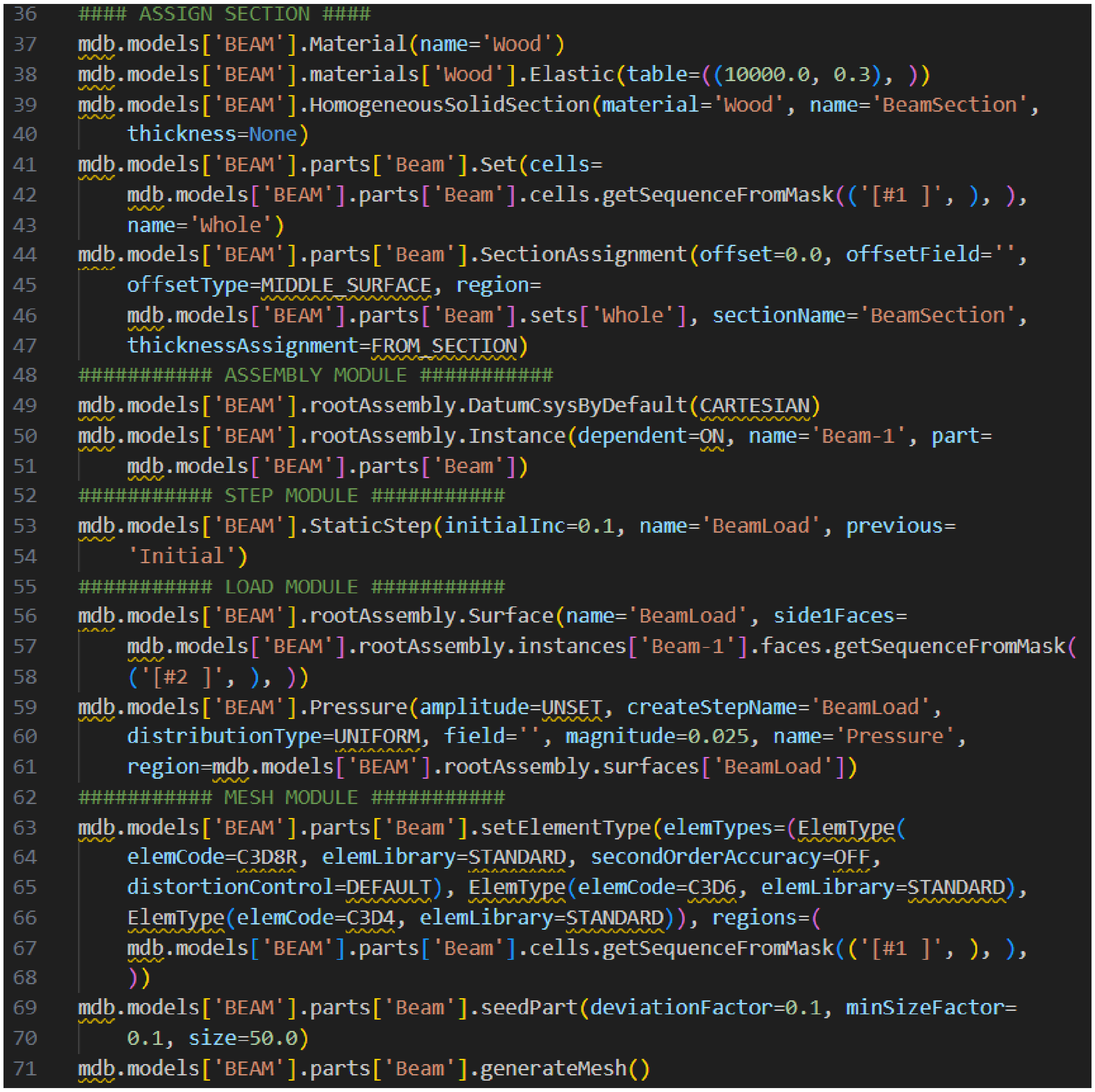
We will now give a brief introduction to the Python scripting interface in Abaqus by using the model for the cantilever beam created in Section 1.

Python is the standard programming language for Abaqus scripting and is used in several ways. We will pay special attention to the various modules required for modeling the cantilever beam. This is done by inspection of the journal (.jnl) file. This file contains the Python configuration commands used to build your model in the Abaqus/CAE GUI and is automatically generated by Abaqus in your Working directory.

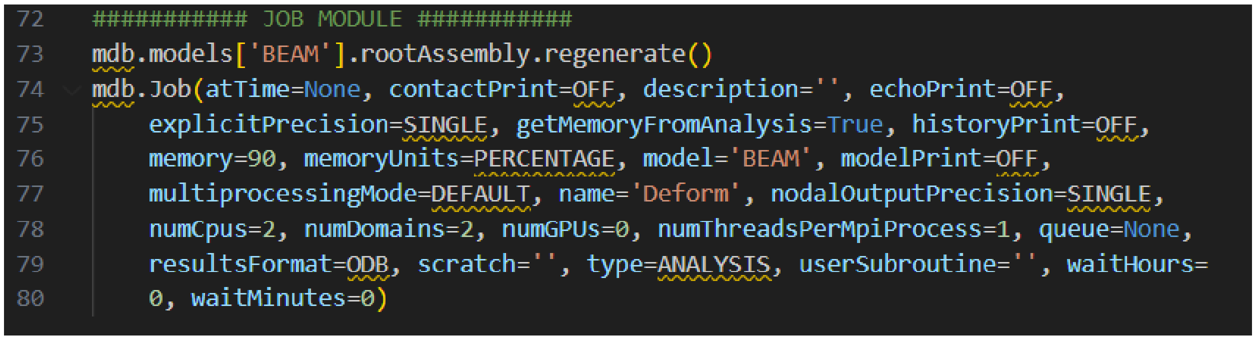
Try to open your journal (.jnl) file. This has the same name as your .cae file (e.g., beam.jnl). The journal file should look something like this:



```
########### IMPORTING THE MODULES ###########
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from optimization import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
########### PART MODULE ###########
mdb.models.changeKey(fromName='Model-1', toName='BEAM')
mdb.models['BEAM'].ConstrainedSketch(name='__profile__', sheetSize=3000.0)
mdb.models['BEAM'].sketches['__profile__'].rectangle(point1=(-1000.0, -75.0),
    point2=(1000.0, 75.0))
mdb.models['BEAM'].Part(dimensionality=THREE_D, name='Beam', type=
    DEFORMABLE_BODY)
mdb.models['BEAM'].parts['Beam'].BaseSolidExtrude(depth=100.0, sketch=
    mdb.models['BEAM'].sketches['__profile__'])
mdb.models['BEAM'].parts['Beam'].Set(faces=
    mdb.models['BEAM'].parts['Beam'].faces.getSequenceFromMask(('[#4 ]', ), ),
    name='Fixed')
########### PROPERTY MODULE ###########
#### ASSIGN MATERIAL PROPERTIES ####
mdb.models['BEAM'].Material(name='Wood')
mdb.models['BEAM'].materials['Wood'].Elastic(table=((10000.0, 0.3), ))
#### DEFINE SECTION ####
mdb.models['BEAM'].Material(name='Wood')
mdb.models['BEAM'].materials['Wood'].Elastic(table=((10000.0, 0.3), ))
mdb.models['BEAM'].HomogeneousSolidSection(material='Wood', name='BeamSection',
    thickness=None)
```



```python
#### ASSIGN SECTION ####
mdb.models['BEAM'].Material(name='Wood')
mdb.models['BEAM'].materials['Wood'].Elastic(table=((10000.0, 0.3), ))
mdb.models['BEAM'].HomogeneousSolidSection(material='Wood', name='BeamSection',
    thickness=None)
mdb.models['BEAM'].parts['Beam'].Set(cells=
    mdb.models['BEAM'].parts['Beam'].cells.getSequenceFromMask(('[#1 ]', ), ),
    name='Whole')
mdb.models['BEAM'].parts['Beam'].SectionAssignment(offset=0.0, offsetField='',
    offsetType=MIDDLE_SURFACE, region=
    mdb.models['BEAM'].parts['Beam'].sets['Whole'], sectionName='BeamSection',
    thicknessAssignment=FROM_SECTION)
########### ASSEMBLY MODULE ###########
mdb.models['BEAM'].rootAssembly.DatumCsysByDefault(CARTESIAN)
mdb.models['BEAM'].rootAssembly.Instance(dependent=ON, name='Beam-1', part=
    mdb.models['BEAM'].parts['Beam'])
########### STEP MODULE ###########
mdb.models['BEAM'].StaticStep(initialInc=0.1, name='BeamLoad', previous=
    'Initial')
########### LOAD MODULE ###########
mdb.models['BEAM'].rootAssembly.Surface(name='BeamLoad', side1Faces=
    mdb.models['BEAM'].rootAssembly.instances['Beam-1'].faces.getSequenceFromMask(
    ('[#2 ]', ), ))
mdb.models['BEAM'].Pressure(amplitude=UNSET, createStepName='BeamLoad',
    distributionType=UNIFORM, field='', magnitude=0.025, name='Pressure',
    region=mdb.models['BEAM'].rootAssembly.surfaces['BeamLoad'])
########### MESH MODULE ###########
mdb.models['BEAM'].parts['Beam'].setElementType(elemTypes=(ElemType(
    elemCode=C3D8R, elemLibrary=STANDARD, secondOrderAccuracy=OFF,
    distortionControl=DEFAULT), ElemType(elemCode=C3D6, elemLibrary=STANDARD),
    ElemType(elemCode=C3D4, elemLibrary=STANDARD)), regions=(
    mdb.models['BEAM'].parts['Beam'].cells.getSequenceFromMask(('[#1 ]', ), ),
    ))
mdb.models['BEAM'].parts['Beam'].seedPart(deviationFactor=0.1, minSizeFactor=
    0.1, size=50.0)
mdb.models['BEAM'].parts['Beam'].generateMesh()
```



```python
########### JOB MODULE ###########
mdb.models['BEAM'].rootAssembly.regenerate()
mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='BEAM', modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='Deform', nodalOutputPrecision=SINGLE,
    numCpus=2, numDomains=2, numGPUs=0, numThreadsPerMpiProcess=1, queue=None,
    resultsFormat=ODB, scratch='', type=ANALYSIS, userSubroutine='', waitHours=
    0, waitMinutes=0)
```

We observe that Abaqus use object-oriented programming (such as objects, constructors, methods, and members) because this aligns well with the Python programming.

For instance, in the Part module Abaqus creates a Model object named **Model-1** when the session is started. We can then access this object by for instance changing the name from **Model-1** to **BEAM** (as in Section 1), i.e., **mdb.models.changeKey[fromName='Model-1, toName='BEAM')**. We can now relate other commands to the object **mdb.models['BEAM']**. That is, the Model object has members with the same names and descriptions as the arguments to the Model method. Most of the Abaqus scripting interface commands are methods. For example,

mdb.models['BEAM'].ConstrainedSkecth(name='__profile__', sheetSize=2500.0)

In this example **ConstrainedSkecth()** is a method of the Model object.

A constructor is a method that creates an object. By convention, all constructor names and all objects start with an uppercase character in the Abaqus scripting interface. The name of a constructor is usually the same as the name of the type of object it creates. In the following example "models" is a constructor that creates a Part object called myFirstModel:

myFirstModel = mdb.models['BEAM'].Part(dimensionality=THREE_D, name='Beam', type=DEFORMABLE_BODY)

And then use a copy constructor to create a second Part object that is a copy of the first:

myFirstModel = mdb.models['BEAM'].Part(dimensionality=THREE_D, name='Beam', type=DEFORMABLE_BODY)

secondModel = mdb.models['BEAM'].Part(name='newBeam', objectToCopy=myFirstModel)

Some objects do not have a constructor. The object is created as a member of another object when the first object is created.

To summarize, after Abaqus creates an object, Abaqus then uses methods of the objects to enter or modify the data associated with the object.

The following list summarizes some of the concepts behind object-oriented programming and how they relate to the Abaqus scripting interface:

- An object encapsulates some data and functions that are used to manipulate those data.
- The data encapsulated by an object are called the members of the object.
- The functions that manipulate the data are called methods.
- The Abaqus scripting interface uses the convention that the name of a type of object begins with an uppercase character; for example, a Model object.
- A method that creates an object is called a constructor. The Abaqus scripting interface uses the convention that constructors begin with an uppercase character. In contrast, methods that operate on an object begin with a lowercase character.
- After you create an object, you then use methods of the object to enter or to modify the data associated with the object. For example, if you are creating an output database, you first create an Odb object. You then use the addNodes and addElements methods of the Part object to add nodes and elements, respectively. Similarly, you use the addData method of the FieldOutput object to add field output data to the output database.

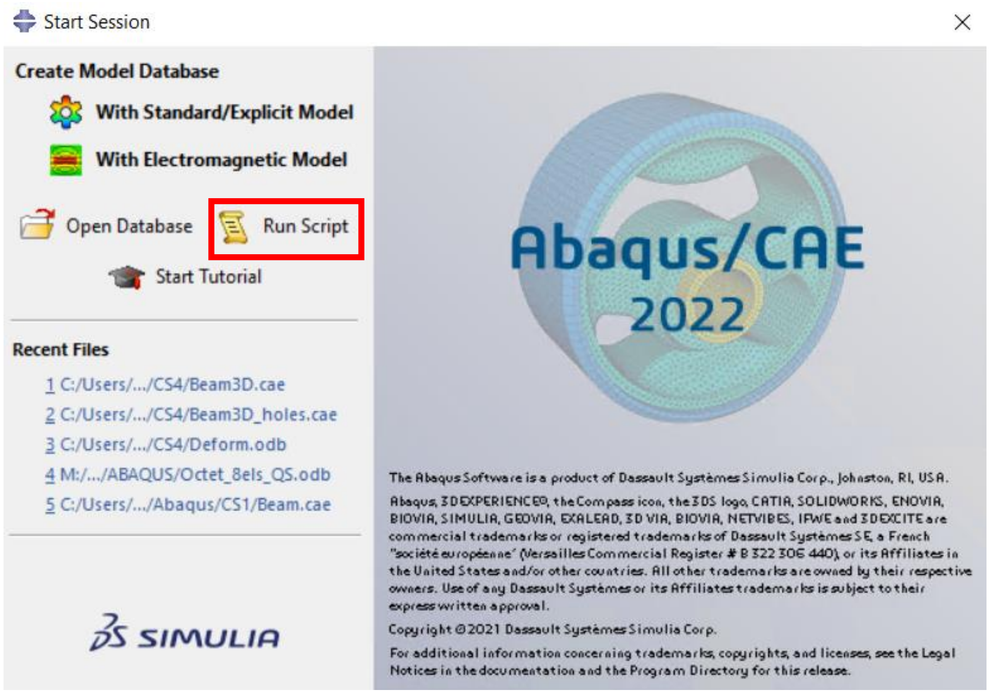
If interested, you can read more about the Abaqus scripting interface in the Abaqus Scripting User's Manual. You will also get help to guide you through the scripting interface if you take the time to ask Google.

If you want to test the Abaqus scripting interface, then save your .jnl file as a .py file. You probably want to clean up some of the commands because they are redundant, i.e., Abaqus saves your entire "clicking"-journey in the Abaqus/CAE GUI.

Note: An alternative to the journal (jnl) file is to use the Macro Manager in Abaqus CAE. You can find the **Macro Manager** in **File** -> **Macro Manager** -> **Create**, assign a **Name** and set **Directory** to **Work**. From when we click Continue, Abaqus will record Python scripting commands until we click **Stop Recording**. These Python commands are saved in a file called **abaqusMacros.py**. Each macro that we create is saved to a file called **abaqusMacros.py** in our **Work directory** as a function. We can copy the functions to another script that we can import into Abaqus.

Reopen Abaqus. Choose **File** → **Run script** → Choose your .py file containing the Python configuration commands to be run to build your model in Abaqus. Execute the script by choosing the .py file and click **OK**. Does it work? If so, then you are ready to for instance parameterize your model in the .py file. Relevant candidates for parameterization are the physical dimensions of your model, the mesh size, and the loading.

You can also run your script by choosing **Run script** directly from the Start session dialog box.


Start Session
Create Model Database
With Standard/Explicit Model
With Electromagnetic Model
Open Database
Run Script
Start Tutorial
Recent Files
1 C:/Users/.../CS4/Beam3D.cae
2 C:/Users/.../CS4/Beam3D_holes.cae
3 C:/Users/.../CS4/Deform.odb
4 M:/.../ABAQUS/Octet_8els_QS.odb
5 C:/Users/.../Abaqus/CS1/Beam.cae
SIMULIA
Abaqus/CAE
2022

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

Abaqus, 3DEXPERIENCE®, the Compass icon, the 3DS logo, CATIA, SOLIDWORKS, ENOVIA, BIOVIA, SIMULIA, GEOVIA, EXALEAD, 3D VIA, BIOVIA, NETVIBES, IFWE and 3DEXCITE are commercial trademarks or registered trademarks of Dassault Systèmes SE, a French "société européenne" (Versailles Commercial Register # B 322 306 440), or its Affiliates in the United States and/or other countries. All other trademarks are owned by their respective owners. Use of any Dassault Systèmes or its Affiliates trademarks is subject to their express written approval.

Copyright ©2021 Dassault Systèmes Simulia Corp.

For additional information concerning trademarks, copyrights, and licenses, see the Legal Notices in the documentation and the Program Directory for this release.