Exercise 26

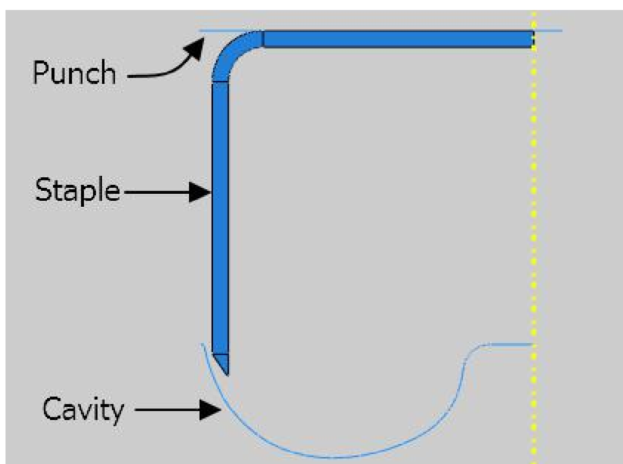
In this exercise we will simulate the bending of a staple using both Abaqus/Standard and Abaqus/Explicit.

A staple is a short piece of wire driven through and bent to bind together papers etc. The bending process takes place very slowly such that inertial effects are negligible and it can be termed as quasi-static. Quasi-static problems can be simulated well with either Abaqus/Standard or Abaqus/Explicit. Typically, quasi static problems are solved with Abaqus/Standard but may face convergence difficulties due to contact or other nonlinearities. Such problems are a good candidate to be solved with Abaqus/Explicit as explicit procedure can resolve complicated contact problems and other discontinuous nonlinearities more easily than Abaqus/Standard.

We will simulate the bending of a staple using both Abaqus/Standard and Abaqus/Explicit and compare. By taking advantage of the symmetry, only half of the problem will be simulated.

Pick **File > Set Work Directory** and set the work directory to Stapler folder

Open the model database **StapleFolding.cae**. It will appear as shown below.



The model consists of three parts: punch, cavity and staple. The punch and the cavity are modeled as discrete rigid parts. The staple is assumed to be made from an alloy with a Young's modulus of 90 GPa and a Poisson's ratio of 0.3. A coefficient of friction of 0.05 is assumed between contacting surfaces. The staple is meshed with bilinear plane stress quadrilateral elements with reduced integration (CPS4R).

In the model tree, it can be seen that there are two models. "Staple_Implicit" model will be used to perform analysis using the Abaqus/Standard and "Staple_Explicit" model will be used to perform analysis using the Abaqus/Explicit.

Solution with Abaqus/Standard

As the inertia effects can be ignored, we can perform the analysis using Abaqus/Standard.

Analysis Steps

The analysis will be performed in one step. The step has already been defined with a total time period set to **1.0** and the initial time increment to **0.01**. Due to large relative sliding of surfaces and friction, it is expected that the magnitude and influence of unsymmetric terms would be significant. So the unsymmetric solver has been specified for this step.

Boundary Conditions

Boundary conditions required for the analysis have already been defined. The punch is moved by a magnitude of 0.0089 in the y-direction with the “Punch” boundary condition. By taking advantage of symmetry, only half of the problem is modeled. Symmetry boundary condition has been defined accordingly for the staple.

Contact Interactions

Two contact pairs interactions, “Cavity-Staple” and “Punch-Staple”, have already been defined. “Cavity-Staple” defines contact between the cavity and the staple and “Punch-Staple” defines contact between the punch and the staple. A contact interaction property, named “Friction”, is specified for both contact interactions. In this contact interaction property, a friction coefficient of 0.05 has been specified.

Job Submission

All the information required for analysis has already been set up in the model. Now we can submit the job for analysis.

So change to **Job** module and open the Job Manager by picking .

Pick **Create** and create a job named BendImplicit or any other suitable name for the Staple_Implicit model .

Pick **Continue** and then **OK**.

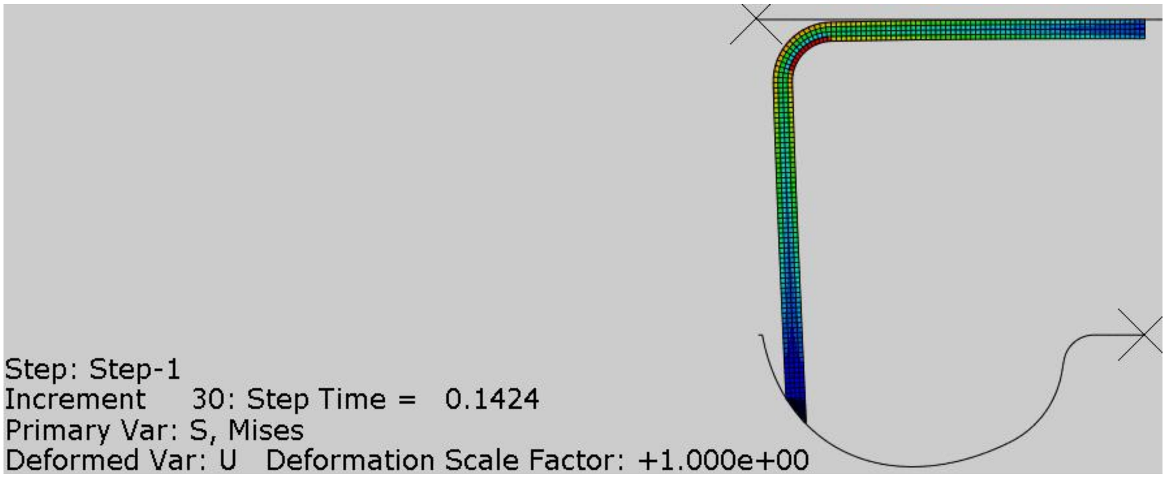
Pick **Submit** to submit the job for analysis and notice that job is aborted due to errors.

Diagnosing the error

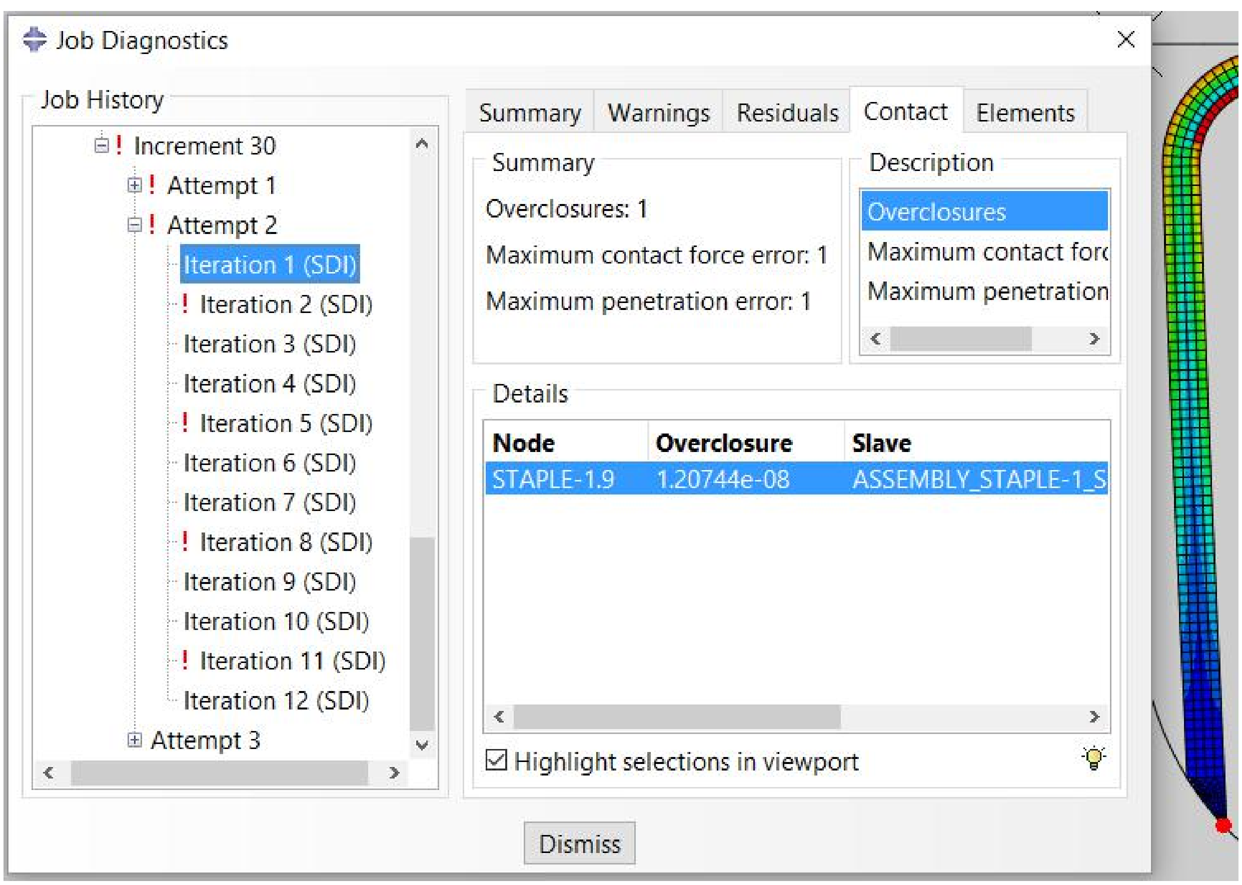
We will try to find the cause of this error in the Visualization module.

So pick **Results** to view the results in the Visualization module.

Pick  to plot the contours on deformed shape. It can be seen in the figure below that contact establishes successfully and job runs till 30th increment completing the 0.1424 of step time.



To investigate the cause of analysis termination, pick **Tools > Job Diagnostics** and Job Diagnostics dialog box will appear. It can be seen that during the 30th increment, system makes 3 attempts before aborting the analysis. During the 2nd attempt, there have been total 12 iterations and all of them are severe discontinuity iterations as shown in the figure below.

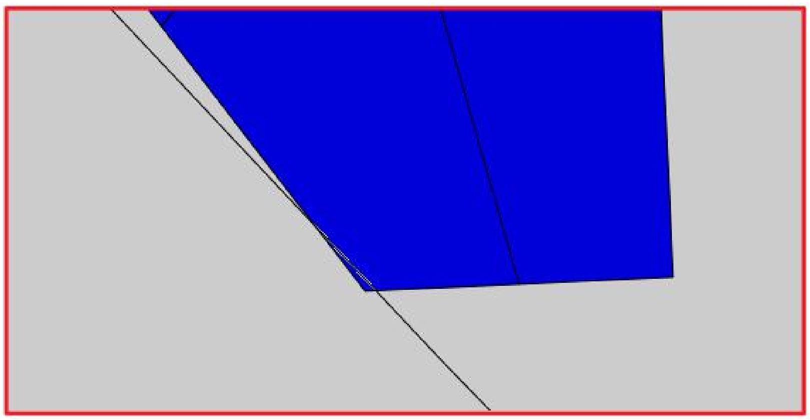


In the above figure, it can be seen that in the 1ST iteration, contact status for node "9" changes from open to closed. In the next iteration, contact status for the same node changes from closed to open.

If contact status for a node changes from closed to open, it is recorded as an "opening." If contact status changes from open to closed, it is recorded as an "overclosure."

When a node's status changes from closed to open in one iteration and in the next iteration its contact status changes from open to closed; it is said to be chattering and is an indication of instability in the contact.

It can be observed that bottom end of staple has a sharp corner which is penetrating the cavity surface as shown in the figure below.



These sharp corners make the resolution of contact a very difficult task. We will introduce the "softened" pressure-overclosure relationship to overcome the convergence problems at the contact interface.

The "softened" contact pressure-overclosure relationships are usually used to model a soft, thin layer on surfaces. In Abaqus/Standard they are also sometimes used because they can make it easier to resolve the contact difficulties.

⇨ Defining Contact Interaction Property

We will define a new contact interaction property using "Tabular" pressure-overclosure relationship. This contact property will be used for the "Cavity-Staple" interaction.

So change to **Interaction** module and make sure that "Staple_Implicit" model is active.

Pick  to create a new contact interaction property.

Select **Contact** as type and enter **Tabular-Friction** as name of the property.

Pick **Continue** and Edit Contact Property dialog box will appear.

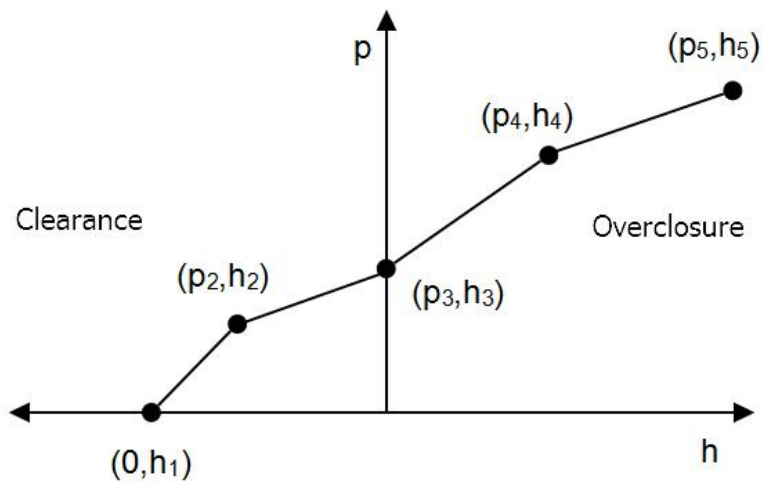
Pick **Mechanical > Tangential Behavior** to specify tangential behavior (friction).

Pick **Penalty** as friction formulation and enter **0.05** as friction coefficient.

Pick **Mechanical > Normal Behavior** to specify normal behavior.

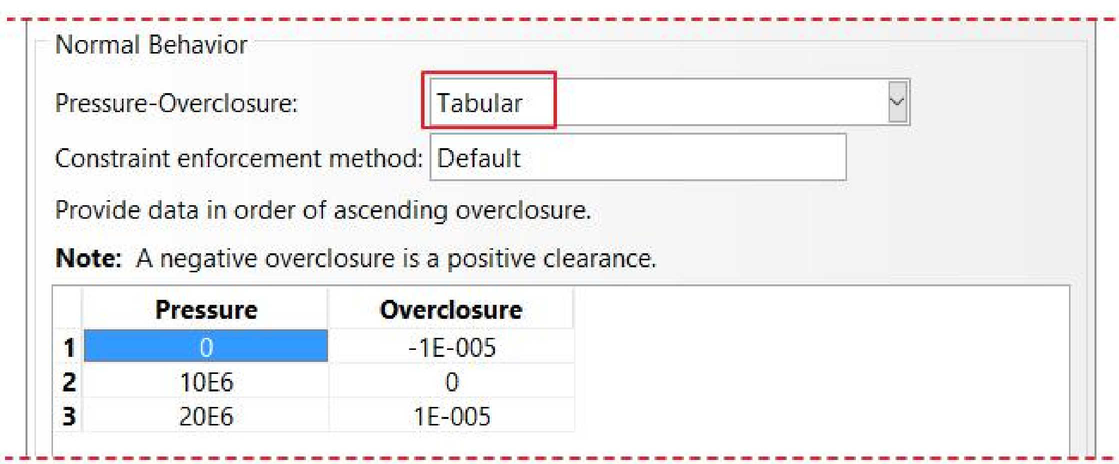
In the Pressure-Overclosure field, select **Tabular**.

Tabular law is used to define a pressure-overclosure relationship in tabular form in which the contact pressure is a piecewise linear function of the overclosure between the surfaces. To define a piecewise-linear relationship in tabular form, data pairs (p_i,h_i) of pressure versus overclosure (clearance corresponds to negative overclosure) are specified as shown in the figure below.



In this relationship the surfaces transmit contact pressure when the overclosure between them, measured in the contact (normal) direction, is greater than h_1, where h_1 is the overclosure at zero pressure. So the surfaces can transmit pressure even there is a clearance between them.

Enter **-1E-5** as the overclosure at **0** contact pressure and **0** as the overclosure for **10E6** contact pressure and **1E-5** as the overclosure for **20E6** contact pressure as shown in the figure below.



.

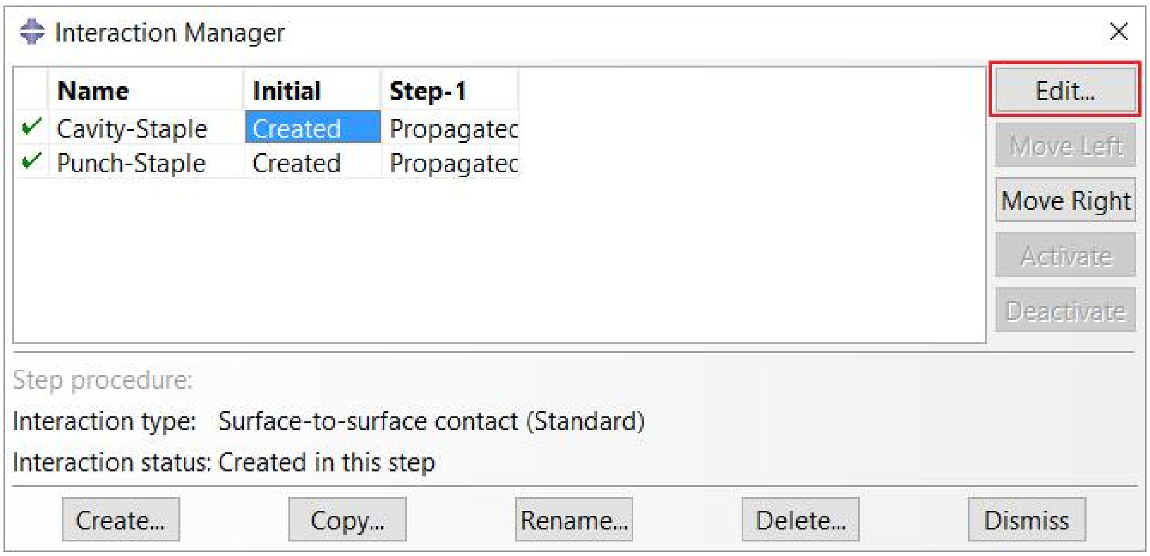
The data must be specified as an increasing function of pressure and overclosure.The data table must begin with a zero pressure. The pressure-overclosure relationship is extrapolated beyond the last overclosure point by continuing the same slope.

Pick **OK** to apply the changes and exit the dialog box.

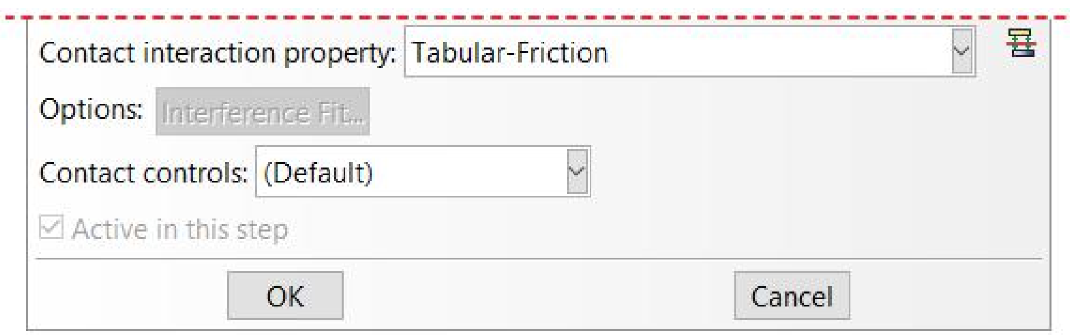
Modifying Contact Interaction

Open the Interaction Manager by picking .

Pick the "Cavity-Staple" interaction under column "Initial" and pick **Edit**.



Pick Tabular-Friction as the contact interaction property.



Pick **OK** to apply the changes and exit the dialog box.

Pick **Dismiss** to close the manager.

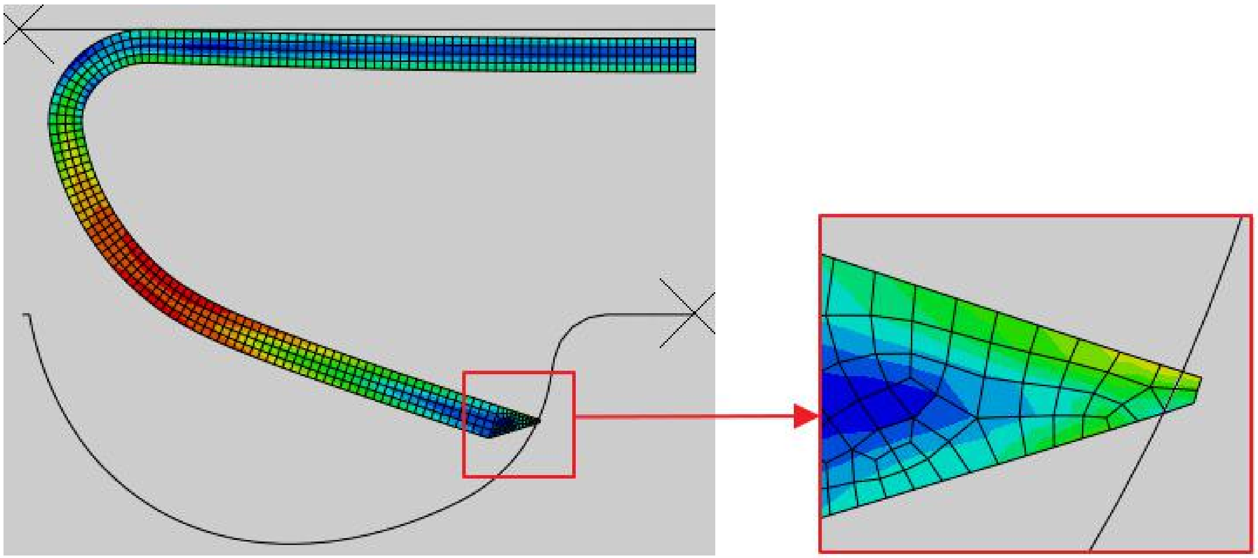
Now we will resubmit the job.
So change to **Job** module and open the Job Manager by picking .

Select BendImplicit and pick **Submit > OK** and notice that job is aborted due to errors.

We will try to find the cause of this error in the Visualization module.

So pick **Results** to view the results in the Visualization module.

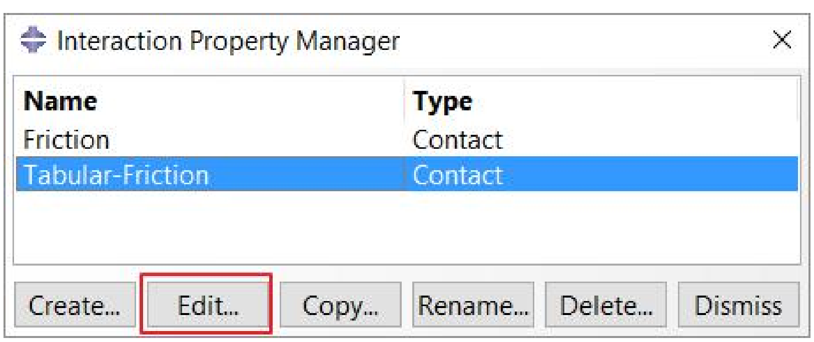
Pick  to plot the contours on deformed shape. The analysis job runs till 127th increment completing the 0.5212 of step time. It can be observed that bottom end of staple is penetrating the cavity surface as shown in the figure below.



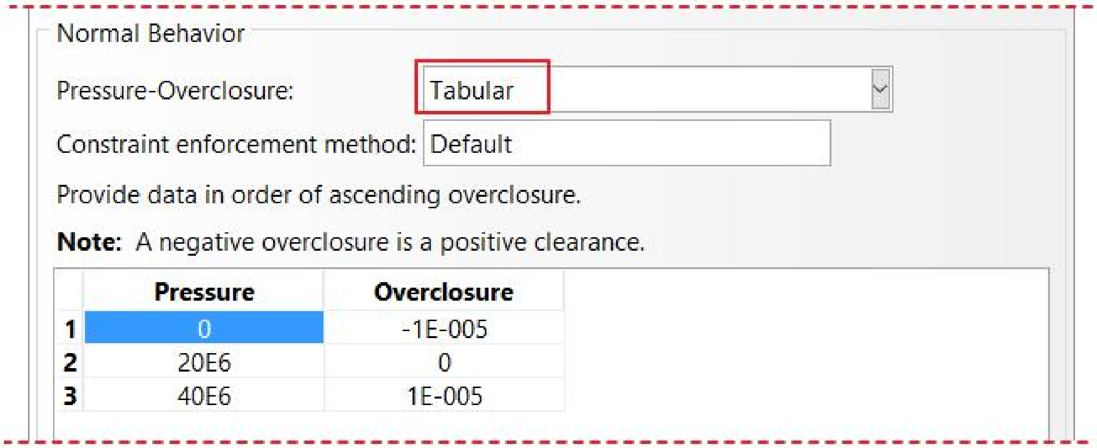
To fix this problem, we will modify the “Tabular-Friction” contact property such that a higher contact pressure is generated at the contact interface. It is assumed that the higher contact pressure will push the staple away from the cavity thus reducing or eliminating the penetration.

So change to **Interaction** module and open the Interaction Property Manager by picking .

Select Tabular-Friction and then pick **Edit** to modify the selected interaction property.



For the normal behavior option, enter **20E6** as contact pressure for the **0** overclosure and **40E6** as contact pressure for the **1E-5** overclosure as shown in the figure below.



Pick **OK** to apply the changes and exit the dialog box.

Now we will resubmit the job.

So change to **Job** module and open the Job Manager by picking .

Select BendImplicit and pick **Submit > OK** and notice that job completes successfully.

Solution with Abaqus/Explicit

For the explicit solution of the problem, we will use the "Staple_Explicit" model so make sure that it is active as shown below.

Analysis Steps

The analysis will be performed in one step. The step has already been defined. To reduce the time required to complete the analysis, the simulation will be performed at an artificially high speed. To find out an approximate lower bound on step time duration, a frequency analysis was performed in Abaqus/Standard. The lowest natural frequency of the staple was found to be 1631.8 Hz, corresponding to a time period of 0.00061 s. To make sure that inertial forces remain insignificant and quasi-static results are obtained, a time period of **0.005** s, a factor of approximately 10 times more than the time period corresponding to the lowest frequency, has been specified for the analysis.

Boundary Conditions

Boundary conditions required for the analysis have already been defined. The punch is moved by a magnitude of 0.0089 in the y-direction with the "Punch" boundary condition. A smooth amplitude curve has been specified for this boundary condition so that displacement loading takes place with smooth acceleration. This makes sure that the quasi-static analysis is performed without generating waves. By taking advantage of symmetry, only half of the problem is modeled. Symmetry boundary condition has been defined accordingly for the staple.

Contact Interactions

A contact pair interaction, “Punch-Staple”, has already been created. This defines contact between the punch and the staple. A contact interaction property, named “Friction”, is specified for the contact interaction. In this contact interaction property, a friction coefficient of 0.05 has been specified.

We will create another interaction to define contact between cavity and the staple.

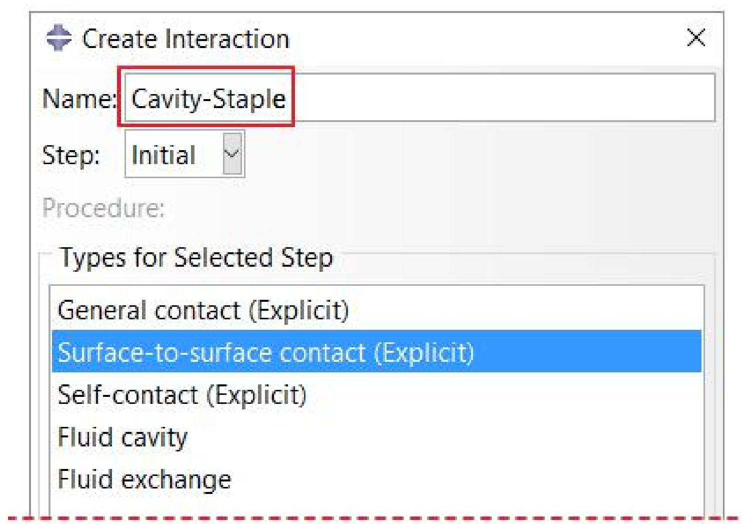
Change to **Interaction** module and open the Interaction Manager by picking .

Pick **Create** to define a new interaction.

Enter **Cavity-Staple** as the name of the interaction.

Pick **Initial** in the Step field.

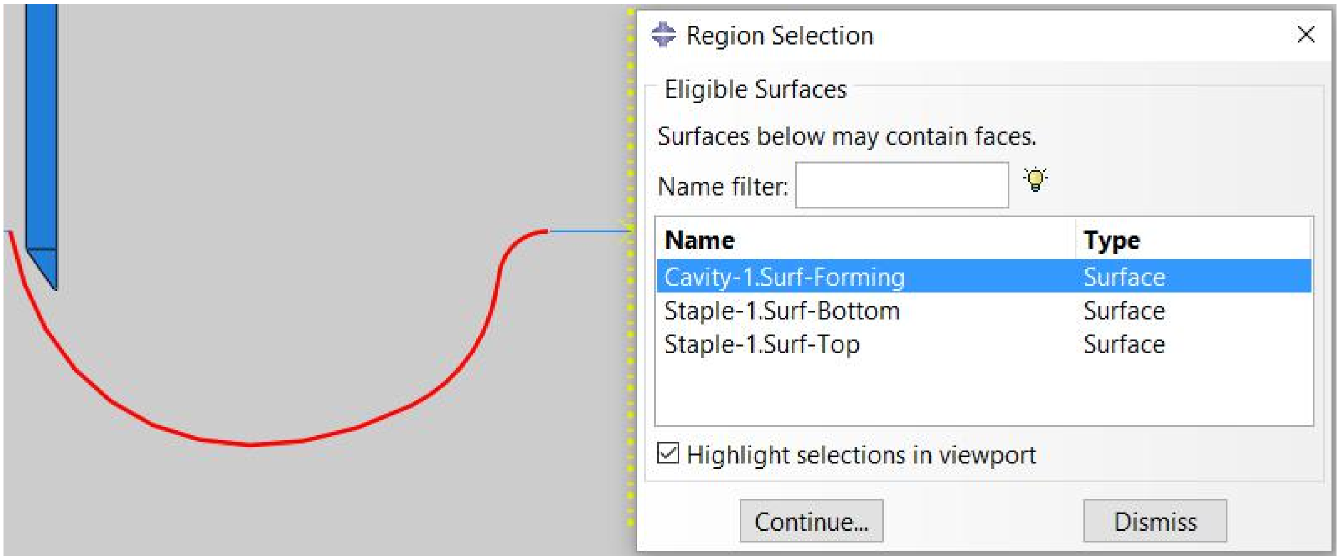
Select **Surface-to-surface contact** as type.



We can not define a general contact interaction in the current model as the general contact algorithm in Abaqus/Explicit can only be used with three-dimensional surfaces.

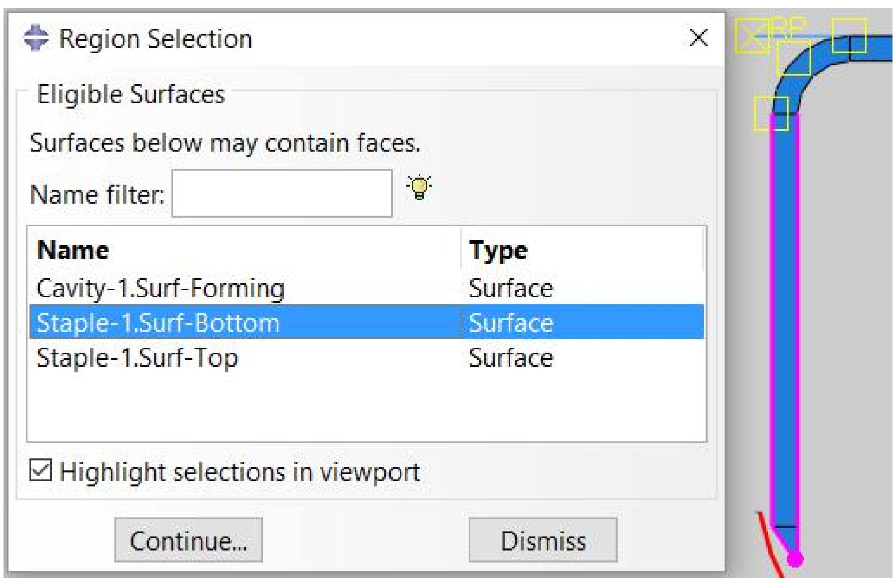
Pick **Continue** to proceed.

We have already defined a surface so pick **Surfaces** on the right side of the prompt area and select the Cavity-1.Surf-Forming.



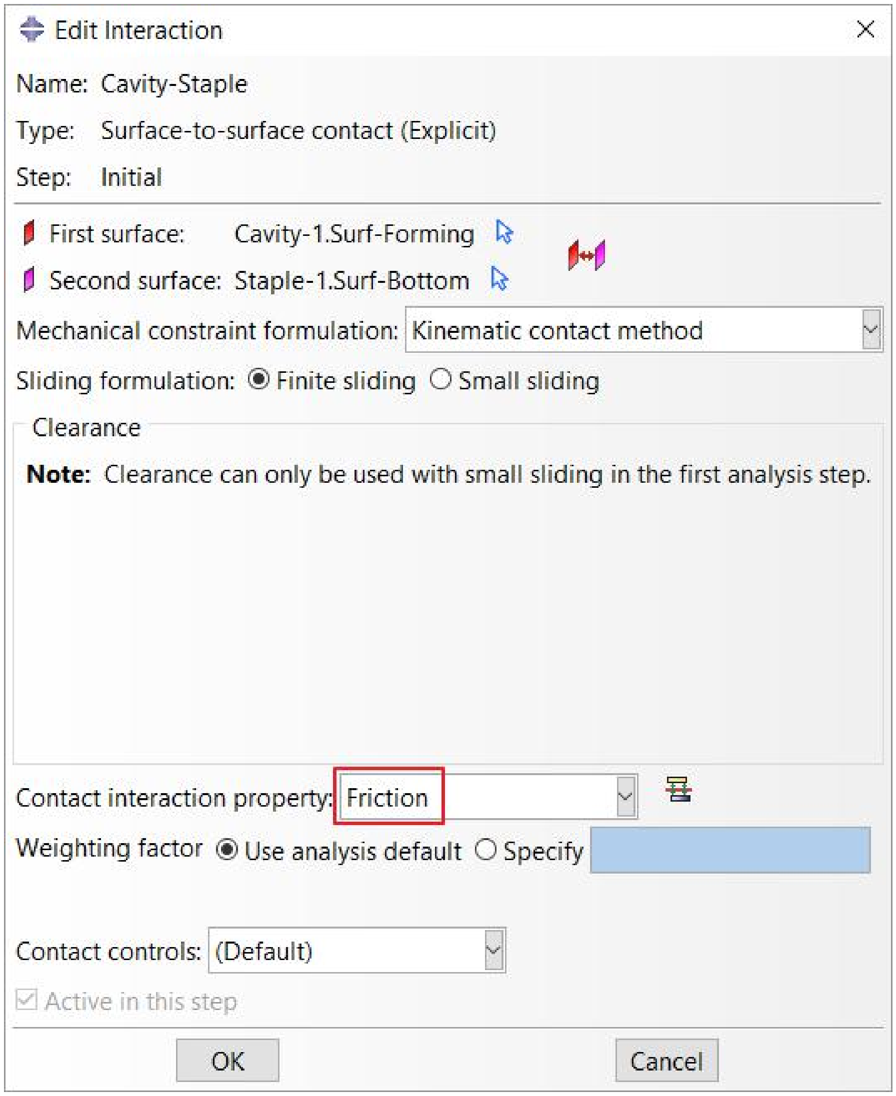
Pick **Continue** to proceed.

For the slave type, pick **Surface** and select the Staple-1.Surf-Bottom.



Pick **Continue** to proceed.

Pick **Friction** as contact interaction property.



Pick **OK** and it completes the definition of interaction.

Pick **Dismiss** to close the manager.

⇨ Job Submission

All the information required for analysis has already been set up in the model. Now we can submit the job for analysis.

So change to **Job** module and open the Job Manager by picking .

Pick **Create** and create a job named BendExplicit or any other suitable name for the Staple_Explicit model .

Pick **Continue** and then **OK**.

Pick **Submit** to submit the job for analysis and notice that job completes successfully.

After the job is completed, pick **Results** to view the results in the Visualization module.

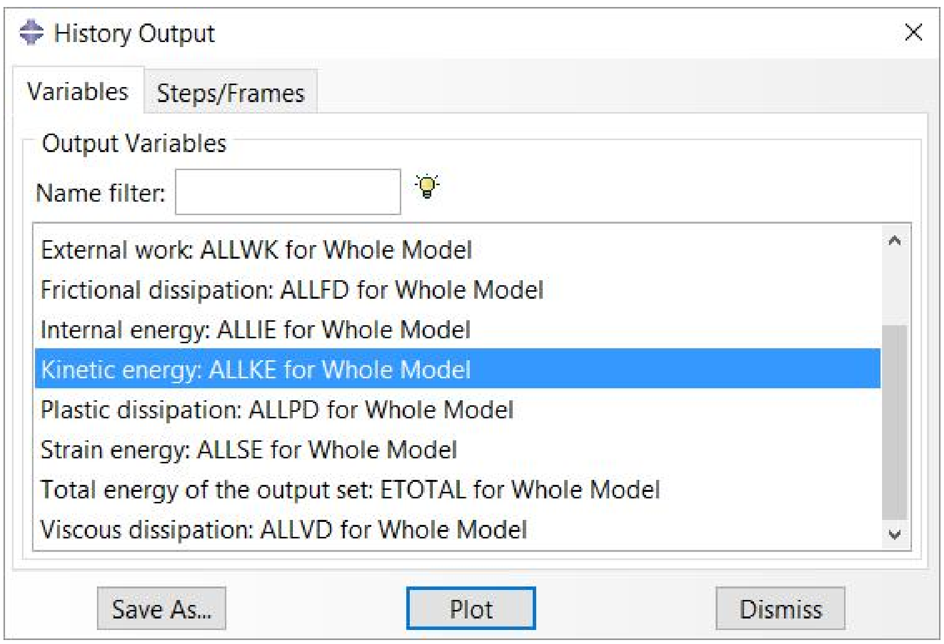
Comparing the Results

First we need to determine whether or not the solution obtained with Abaqus/Explicit is quasi-static. For that purpose we will compare the kinetic energy history to the internal energy history.

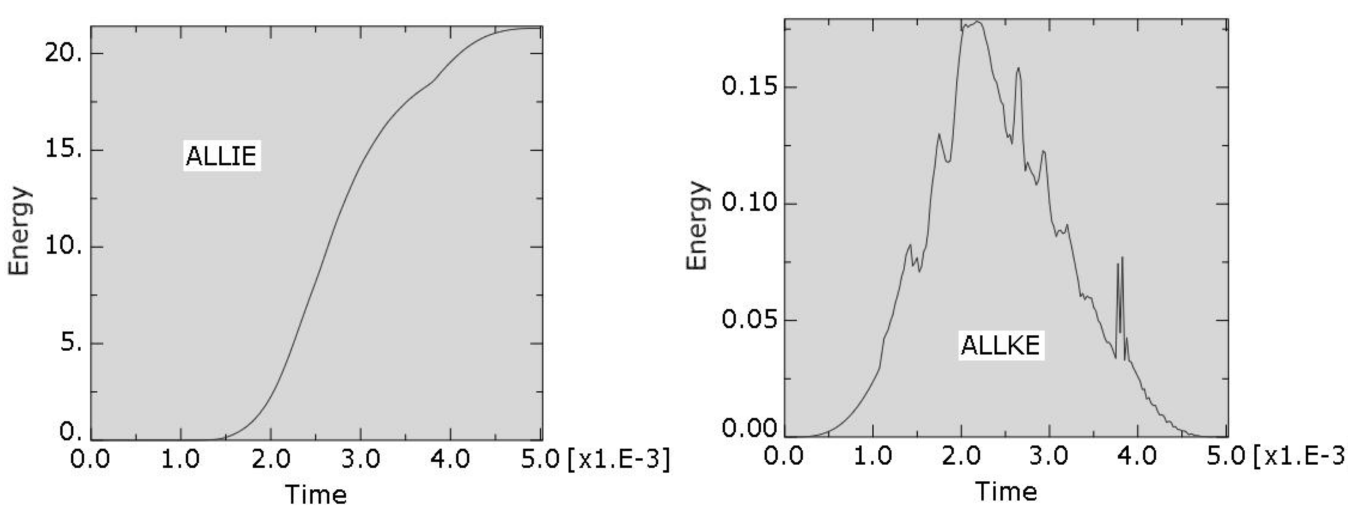
Make sure that BendExplicit is active in the Visualization module and pick .

In the XY Data dialog box, pick **ODB history output > Continue**

In the History Output dialog box, select **ALLKE** to plot the kinetic energy history and **ALLIE** to plot the internal energy history.



Pick **Plot** and the graph will appear. The following figure shows the graphs for both energy histories.



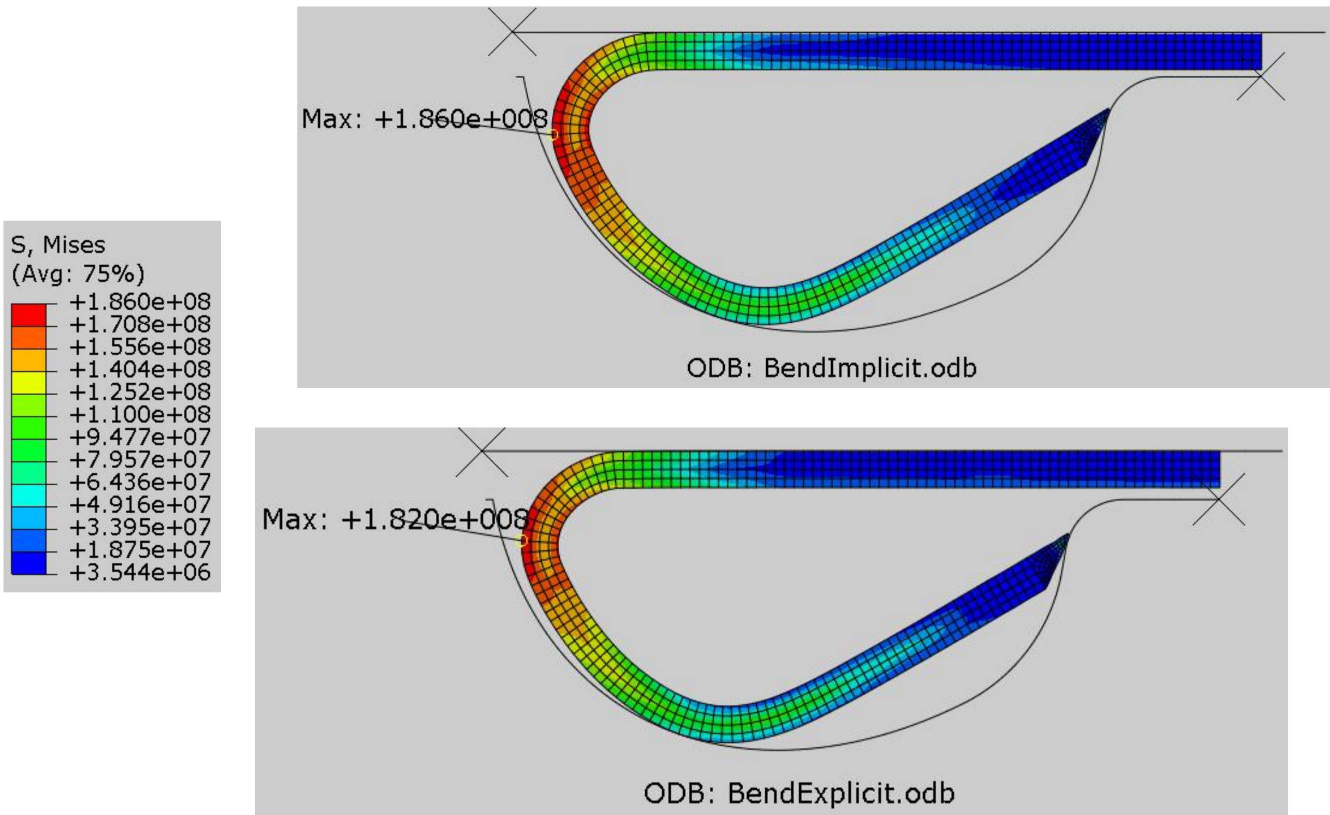
It can be seen that kinetic energy is less that 1% of the internal energy of the staple. Furthermore both histories seem appropriate and reasonable (no considerable oscillations). So we can conclude that solution obtained is indeed quasi-static.

Now we will compare the simulation results obtained by using Abaqus/Standard with those obtained with Abaqus/Explicit.

Pick  to plot the contours on deformed shape.

Select Mises in the Field Output toolbar if not selected by default.

The following figure compares the distribution of von Mises stresses. (Note: Legend is same for both plots).



The contour plots show that the stress distribution is almost similar for both cases and peak stresses are also very close to each other.

Select **File > Save** to save the changes we made so for.

Exercise 27

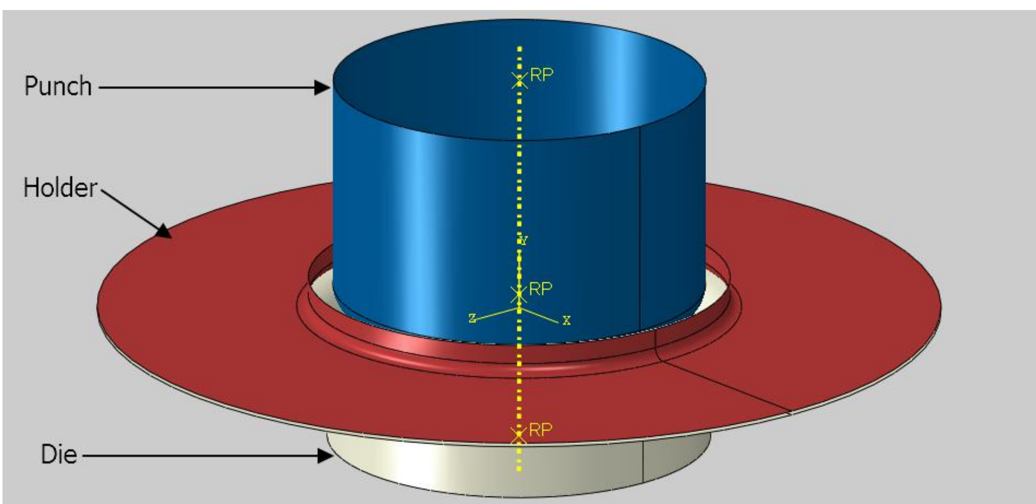
In this exercise we will simulate the deep drawing of a cup using both Abaqus/Standard and Abaqus/Explicit.

Deep drawing is a forming process in which a blank of sheet metal is drawn into a die by a moving punch. The blank is clamped by a blank holder against the die. The deep drawing process takes place slowly such that inertial effects are negligible and it can be termed as quasi-static. Quasi-static problems can be simulated well with either Abaqus/Standard or Abaqus/Explicit. Typically, quasi static problems are solved with Abaqus/Standard but may face convergence difficulties due to contact or other nonlinearities. Such problems are a good candidate to be solved with Abaqus/Explicit as explicit procedure can resolve complicated contact problems and other discontinuous nonlinearities more easily than Abaqus/Standard.

We will simulate the forming of the cup using both Abaqus/Standard and Abaqus/Explicit and compare. By taking advantage of the symmetry, only a quarter of the blank will be considered for simulation.

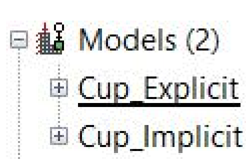
Pick **File > Set Work Directory** and set the work directory to Cup folder

Open the model database **DeepDraw.cae**. It will appear as shown below.



The model consists of four three-dimensional parts: punch, die, holder and blank. The punch, die and holder are modeled as analytical rigid parts. The blank is assumed to be made from steel with a Young's modulus of 200 GPa and a Poisson's ratio of 0.3. A coefficient of friction of 0.1 is assumed between contacting surfaces. The blank is meshed with linear, quadrilateral shell elements with reduced integration (S4R).

In the model tree, it can be seen that there are two models. "Cup_Implicit" model will be used to perform analysis using the Abaqus/Standard and "Cup_Explicit" model will be used to perform analysis using the Abaqus/Explicit.



Solution with Abaqus/Standard

As the inertia effects can be ignored, we can perform the analysis using Abaqus/Standard.

Analysis Steps

The analysis is completed in two steps. In the first step, a load is applied on the holder and in the second step the punch is moved downwards by applying a boundary condition.The steps have already been defined with a total time period set to **1.0** and the initial time increment to **0.01** for each step. Due to large relative sliding of surfaces and friction, it is expected that the magnitude and influence of unsymmetric terms would be significant during second step. So the unsymmetric solver has been specified.

Boundary Conditions

Boundary conditions required for the analysis have already been defined. The punch is moved by a magnitude of 0.03 in the y-direction with the “Punch” boundary condition. By taking advantage of symmetry, only a quarter of the blank is modeled. Symmetry boundary conditions have been defined accordingly for the blank. The holder can move only in the y-direction which allows the holder to accommodate changes in the blank thickness during deep drawing process.

Contact Interactions

Three contact pair interactions, “Die-Blank”, “Punch-Blank” and “Holder-Blank” have already been defined. “Die-Blank” defines contact between the die and the blank. “Punch-Blank” defines contact between the punch and the blank. “Holder-Blank” defines contact between the holder and the blank. A contact interaction property, named “Friction”, is specified for each contact interaction. In this contact interaction property, a friction coefficient of 0.1 has been specified.

Job Submission

All the information required for analysis has already been set up in the model. Now we can submit the job for analysis.

So change to **Job** module and open the Job Manager by picking .

Pick **Create** and create a job named DrawImplicit or any other suitable name for the Cup_Implicit model .

Pick **Continue** and then **OK**.

Pick **Submit** to submit the job for analysis and notice that job completes successfully.

Solution with Abaqus/Explicit

For the explicit solution of the problem, make sure that "Cup_Explicit" model is active as shown below.

Analysis Steps

The analysis will be performed in two steps. In the first step, a load is applied on the holder in a time period of 0.0001 s. This time period is considered to be long enough to avoid any dynamic effects but short enough so that it does not cause a significant impact on the run time of the analysis job. In the second step the punch is moved downwards by applying a boundary condition. For forming processes, typically, the punch speed is on the order of 1 m/s. Usually speed of forming events is increased artificially to obtain an economical solution. A time period of 0.006 s has been specified for the second step in which punch is moved by a magnitude of 0.03 in the y-direction. This gives average speed of 5 m/s to the punch.

Boundary Conditions

Boundary conditions required for the analysis have already been defined. The punch is moved by a magnitude of 0.03 in the y-direction with the "Punch" boundary condition. A smooth amplitude curve has been specified for this boundary condition so that displacement loading takes place with smooth acceleration. This makes sure that there is no sudden impact load onto the blank at the start of the analysis. By taking advantage of symmetry, only a quarter of the blank is modeled. Symmetry boundary conditions have been defined accordingly for the blank. The holder can move only in the y-direction which allows the holder to accommodate changes in the blank thickness during deep drawing process.

Contact Interaction

Now we will create a general contact interaction to define contact for the entire model.

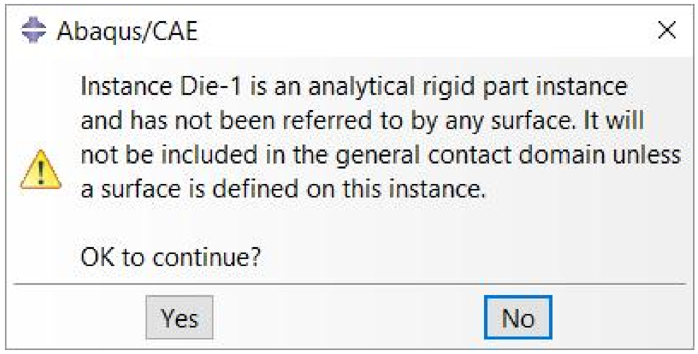
Change to **Interaction** module and open the Interaction Manager by picking .

Pick **Create** to define a new interaction.

Enter **All_General** as the name of the interaction.

Pick **Initial** in the Step field.

Select **General contact** and pick **Continue** to proceed. The following message will appear on screen.



General contact interaction requires an analytical part to be referred by a surface for it to be considered for a contact interaction. So we need to define a surface referring the analytical rigid part. We will continue to define the contact interaction and later define a surface referring the die. We have already defined surfaces for other rigid parts therefore there is no error message for them.

Pick **Yes** and Edit Interaction dialog box will appear.

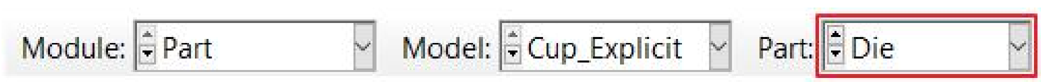
Pick **Friction** as the global contact interaction property.

Pick **OK** and it completes the definition of interaction.

Pick **Dismiss** to close the manager.

Now we will define a surface referring the die.

Change to **Part** module and select the Die part.

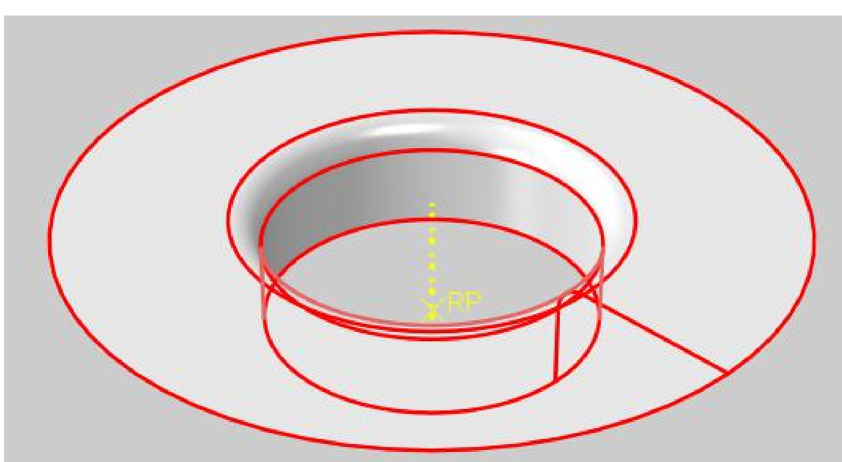


Open the Surface Manager by picking **Tools > Surface > Manager.**

Pick **Create** to define a new surface.

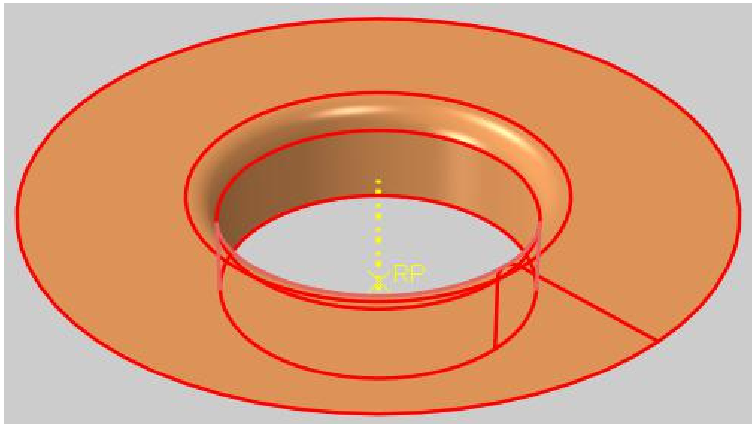
Pick **Continue** to create the surface with default name.

Pick the following surface as reference.



Pick **Done** to proceed.

Pick **Brown** for the side of shell in contact.



As a surface can contact on either side, so it is important to specify the desired side of the surface to be in contact. It is done by choosing the color associated with the desired side.

Pick **Dismiss** to close the manager.

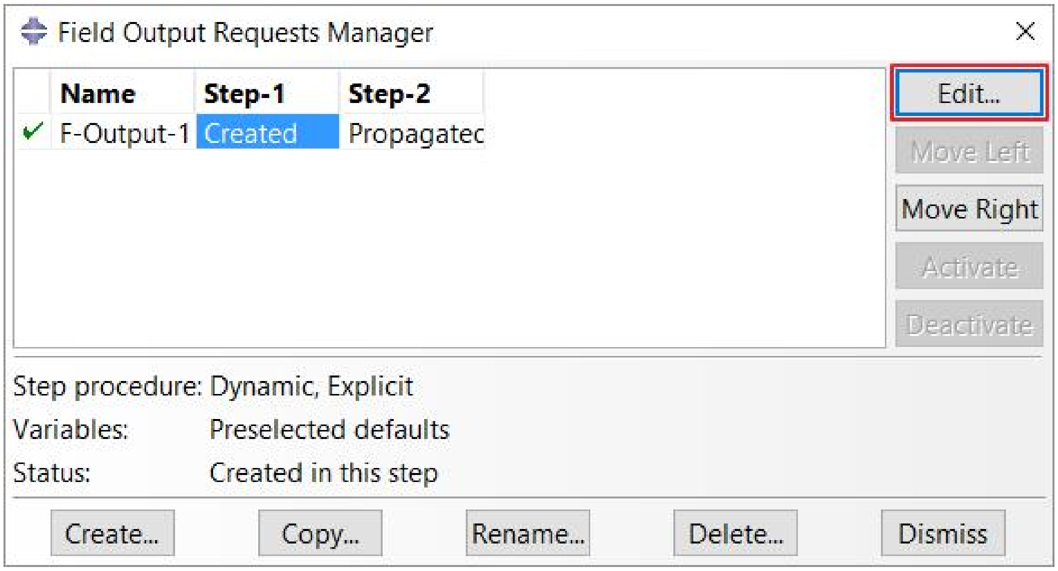
Field Output Requests

In sheet metal forming processes, changes in the thickness of blank are of great interest. The thickness for the shell elements can be monitored with output variable STH. This information is not included by default in the ODB file. We will modify the existing field and request STH variable.

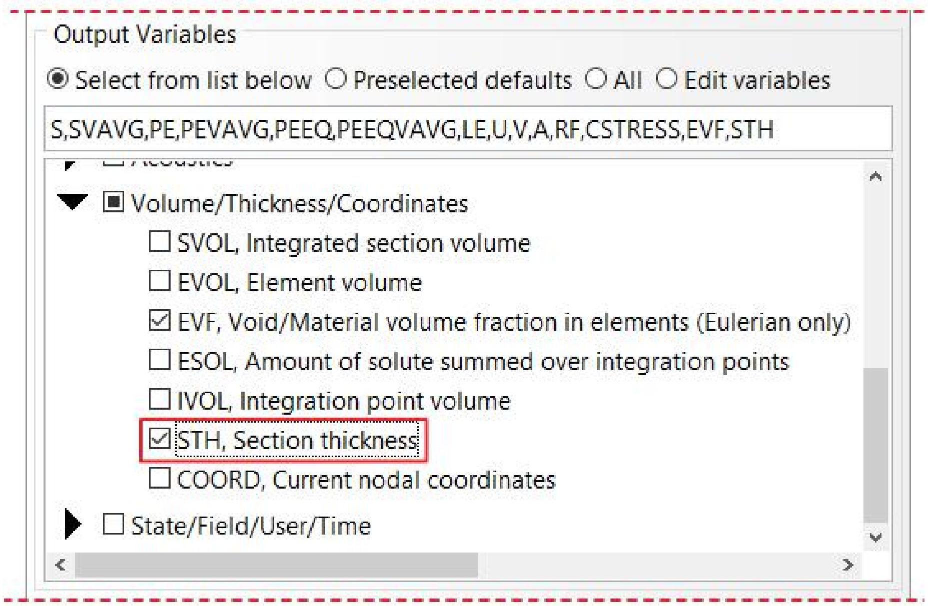
Change to **Step** module.

Open the Field Output Manager by picking  .

Select the F-Output-1 field and pick **Edit** as shown below.



Check the **STH** variable (located under the Volume/Thickness/Coordinates container).



Pick **OK** to apply and exit.

Pick **Dismiss** to close the manager.

⇨ Job Submission

All the information required for analysis has already been set up in the model. Now we can submit the job for analysis.

So change to **Job** module and open the Job Manager by picking .

Pick **Create** and create a job named DrawExp or any other suitable name for the Cup_Explicit model .

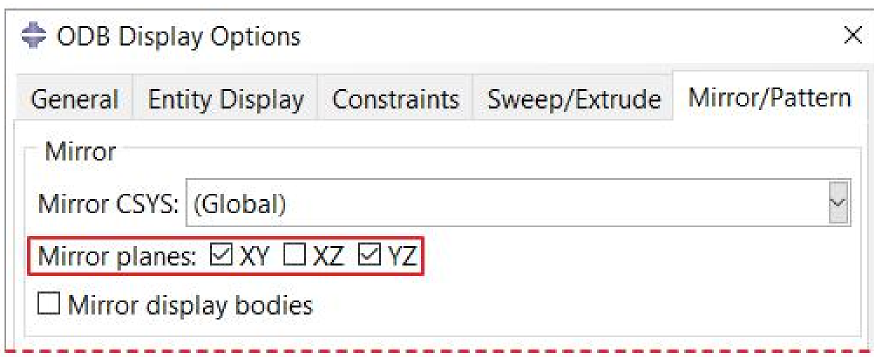
Pick **Continue** and then **OK**.

Pick **Submit** to submit the job for analysis and notice that job completes successfully.

Comparing the Results

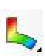
We have simulated only quarter of the problem by taking advantage of the symmetry. Now we will mirror the simulation results about symmetry planes so that complete shape of the cup could be seen on screen.

While in the Visualization module, pick **View > ODB Display Options** and check the XY an YZ options located under the Mirror/Pattern tab.



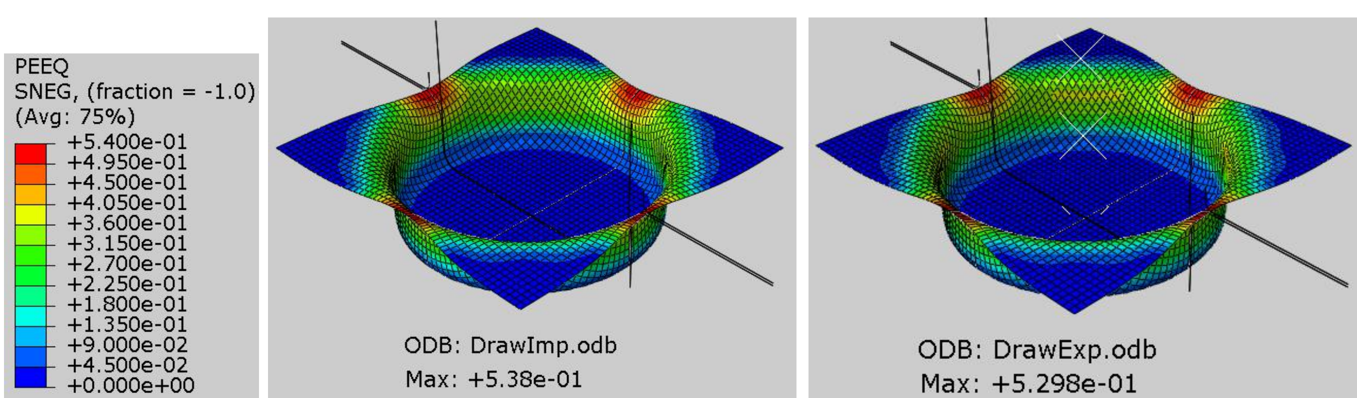
This will mirror the analysis results about the XY and YZ datum plane of the model.

Now we will compare the simulation results obtained by using Abaqus/Standard with those obtained with Abaqus/Explicit.

Pick  to plot the contours on deformed shape.

Select PEEQ in the Field Output toolbar.

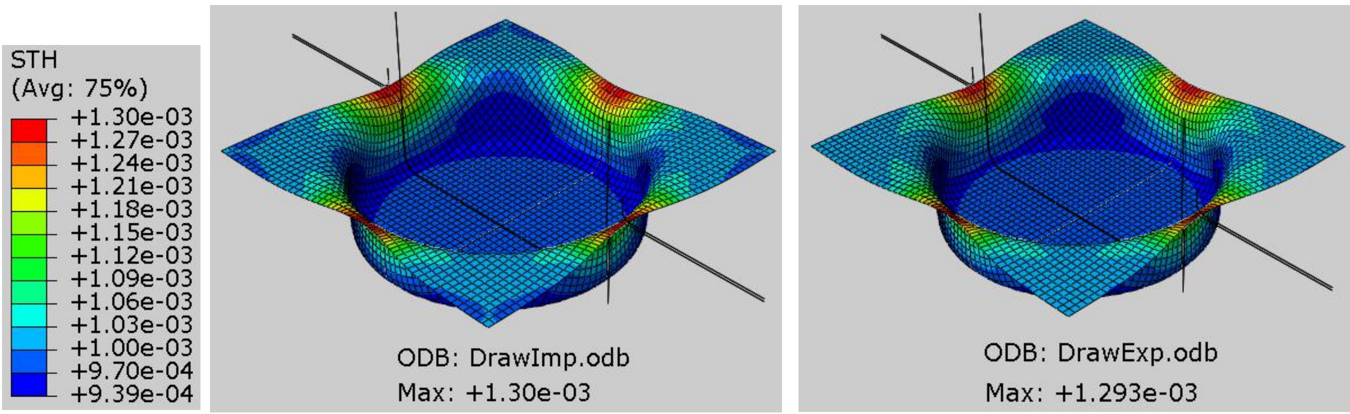
The following figure compares the distribution of equivalent plastic strain. (Note: Legend is same for both plots).



The contour plots show that the equivalent plastic strain distribution is almost similar for both cases.

Select STH in the Field Output toolbar to show contours of shell thickness in the blank.

The following figure compares the changes in the thickness of blank. (Note: Legend is same for both plots).



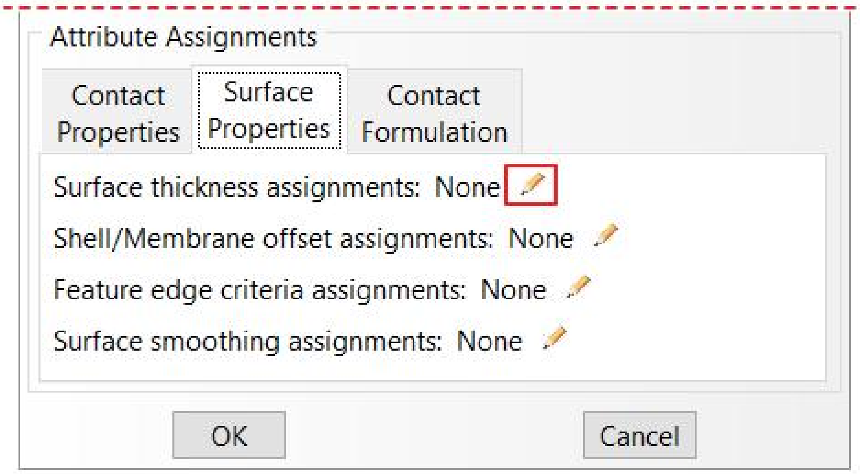
The predicted results are very similar. The contour plots show that in some areas thickness of blank has increased while in other areas thickness has reduced. The changes in thickness are significant, however general contact algorithm does not account for changes in shell thickness by default. In a sheet forming analysis, thinning of a sheet significantly influences contact. We will modify the general contact interaction so that thickness changes are considered.

Modifying Contact Interaction

Change to **Interaction** module and open the Interaction Manager by picking .

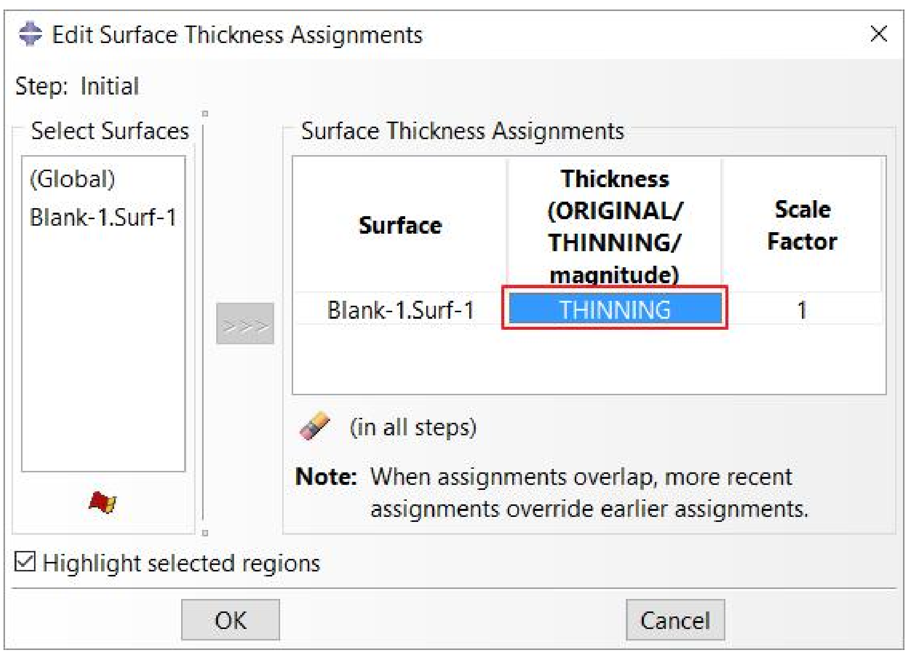
Pick the "All_General" interaction under column "Initial" and pick **Edit**.

Pick under the Surface Properties tab to specify surface thickness assignments.



Pick **Blank-1.Surf-1** and pick >>> in the dialog box to transfer the selection to the list of surface thickness assignments.

Enter **THINNING** in the thickness field. (Tip: Enter T and hit enter key and system will write THINNING in the thickness field.



THINNING option is used to consider the current shell thickness instead of the original shell thickness throughout the analysis.

Pick **OK** to apply the changes and exit dialog box.

Pick **OK** and it completes the definition of interaction.

Pick **Dismiss** to close the manager.

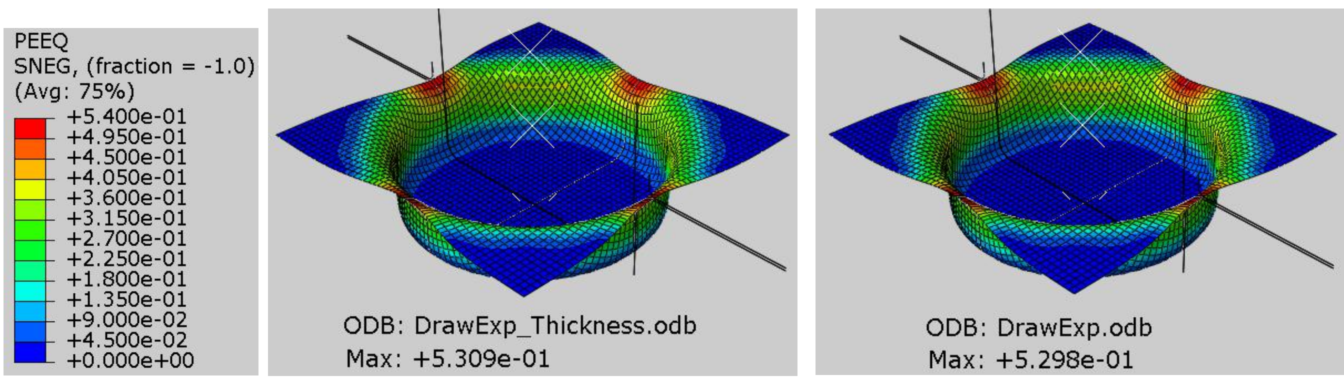
Now we will submit the job. So change to **Job** module and open the Job Manager by picking .

Pick **Create** and create a job named DrawExp_Thickness or any other suitable name for the Cup_Explicit model .

Pick **Continue** and then **OK**.

Pick **Submit** to submit the job for analysis and notice that job completes successfully.

The following figure compares the distribution of equivalent plastic strain. (Note: Legend is same for both plots).



The contour plots show that the distribution of equivalent plastic strain is slightly different when changes in thickness of blank are considered.

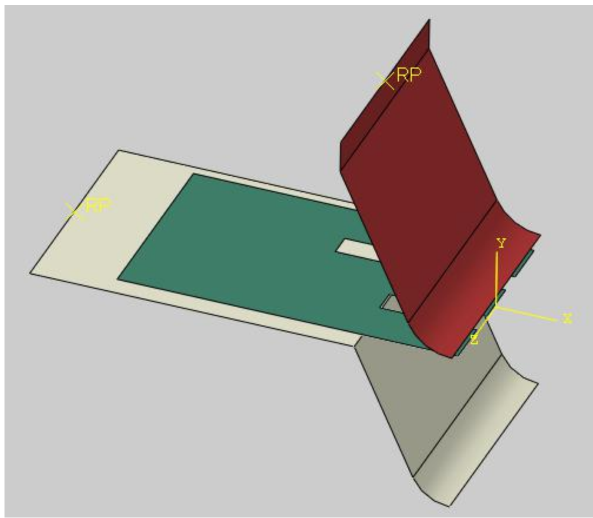
Select **File > Save** to save the changes we made so for.

Exercise 28

In this exercise we will simulate the v-bending of a sheet metal blank using the Abaqus/Explicit. We will compare the results obtained with contact pair approach and general contact approach.

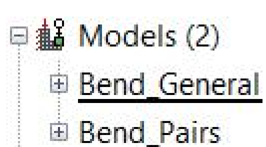
Pick **File > Set Work Directory** and set the work directory to VBending folder

Open the model database **Bending.cae**. It will appear as shown below.



The model consists of three three-dimensional parts: punch, die and blank. The punch and die are modeled as analytical rigid parts. The blank is assumed to be made from steel with a Young's modulus of 180 GPa and a Poisson's ratio of 0.3. A coefficient of friction of 0.1 is assumed between contacting surfaces. The blank is meshed with linear, quadrilateral shell elements with reduced integration (S4R). By taking advantage of the symmetry, only a half of the blank will be considered for simulation.

In the model tree, it can be seen that there are two models. "Bend_General" model will be used to perform analysis using the general contact approach and "Bend_Pairs" model will be used to perform analysis using the contact pairs.



Analysis Steps

The analysis will be performed in one step. In this step the punch is moved downwards by applying a boundary condition. For forming processes, typically, the punch speed is on the order of 1 m/s. Usually speed of forming events is increased artificially to obtain an economical solution. A time period of 0.004 s has been specified for the step while the punch is moved by a magnitude of 0.016 in the y-direction. This gives an average speed of 4 m/s to the punch.

Boundary Conditions

Boundary conditions required for the analysis have already been defined. The punch is moved by a magnitude of 0.016 in the y-direction with the "Punch" boundary condition. A smooth amplitude curve has been specified for this boundary condition so that displacement loading takes place with smooth acceleration. This makes sure that there is no sudden impact load onto the blank at the start of the analysis. By taking advantage of symmetry, only a half of the blank is modeled. Symmetry boundary conditions have been defined accordingly for the blank.

Contact Interaction

Now we will create a general contact interaction to define contact for the entire model.

First make sure that "Bend_General" model is active as shown below.

Change to **Interaction** module and open the Interaction Manager by picking .

Pick **Create** to define a new interaction.

Enter **All_General** as the name of the interaction.

Pick **Initial** in the Step field.

Select **General contact** and pick **Continue** to proceed.

Pick **Friction** as the global contact interaction property.

Pick **OK** and it completes the definition of interaction.

Pick **Dismiss** to close the manager.

Job Submission

All the information required for analysis has already been set up in the model. Now we can submit the job for analysis.
So change to **Job** module and open the Job Manager by picking .

Pick **Create** and create a job named VBendGen or any other suitable name for the Bend_General model .

Pick **Continue** and then **OK**.

Pick **Submit** to submit the job for analysis and notice that job completes successfully.

Also submit a job for the Bend_Pairs model, named VBendPairs, so that a comparison could be made.

Comparing the Results

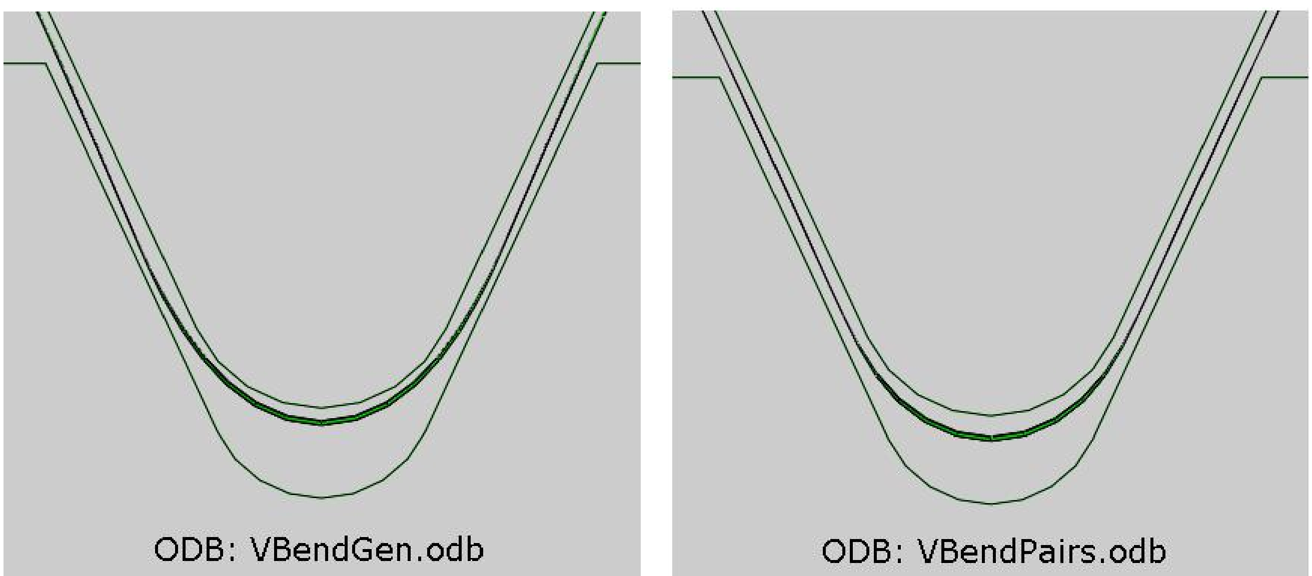
We have simulated only half of the problem by taking advantage of the symmetry. Now we will mirror the simulation results about symmetry plane.

While in the Visualization module, pick **View > ODB Display Options** and check the YZ option located under the Mirror/Pattern tab. This will mirror the analysis results about the YZ datum plane of the model.

Now we will compare the simulation results obtained by using contact pair approach with general contact approach.

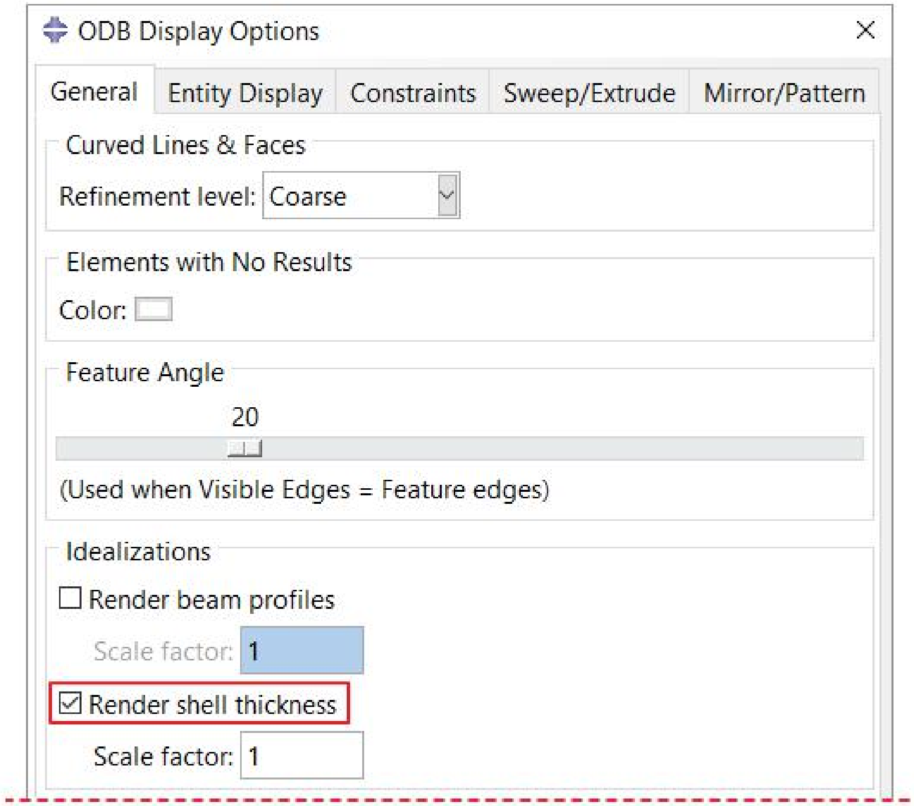
Pick  to plot the deformed shape.

The following figure compares the deformed shapes of both parts.



The deformed shape plots show that results are not similar for both cases. For clarity, we will visualize the shell thickness.

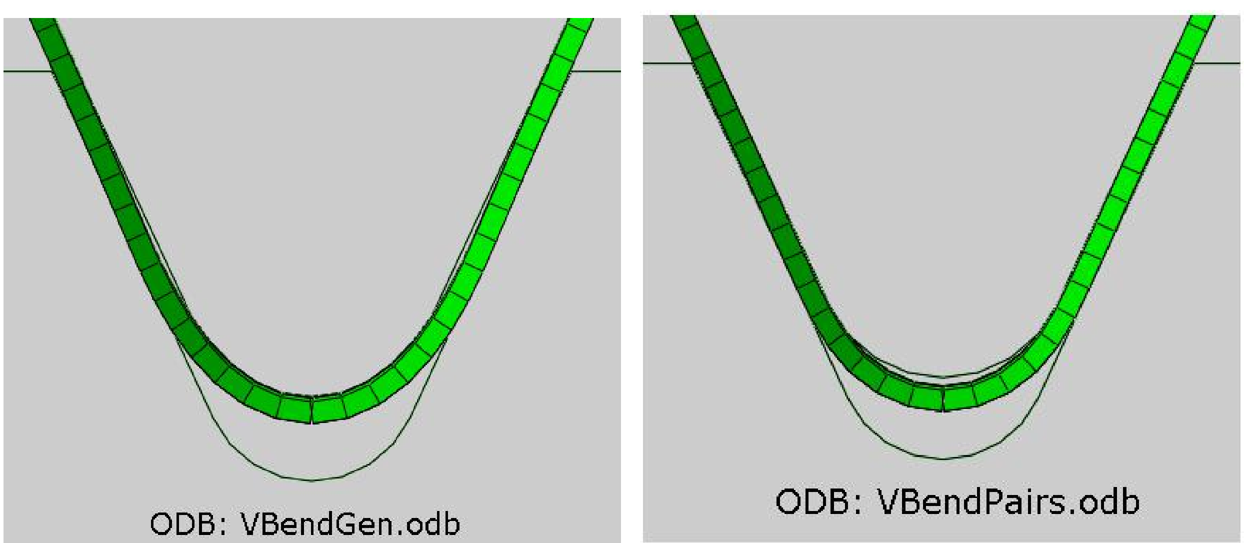
While in the Visualization module, pick **View > ODB Display Options** and check the **Render shell thickness** option located under the General tab.



The thickness representation uses the thickness defined in the section assignment properties to render the shell thickness.

Pick **OK** to apply and exit.

The following figure compares the deformed shape of blank for both cases while visualizing the shell thickness.



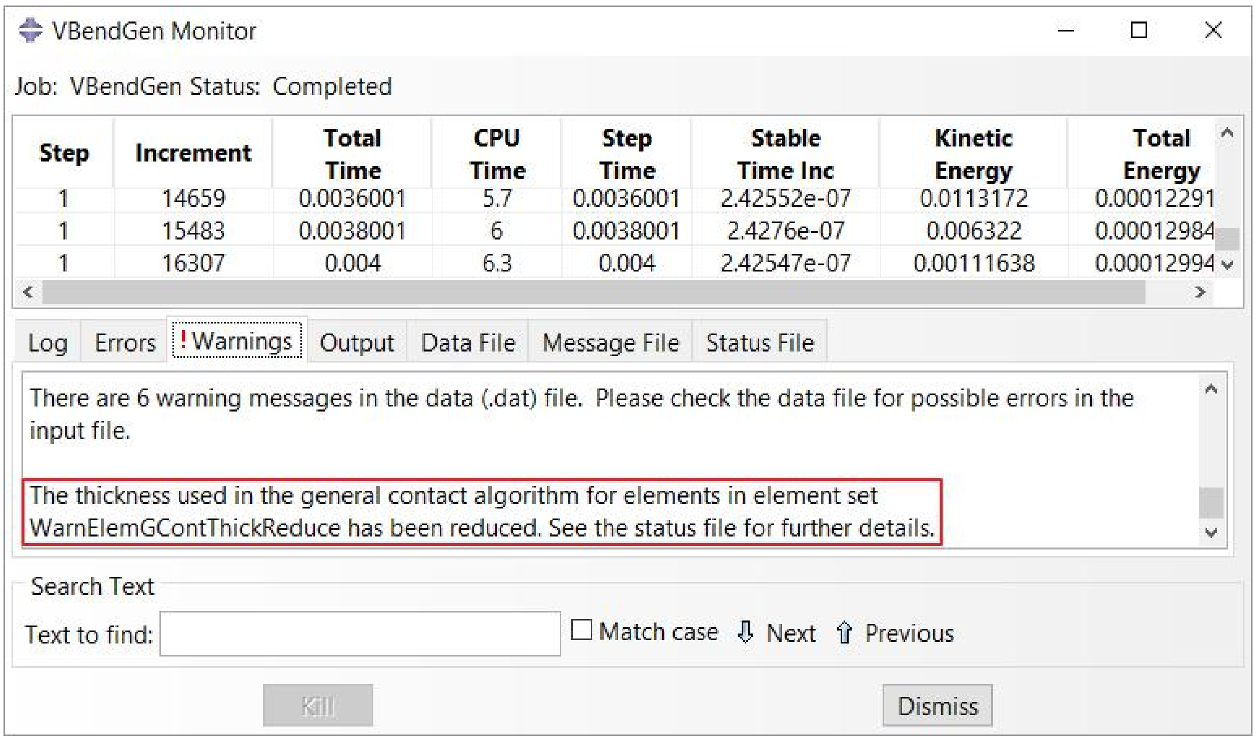
It appears that blank is penetrating the die in the solution obtained with general contact approach. So we need to investigate it further. We will start the investigation by using job monitor.

So change to **Job** module and open the Job Manager by picking .

Now select VBendGen job in the manager and pick **Monitor**

No error message is reported in the Errors tab.

Pick the **Warnings** tab and notice the warning about thickness reduction as highlighted below.

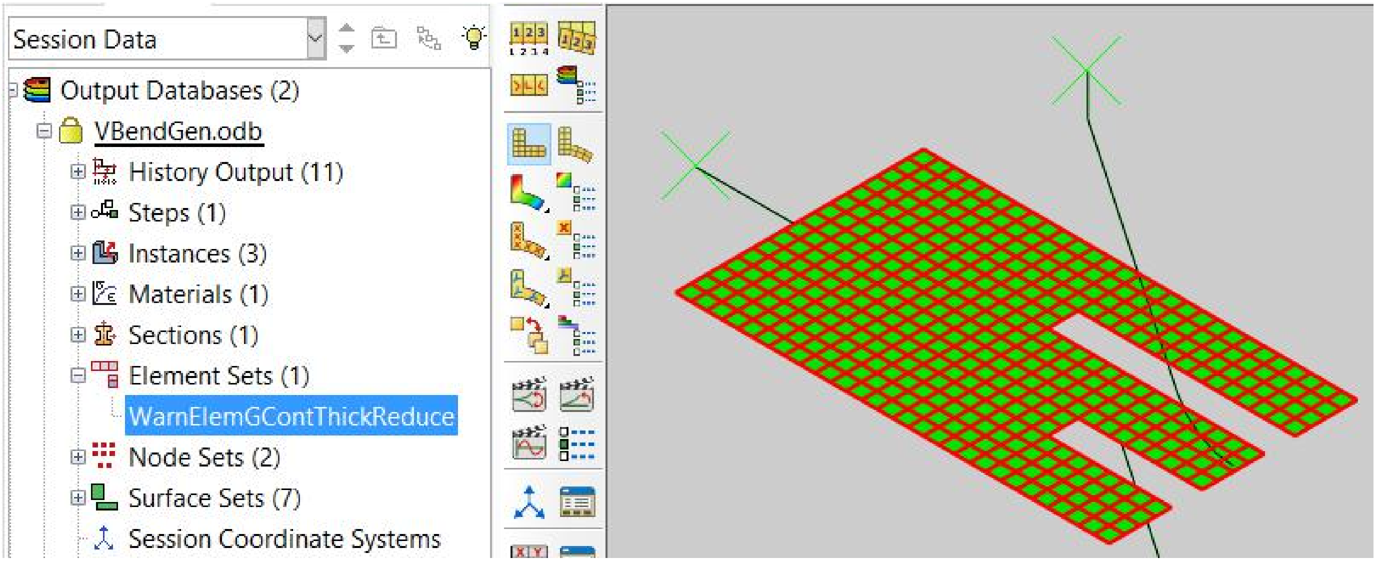


We can view the mentioned element set on screen in the Visualization module.

So pick **Results** in the manager to view the results in the Visualization module.

In the Results Tree, expand the Element Sets container underneath the VBendGen.odb

Pick on the WarnElemGContThickReduce and corresponding element set will highlight on the screen as shown below.



So Abaqus has reduced the thickness of all the elements of the blank.

By default, the general contact algorithm requires that the contact thickness does not exceed a certain fraction of the surface facet edge lengths or diagonal lengths. This fraction generally varies from 20% to 60% based on the geometry of the element and whether the element is near a shell perimeter. The general contact algorithm will scale back the contact thickness automatically where necessary.

To view the reduction in shell thickness in a general contact interaction, Abaqus provides CTHICK output variable. So we will request the the CTHICK variable and resubmit the job.

Field Output Requests

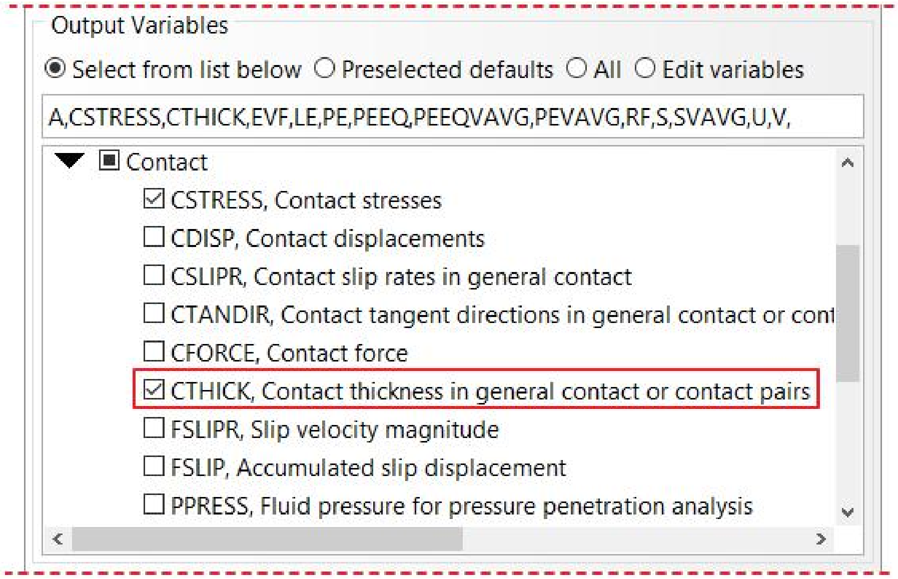
CTHICK variable is not included by default in the ODB file. We will modify the existing field and request CTHICK variable.

Change to **Step** module.

Open the Field Output Manager by picking  .

Select the F-Output-1 field and pick **Edit**.

Check the **CTHICK** variable (located under the Contact container).



Pick **OK** to apply and exit.

Pick **Dismiss** to close the manager.

Now we will resubmit the job.

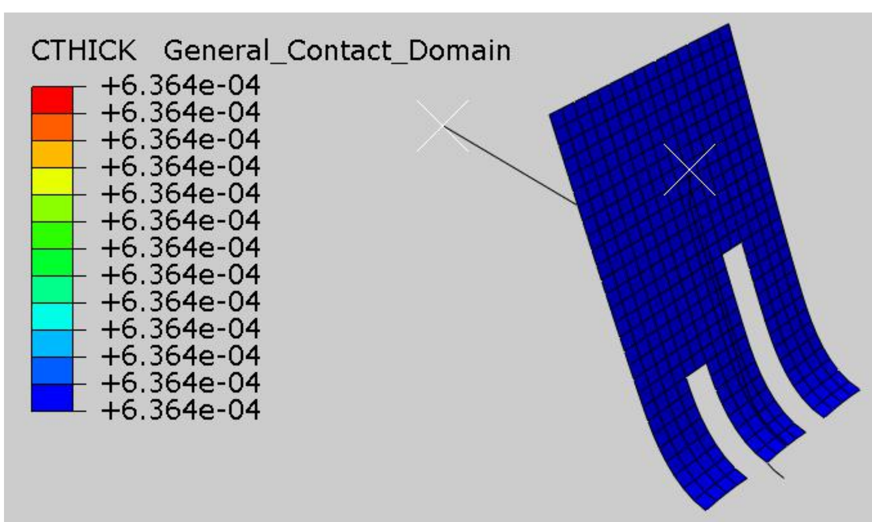
So change to **Job** module and open the Job Manager by picking .

Select VBendGen and pick **Submit > OK.**

After the job is completed, pick **Results** to view the results in the Visualization module.

Select the **CTHICK** variable in the Field Output toolbar .

Now the simulation results will appear as shown below.



It can be seen that the Abaqus has reduced the shell thickness to 0.0006364 (in the section definition of the shell a thickness of 0.001 is specified.)

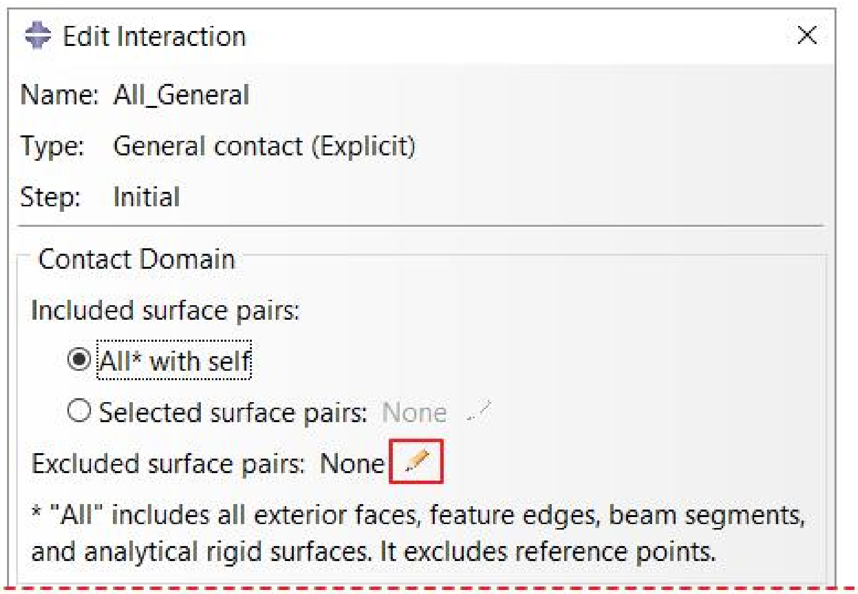
There are two ways to fix this issue: one is to use a coarser mesh and other is to exclude self-contact for the blank. In this exercise we will use the second method to avoid the reduction in shell thickness.

Modifying Contact Interaction

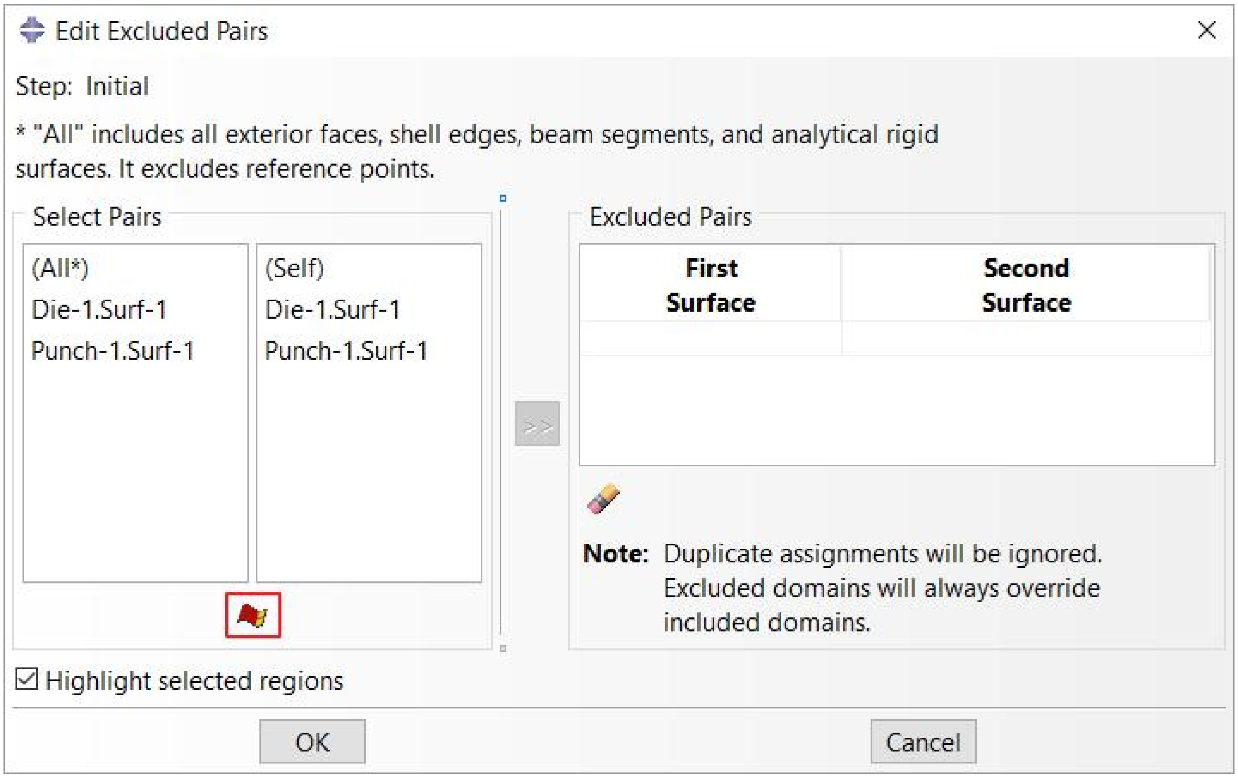
Change to **Interaction** module and open the Interaction Manager by picking .

Pick the “All_General” interaction under column “Initial” and pick **Edit**.

Pick to specify surface exclusions.

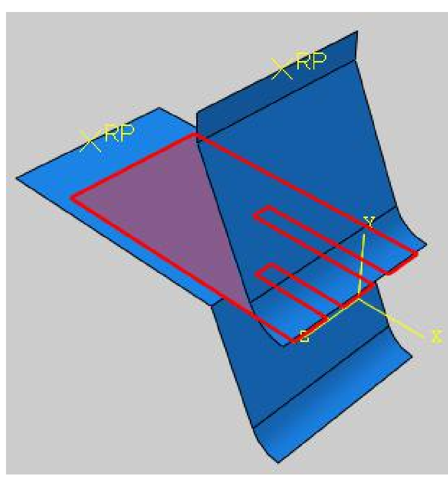


In the Edit Excluded Pairs dialog box it can be seen that there is no surface representing the blank. So pick to define a new surface.



Enter **Surf-Blank** as the name of the surface and pick **Continue**.

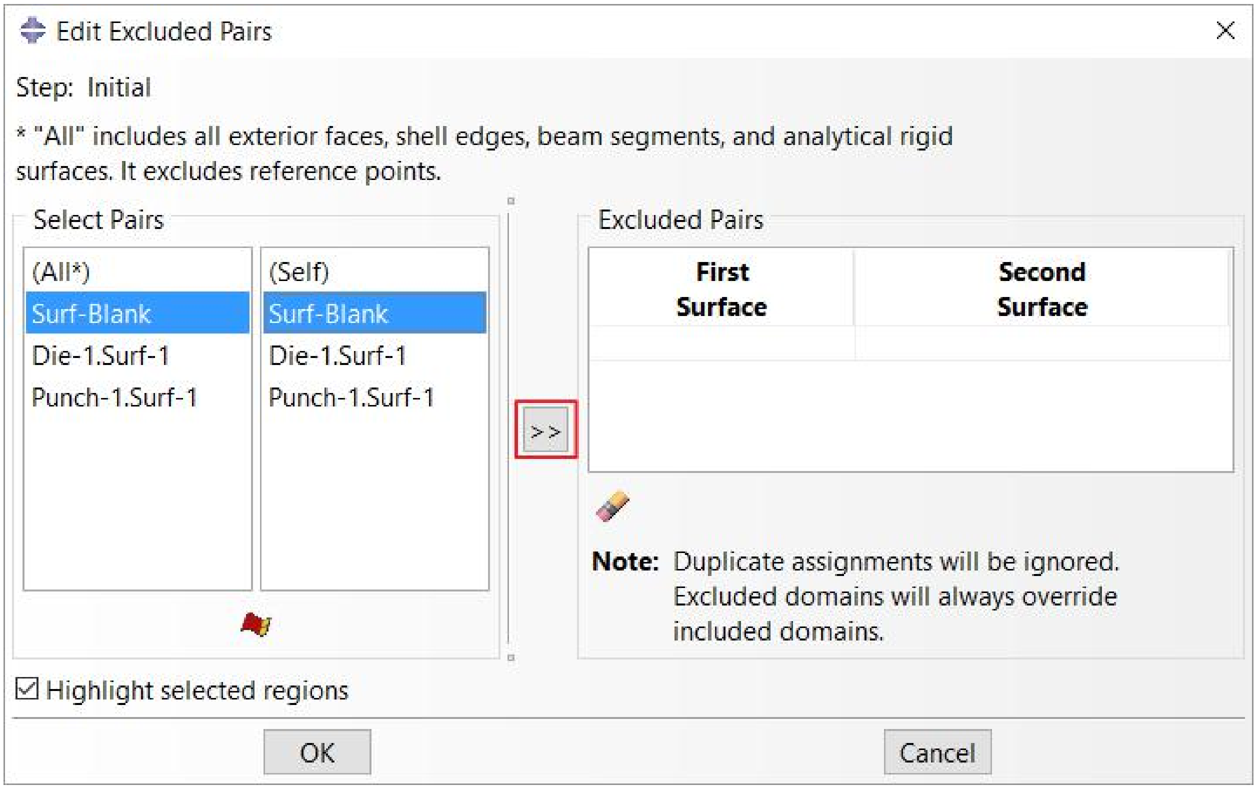
Pick the blank surface as shown below.



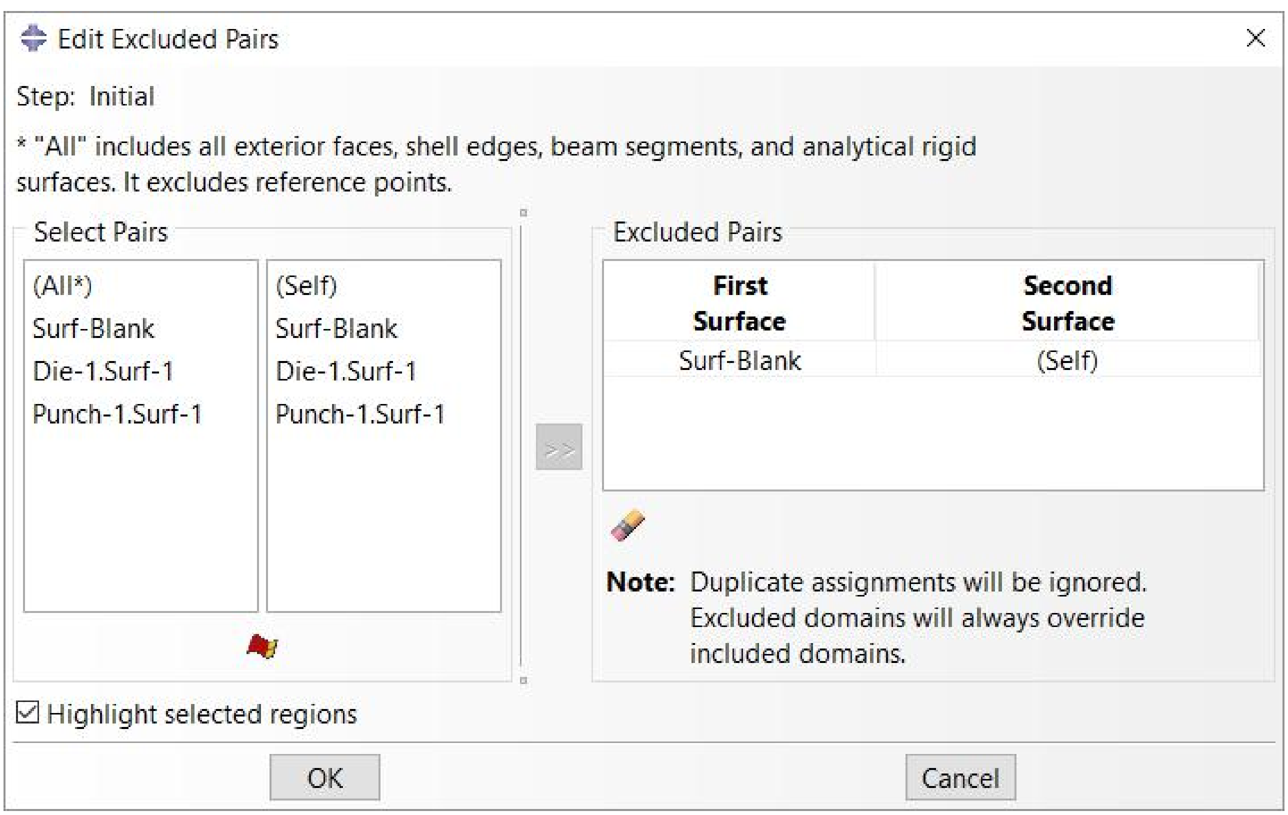
Pick **Done** to proceed.

Pick **Both sides** to include both sides of shell in the surface definition.

Pick **Surf-Blank** both in the first and second columns.



Pick >>> in the dialog box to transfer the selection to the list of excluded pairs.



So the selected surface is excluded from self-contact consideration.

Pick **OK** to apply the changes and exit dialog box.

Pick **OK** and it completes the modification of the interaction.

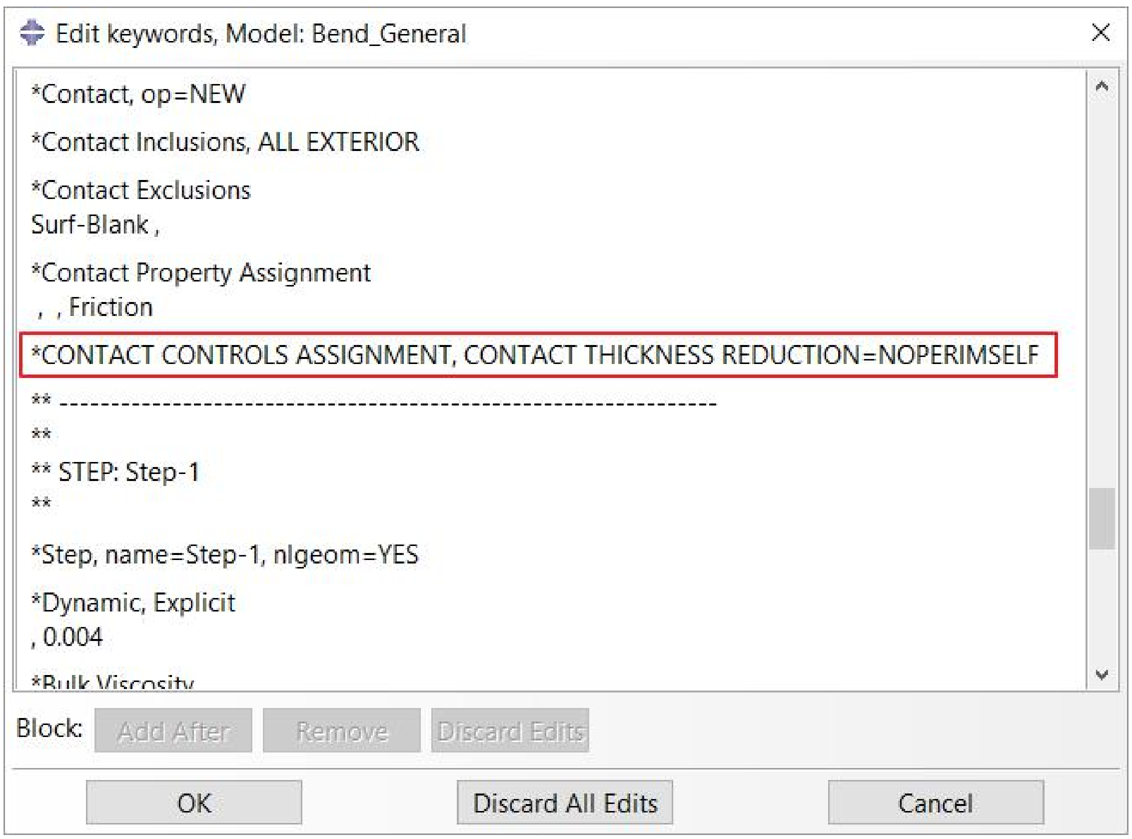
Pick **Dismiss** to close the manager.

Now we will add keywords in the Keywords Editor to eliminate thickness reductions in regions of the model that are excluded from self-contact.

Pick **Model > Edit Keywords > Bend_General** from the main menu and enter the following keywords after the contact property assignment.

*CONTACT CONTROLS ASSIGNMENT, CONTACT THICKNESS REDUCTION=NOPERIMSELF

The keyword property editor is shown below.



These keywords eliminate thickness reductions in regions of the model that are excluded from self-contact and at all shell perimeters.

Pick **OK** to apply and exit.

Now we will resubmit the job.

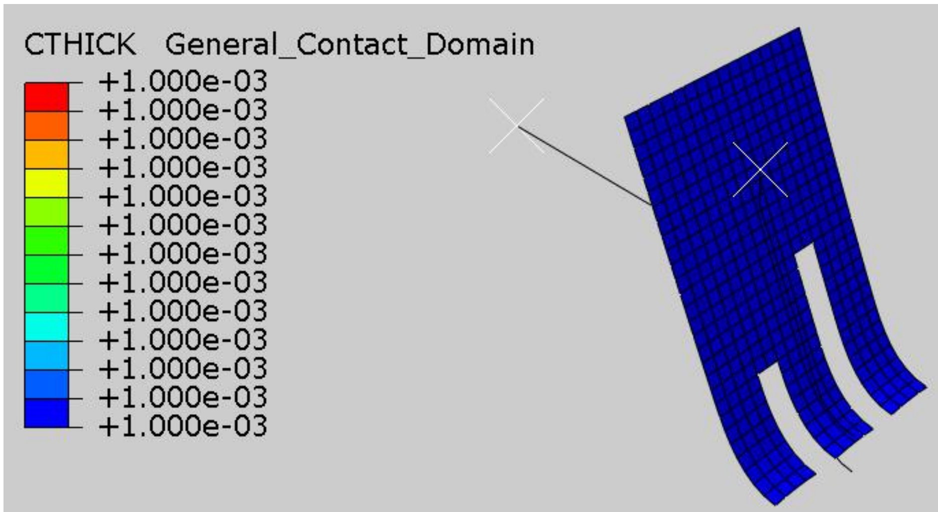
So change to **Job** module and open the Job Manager by picking .

Select VBendGen and pick **Submit > OK.**

After the job is completed, pick **Results** to view the results in the Visualization module.

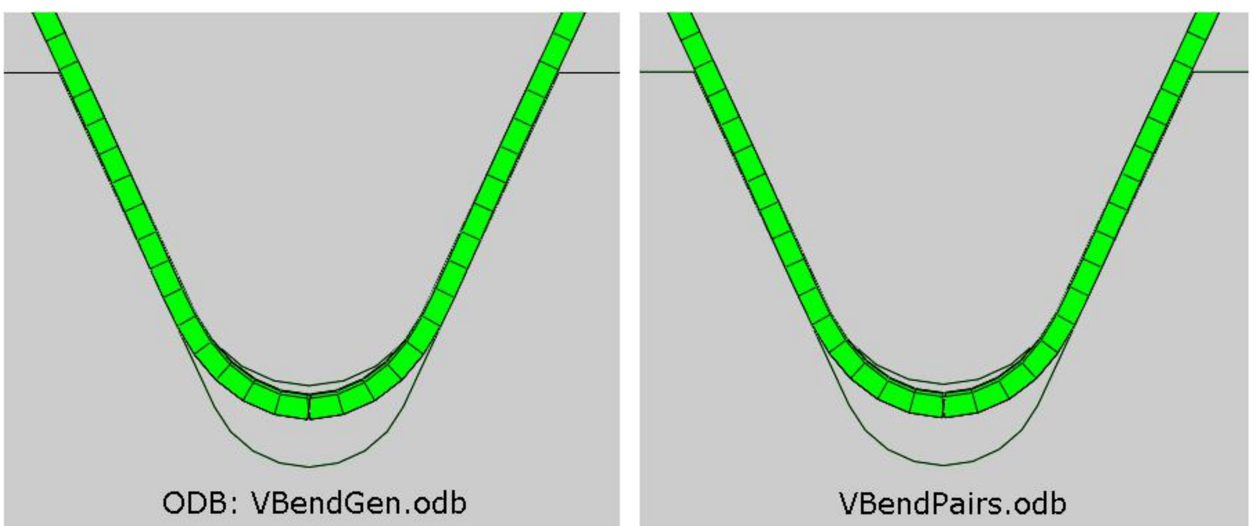
Select the **CTHICK** variable in the Field Output toolbar .

Now the simulation results will appear as shown below.



It can be seen that thickness of the blank remains unaffected.

The following figure compares the deformed shape of blank obtained by using contact pair approach with general contact approach while visualizing the shell thickness.



It can be seen that result are very similar when thickness reduction does not take place in general contact approach.

Note: The general contact and contact pairs algorithms in Abaqus/Explicit use completely separate implementations with many key differences in the designs of the numerical algorithms.

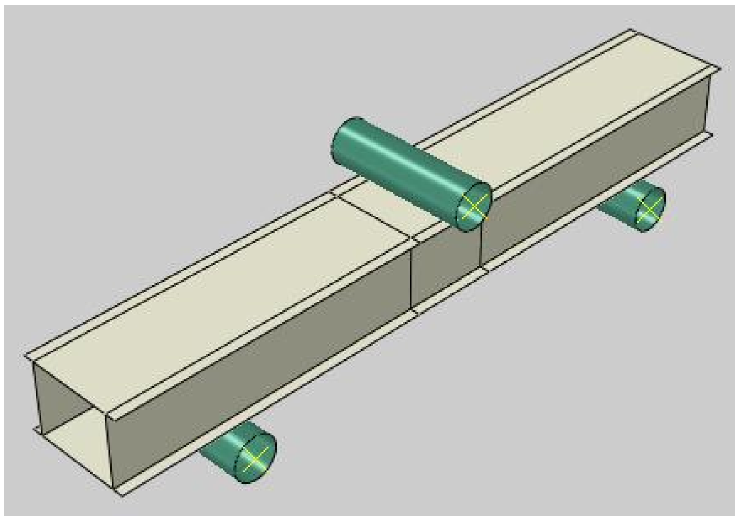
Select **File > Save** to save the changes we made so for.

Exercise 29

In this exercise we will simulate the bending of an extrusion using the Abaqus/Explicit. We will use both the contact pair and the general contact approach.

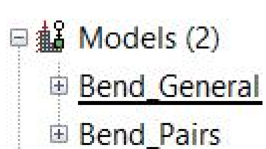
Pick **File > Set Work Directory** and set the work directory to Extrusion folder

Open the model database **Bending.cae**. It will appear as shown below.



The model consists of four three-dimensional parts. Two cylinder located at the bottom of extrusion act as supports while the third cylinder, located at top of the extrusion, acts as punch. The punch and supports are modeled as analytical rigid parts. The extrusion is assumed to be made from an alloy with a Young's modulus of 70 GPa and a Poisson's ratio of 0.31. A coefficient of friction of 0.2 is assumed for the self-contact of extrusion and 0.05 for the contact between the rigid cylinders and the extrusion. The extrusion is meshed with linear, quadrilateral shell elements with reduced integration (S4R).

In the model tree, it can be seen that there are two models. "Bend_General" model will be used to perform analysis using the general contact approach and "Bend_Pairs" model will be used to perform analysis using the contact pairs.



Analysis Steps

The analysis will be performed in one step under quasi-static loading conditions. In this step the punch is moved downwards by applying a boundary condition. A time period of 0.05 s has been specified for this step while the punch is moved by a magnitude of 0.125 in the y-direction.

Boundary Conditions

Boundary conditions required for the analysis have already been defined. The punch is moved by a magnitude of 0.125 in the y-direction with the "Punch" boundary condition. A smooth amplitude curve has been specified for this boundary condition so that displacement loading takes place with smooth acceleration. This makes sure that there is no sudden impact load onto the extrusion at the start of the analysis.

General Contact Approach

Now we will create a general contact interaction to define contact for the entire model.

First make sure that "Bend_General" model is active as shown below.

Change to **Interaction** module and open the Interaction Manager by picking .

Pick **Create** to define a new interaction.

Enter **All_General** as the name of the interaction.

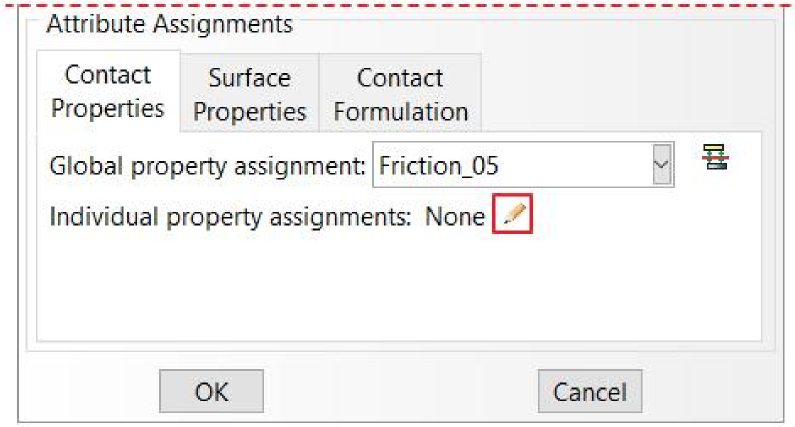
Pick **Initial** in the Step field.

Select **General contact** and pick **Continue** to proceed.

Pick **Friction_05** as the global contact interaction property.

"Friction_05" interaction property defines a friction coefficient of 0.05. As we intend to specify a friction coefficient of 0.2 for the self contacting surfaces of extrusion, we will define an individual property assignment. Individual property assignments are used to assign different contact properties to individual surface pairs.

Pick  to create individual property assignment.

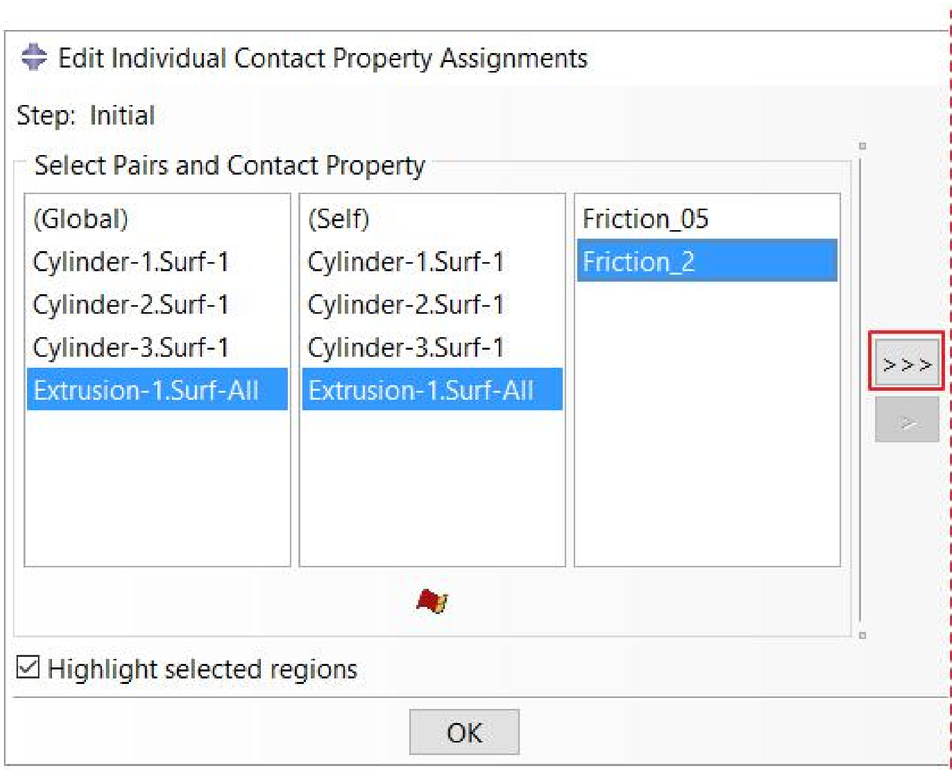


The surface required to specify the property assignment have been defined beforehand.

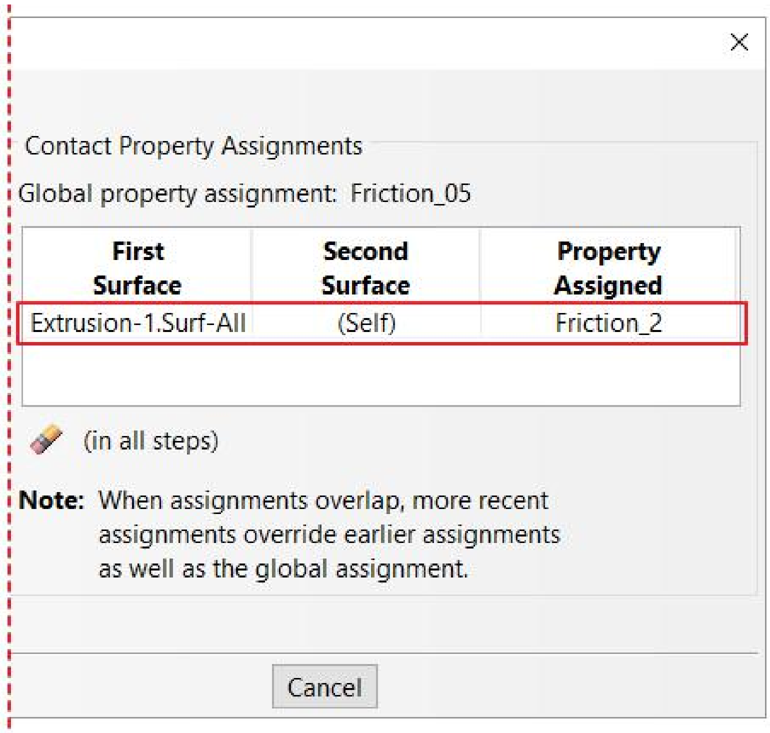
Pick **Extrusion-1.Surf-All** both in the first and second column to define the surface pairings.

In the third column, select the “Friction_2” as the contact property.

Pick >>> to transfer the selection to the list of contact property assignments.



Notice that system has assigned the property for the self-contact of selected surface.



To assign a property for self-contacting surface, select either the same surface name or (Self) in the second column.

Pick **OK** to apply the changes and exit dialog box.

Notice that Edit Interaction dialog box shows the number of individual property assignments to 1.

Pick **OK** and it completes the definition of interaction.

Pick **Dismiss** to close the manager.

Contact Pair Approach

Make sure that "Bend_Pairs" model is active as shown below.

Three contact pair interactions, "Cylinder1-Extrusion", "Cylinder2-Extrusion" and "Punch-Extrusion" have already been defined. These interactions define contact between the analytical rigid parts and the extrusion. Now we will create another interaction defining the self-contact of extrusion surfaces.

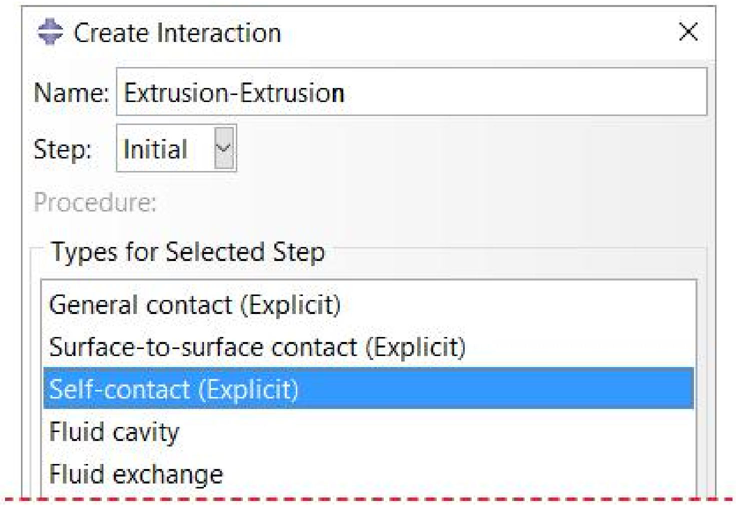
Change to **Interaction** module and open the Interaction Manager by picking .

Pick **Create** to define a new interaction.

Enter **Extrusion-Extrusion** as the name of the interaction.

Pick **Initial** in the Step field.

Select **Self-contact** as type.



A self-contact interaction allows to define contact between different areas of a single surface.

Pick **Continue** to proceed.

A surface has already been defined, so pick **Surfaces** on the right side of the prompt area and select the Extrusion-1.Surf-All.

Self-contact models contact interaction between a single surface and itself by specifying only a single surface.

Pick **Continue** and Edit Interaction dialog box will appear.

Pick **Friction_05** as the contact interaction property.

Pick **OK** and it completes the definition of interaction.

Pick **Dismiss** to close the manager.

Job Submission

All the information required for analysis has already been set up in the model. Now we can submit the job for analysis.

So change to **Job** module and open the Job Manager by picking .

Pick **Create** and create a job named ExtrusionGen for the Bend_General model .

Pick **Continue** and then **OK**.

Pick **Submit** to submit the job for analysis and notice that job completes successfully.

Also submit a job for the Bend_Pairs model, named ExtrusionPairs. Notice that job is aborted due to errors.

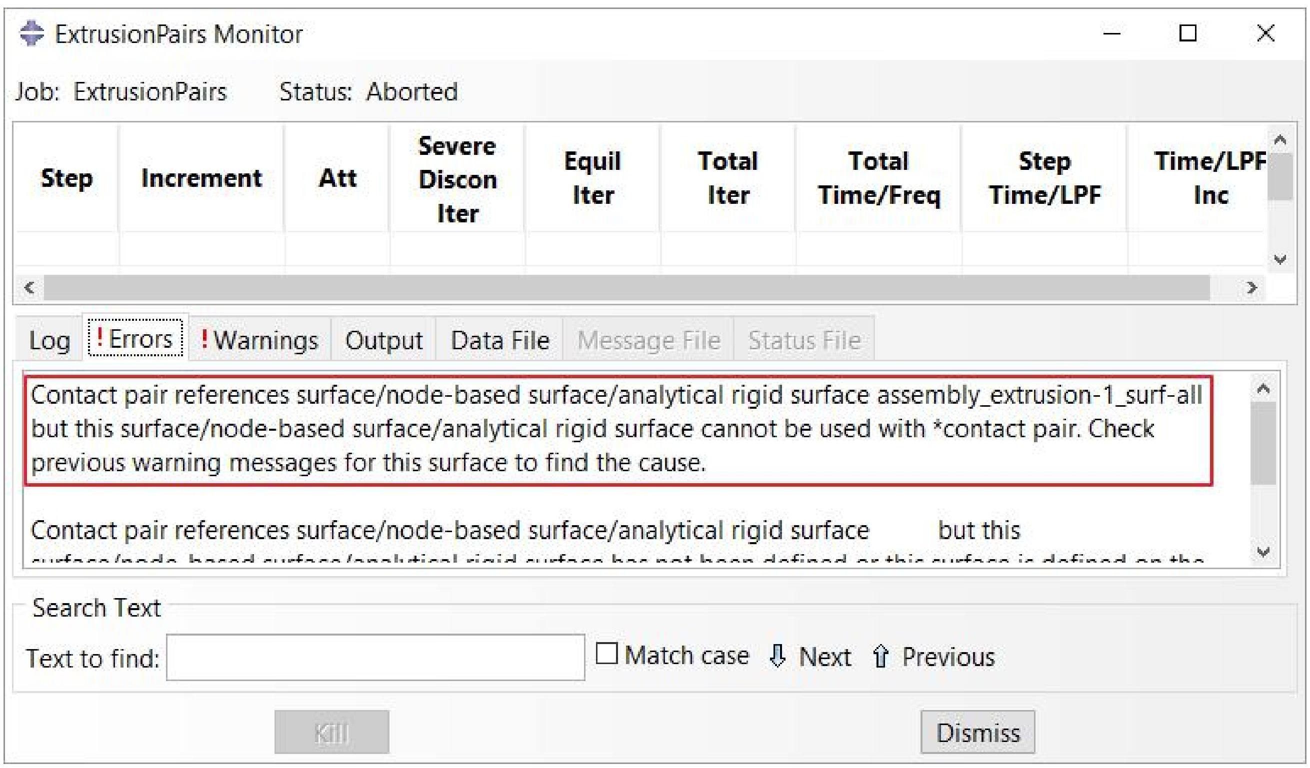
Diagnosing the error

We will start the investigation of the error by using job monitor.

So change to **Job** module and open the Job Manager by picking .

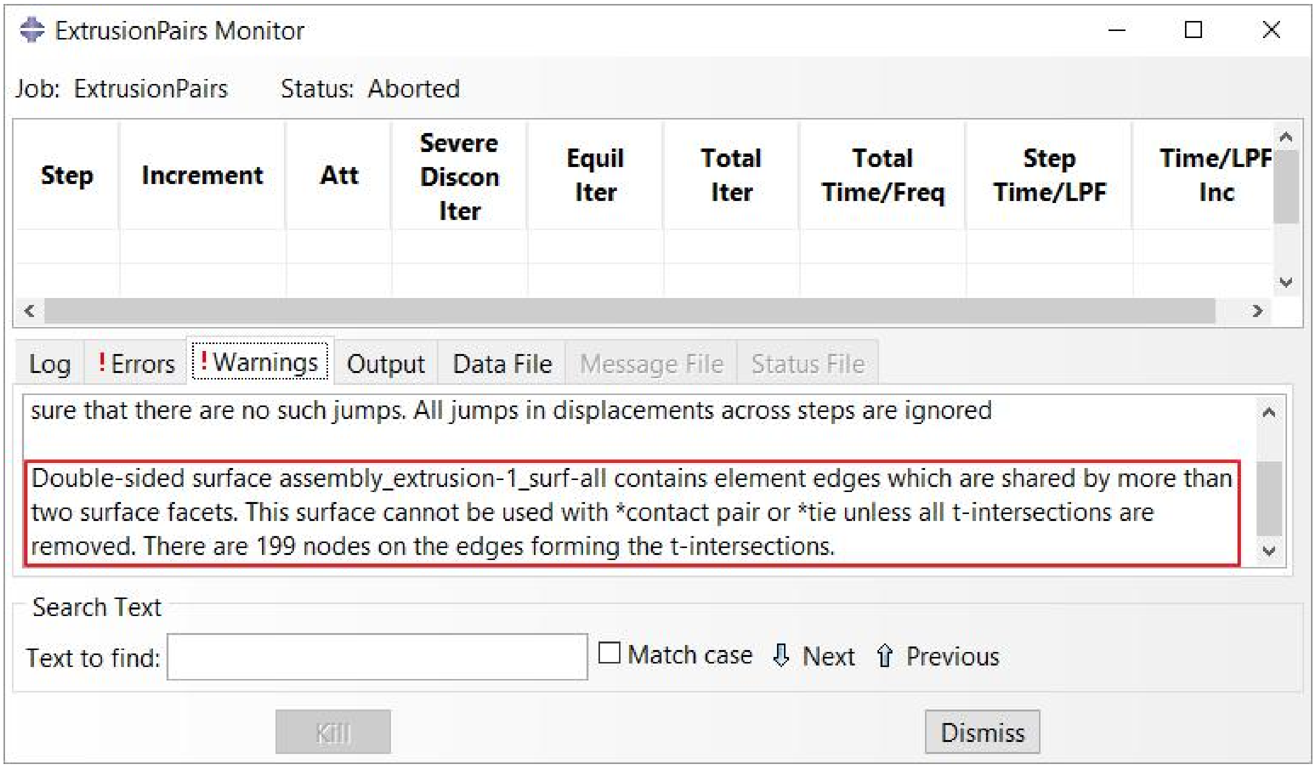
Now select ExtrusionPairs job in the manager and pick **Monitor**

An error message appears in the Errors tab about the contact definition as shown in the figure below.



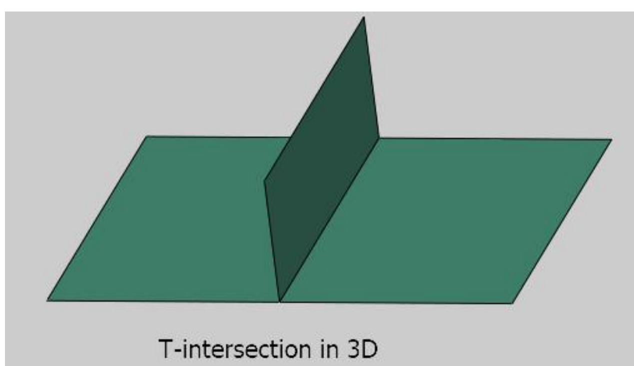
From this error message, it is not clear what is the problem in the definition of contact interaction. So we will look at the warning messages as suggested in the error message.

Pick the **Warnings** tab and notice the warning about t-intersections in the selected surface as highlighted below.



So the analysis is aborted due to presence of t-intersections in the Extrusion-1.Surf-All surface.

The contact pair algorithm in Abaqus/Explicit does not allow T-intersections in shells. In a self-contact interaction, self-contacting surface act as both master and slave surfaces, therefore, if a restriction applies to contact pair, it also applies to self-contact.


T-intersection in 3D

Modifying Contact Interaction

To avoid this error, we will not use a self-contact interaction. Instead contact pair interactions will be defined in such a way that no t-intersection occurs in the surface definition.

Change to **Interaction** module and open the Interaction Manager by picking .

Select **Extrusion-Extrusion** and pick **Delete**.

To create a new interaction, pick **Create**.

Enter **Top-Right** as the name of the interaction.

Pick **Initial** in the Step field.

Select **Surface-to-surface contact** as type.

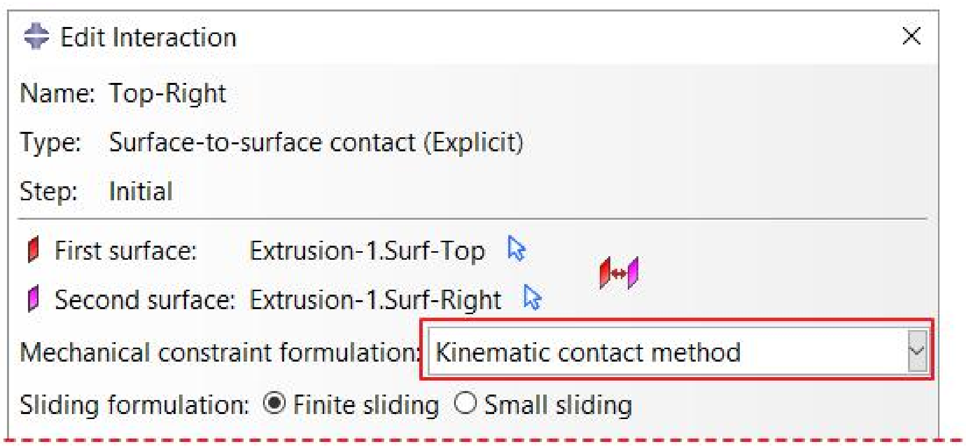
Pick **Continue** to proceed.

We have already defined a surface so pick **Surfaces** on the right side of the prompt area and select the **Extrusion-1.Surf-Top**.

Pick **Continue** to proceed.

For the slave type, pick **Surface** and select the **Extrusion-1.Surf-Right**.

Pick **Continue** to proceed and Edit Interaction dialog box will appear. In this dialog box it can be seen that Kinematic contact method is selected by default



The kinematic contact algorithm uses a kinematic predictor/corrector contact algorithm to strictly enforce contact constraints and no penetrations are allowed. In addition to kinematic contact method, Abaqus/Explicit provides penalty contact method which enforces contact constraints weakly. The penalty contact algorithm results in less stringent enforcement of contact constraints than the kinematic contact algorithm.

Pick **Friction_2** as contact interaction property.

Pick **OK** and it completes the definition of interaction.

Similarly create Top-Left, Top-Bottom, Bottom-Right and Bottom-Left interactions where corresponding surfaces have already been defined.

After creating all these interactions, we can resubmit the job.

So change to **Job** module and open the Job Manager by picking .

Select ExtrusionPairs and pick **Submit > OK** and notice that job completes successfully.

Pick **Dismiss** to close the manager.

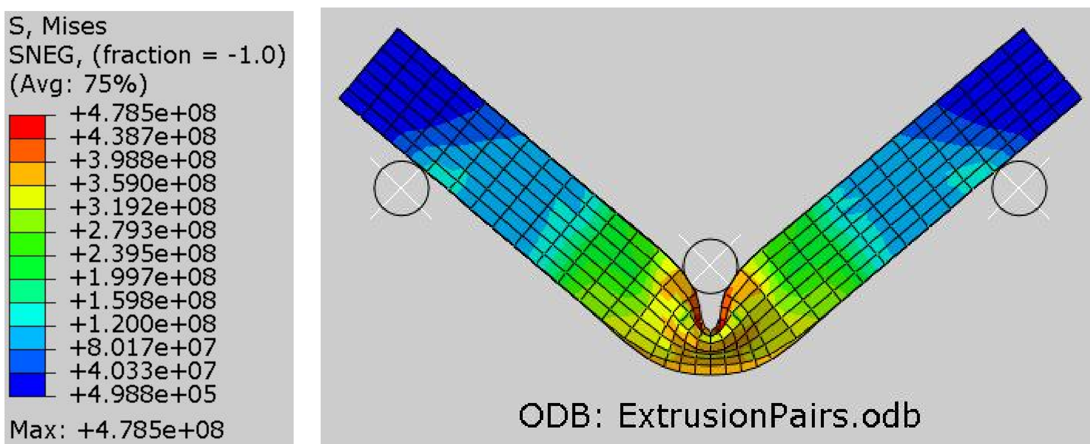
Comparing the Results

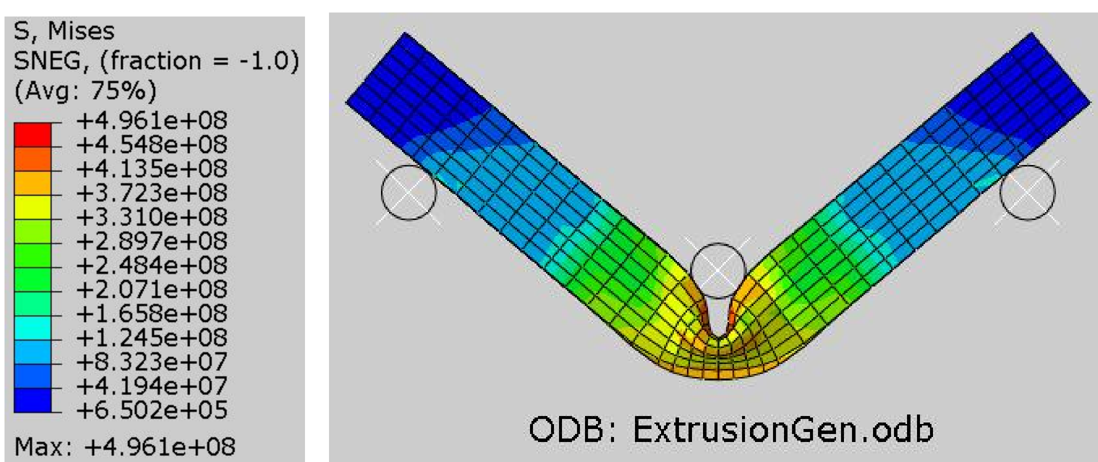
Now we will compare the simulation results obtained by using contact pair approach with general contact approach.

Pick  to plot the contours on deformed shape.

Select Mises in the Field Output toolbar if not selected by default.

The following figure compares the distribution of von Mises stresses.

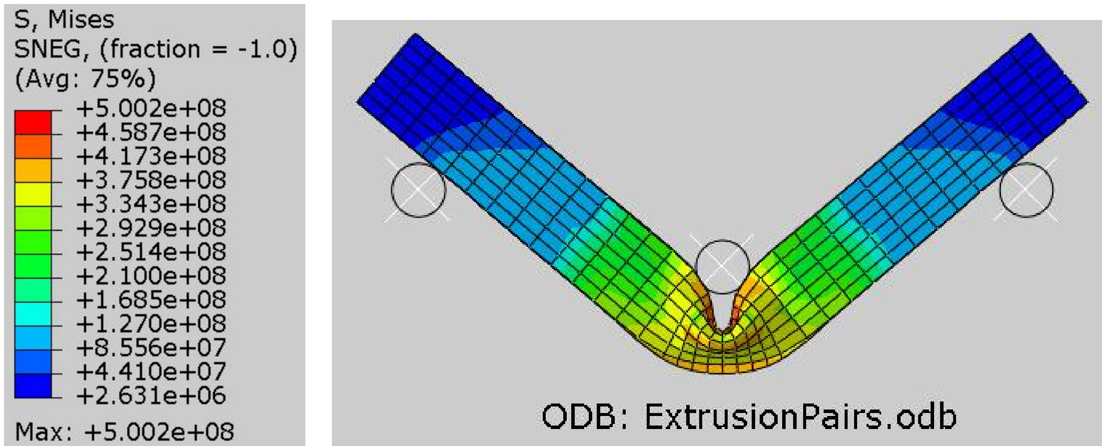




The figures show that results are not similar for both cases. This is due to the fact that the general contact algorithm in Abaqus/Explicit uses penalty enforcement method while the contact pair interactions uses kinematic contact method by default.

General contact does not support kinematic constraint enforcement so for comparison purposes, we will change the contact constraints enforcement method to "Penalty contact method".

The following figure shows the distribution of von Mises stresses when penalty contact method is used for all the contact pair interactions in the Bend_Pairs model.



It can be seen that results are very close to those obtained with general contact interaction.

It can be concluded that general contact greatly simplifies the process of contact definition and has only few limitations.

Select **File > Save** to save the changes we made so for.

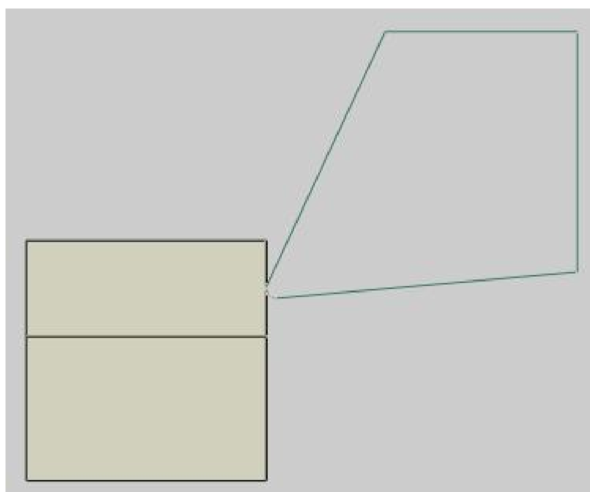
Exercise 30

In this exercise we will simulate the metal cutting using the contact pair approach in Abaqus/Explicit.

Abaqus provides element deletion functionality to model erosion due to material failure. Material failure refers to the complete loss of load-carrying capacity that results from progressive degradation of the material stiffness. A material model can contain multiple failure mechanism acting simultaneously on the specified elements. Once the material stiffness is fully degraded, elements can be removed from the calculations. We will use this functionality to simulate metal cutting.

Pick **File > Set Work Directory** and set the work directory to Tool folder

Open the model database **Machining.cae**. It will appear as shown below.



The model consists of two two-dimensional parts: tool and workpiece . The tool is modeled as an analytical rigid part. The workpiece is assumed to be made from steel with a Young's modulus of 205 GPa and a Poisson's ratio of 0.3. The material definition also includes a failure model. Abaqus/Explicit removes the elements from the mesh as they fail. A coefficient of friction of 0.1 is assumed for the contacting surfaces. The workpiece is meshed with bilinear plane strain quadrilateral elements with reduced integration (CPE4R).

Analysis Steps

The analysis will be performed in a dynamic, explicit step under quasi-static loading conditions. In this step the tool is moved into the workpiece material by applying a boundary condition. A time period of 0.004 s has been specified for this step while the punch is moved by a magnitude of 0.004 in the x-direction.

Boundary Conditions

Boundary conditions required for the analysis have already been defined. The punch is moved by a magnitude of 0.004 in the x-direction with the "Tool" boundary condition. A smooth amplitude curve has been specified for this boundary condition so that displacement loading takes place with smooth acceleration.

Contact Interaction

We will use a node-based surface as slave to model surface erosion. We can use either the general contact or contact pair algorithm for that purpose. It is quite easy to use contact pair interaction for a node-based surface so we will create a contact pair interaction to define the contact between tool and workpiece.

Change to **Interaction** module and open the Interaction Manager by picking .

Pick **Create** to define a new interaction.

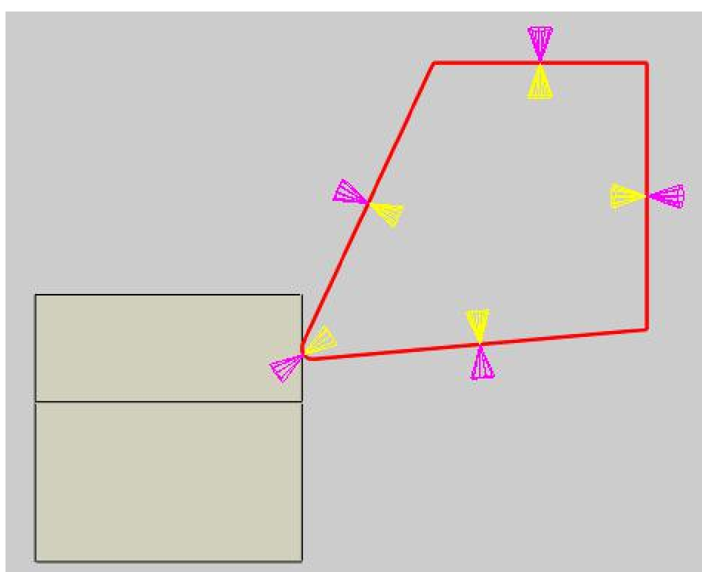
Enter **Tool-Workpiece** as the name of the interaction.

Pick **Initial** in the Step field.

Select **Surface-to-surface contact** as type.

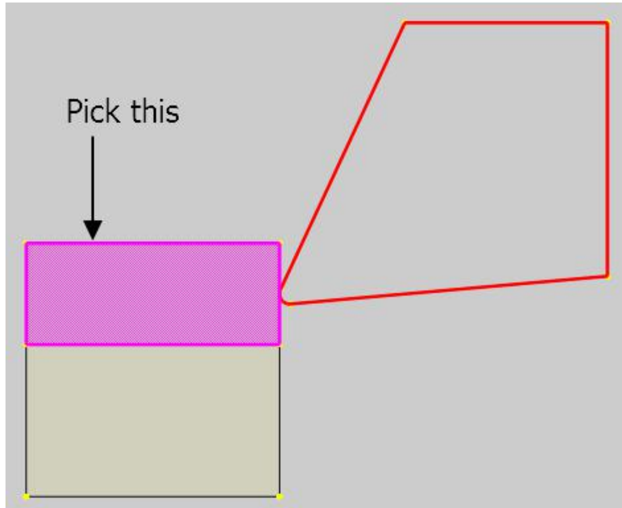
Pick **Continue** and select the rigid surface as master.

Pick **Done** and select **Magenta** for the side of surface in contact.



As a surface can contact on either side, so it is important to specify the desired side of the surface to be in contact. It is done by choosing the color associated with the desired side.

For the slave type, pick **Node Region** and select the region highlighted in the figure below.



Pick **Done** and Edit Interaction dialog box will appear.

Pick **Friction** as the contact interaction property.

Pick **OK** and it completes the definition of interaction.

Pick **Dismiss** to close the manager.

Job Submission

All the information required for analysis has already been set up in the model. Now we can submit the job for analysis.

So change to **Job** module and open the Job Manager by picking .

Pick **Create** and create a job named MetalCutting.

Pick **Continue** and then **OK**.

Pick **Submit** to submit the job for analysis and notice that job completes successfully.

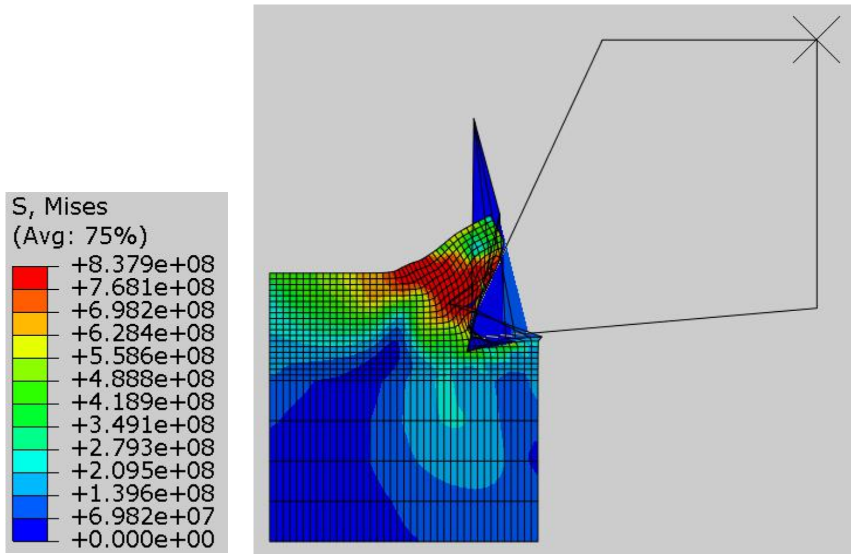
Postprocessing

Select MetalCutting job in the manager and pick **Results**

Pick  to plot the contours on deformed shape.

Select Mises in the Field Output toolbar if not selected by default.

The following figure shows the distribution of von Mises stresses after completion of 0.0016 s of step time.



It appears that some of the elements are distorting extensively. In reality these are the failed elements which have not been removed from the display although they have been removed for computation purposes. We will request the STATUS output variable so that failed elements can be excluded from the display.

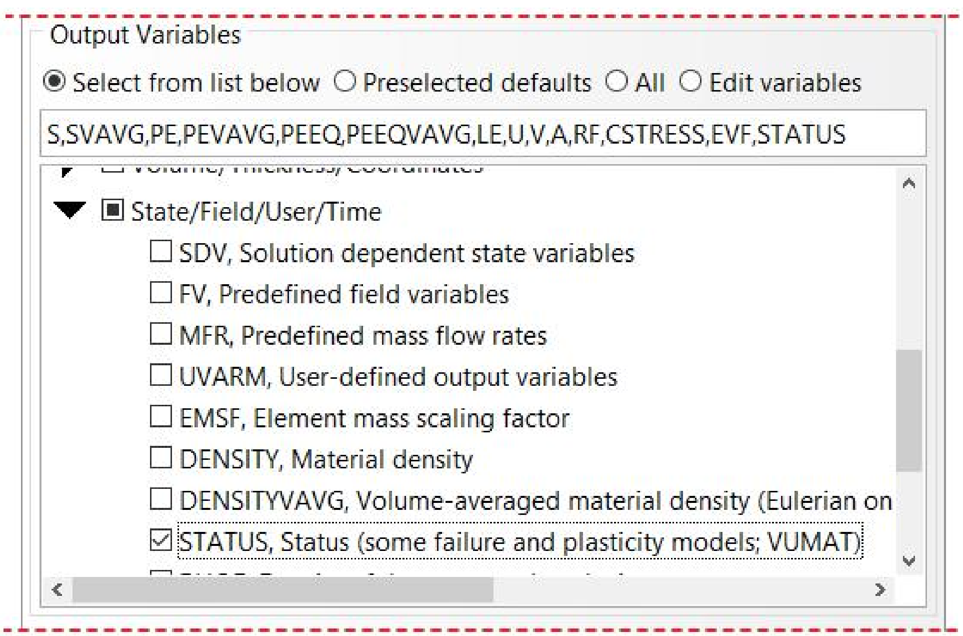
Field Output Request

Change to **Step** module.

Open the Field Output Manager by picking  .

Select the F-Output-1 field and pick **Edit**.

Check the **STATUS** variable (located under the State/Field/User/Time container).



Pick **OK** to apply and exit.

Pick **Dismiss** to close the manager.

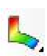
Now we will resubmit the job.

So change to **Job** module and open the Job Manager by picking .

Select MetalCutting and pick **Submit > OK.**

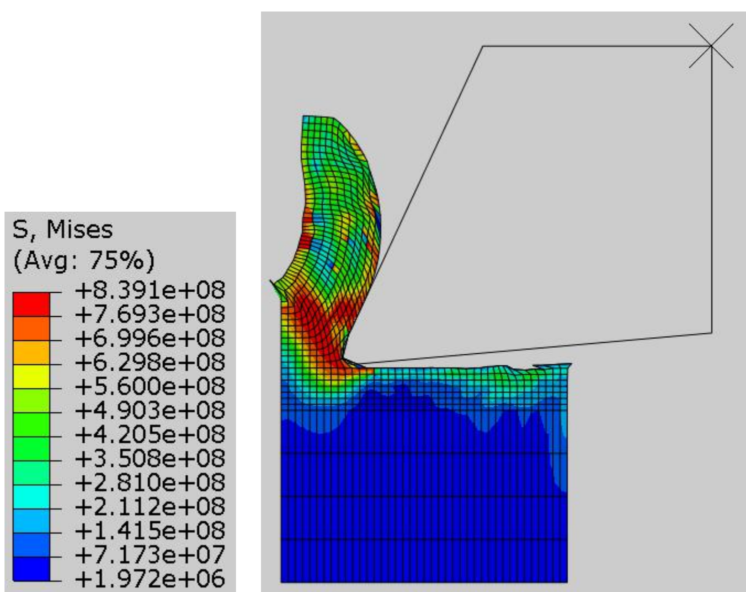
After the job is completed, pick **Results** to view the results in the Visualization module.

Postprocessing

Pick  to plot the contours on deformed shape.

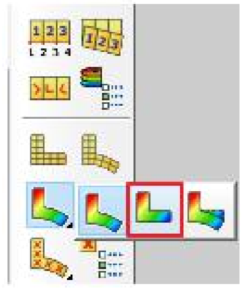
Select Mises in the Field Output toolbar if not selected by default.

The following figure shows the distribution of von Mises stresses after completion of the simulation. It can be seen that no failed element appears in the display.



It is also of interest to know which elements have failed in the undeformed configuration. So first we will plot the stress contours on undeformed shape and then remove the failed elements from display.

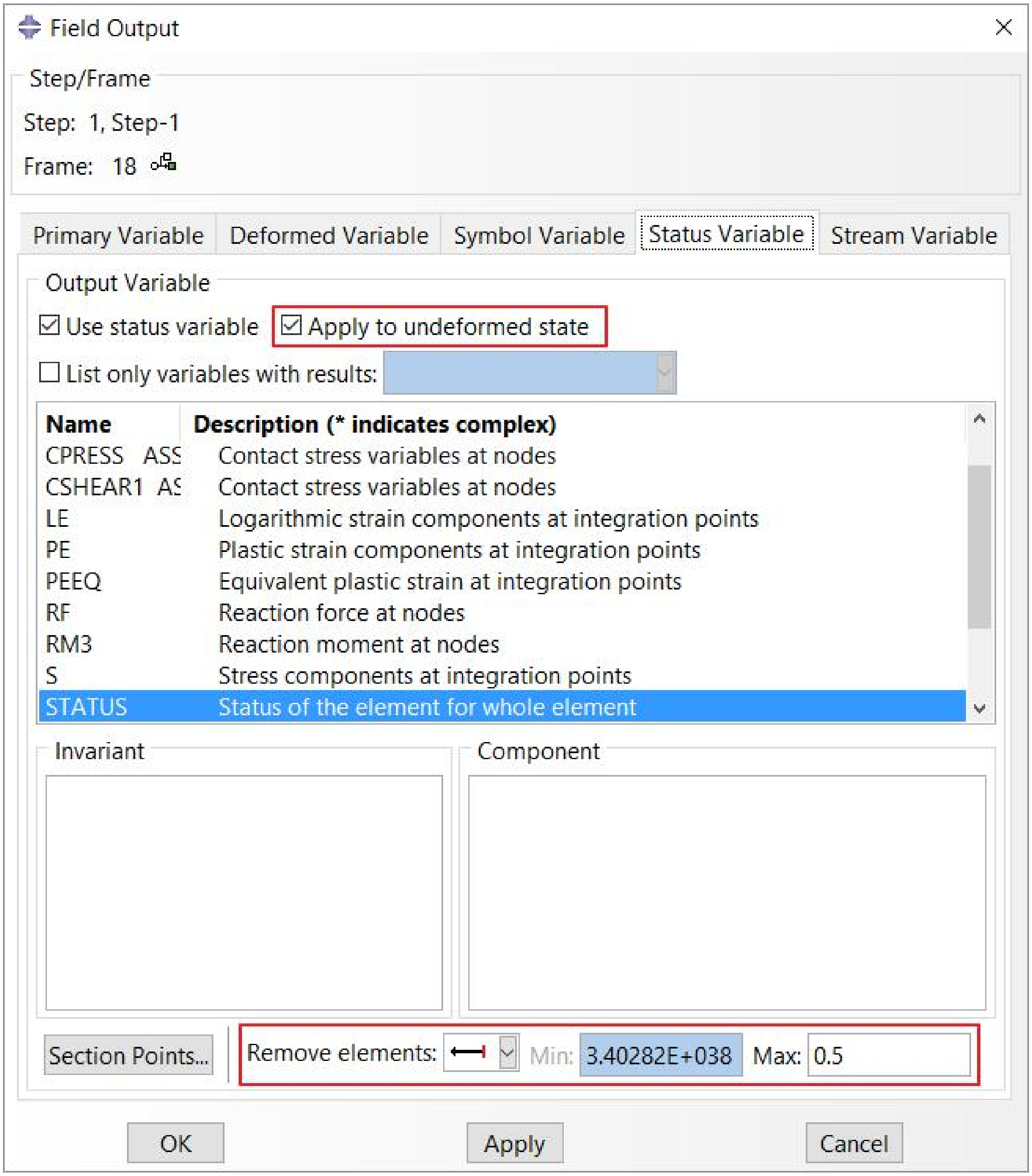
Pick  to plot contours on undeformed shape.



From the main menu bar, select **Result > Field Output** and Field Output dialog box will appear.

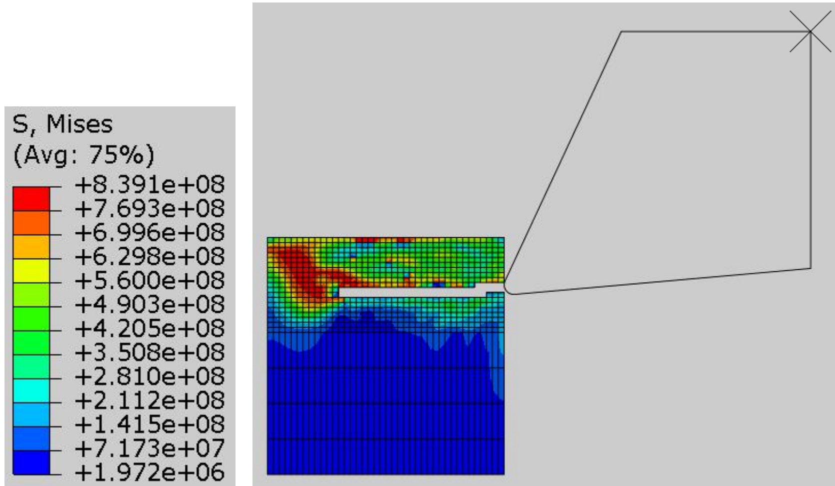
Click the **Status Variable** tab and notice that system removes elements which have a STATUS value of less than 0.5 but this is not applied to undeformed state by default.

Check the **Apply to undeformed state** option as shown below.



Pick **OK** to apply and exit.

The following figure shows the distribution of von Mises stresses in the undeformed shape at the end of the simulation. It can be seen that failed elements do not appear in the display.



ALE Adaptive Meshing

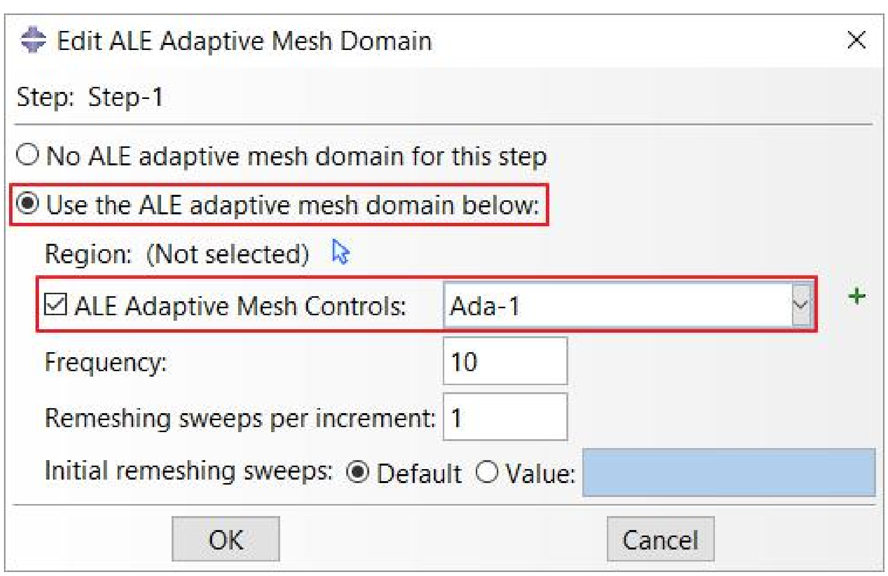
In the contour plots on the deformed shape, it can be seen that mesh quality is not satisfactory for the analysis. To maintain a high-quality mesh throughout the analysis, we will introduce ALE adaptive meshing technique.

So change to **Step** module.

From the main menu bar, select **Other > ALE Adaptive Mesh Domain > Edit > Step-1** and Edit ALE Adaptive Mesh Domain dialog box will appear.

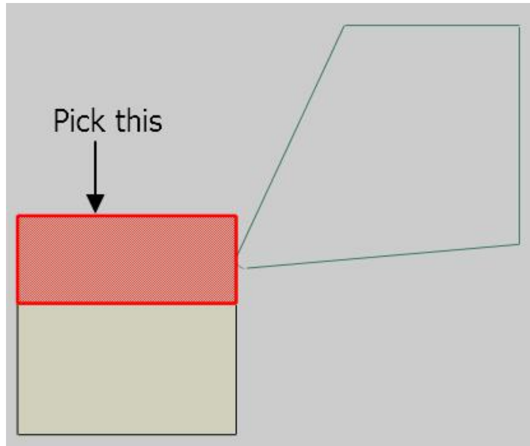
Select the **Use the ALE adaptive mesh domain below** option.

Activate the **ALE Adaptive Mesh Controls** option as shown below.



To define the region for ALE adaptive mesh, pick .

Select the region highlighted in the figure below and pick **Done**.



Pick **OK** to apply and exit.

Now we can submit the job for analysis.

So change to **Job** module and open the Job Manager by picking .

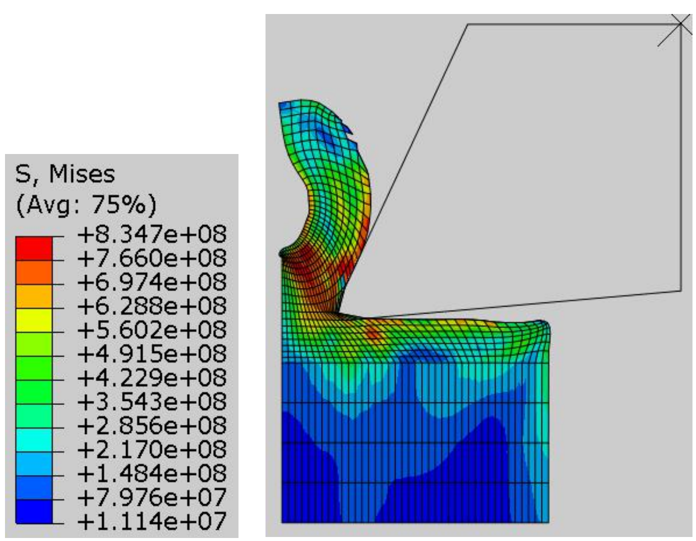
Pick **Create** and create a job named MetalCuttingALE.

Pick **Continue** and then **OK**.

Pick **Submit** to submit the job for analysis and notice that job completes successfully

After the job is completed, pick **Results** to view the results in the Visualization module.

The following figure shows the distribution of von Mises stresses after completion of the simulation.



The results obtained with ALE adaptive meshing show improved mesh quality.

Select **File > Save** to save the changes we made so for.

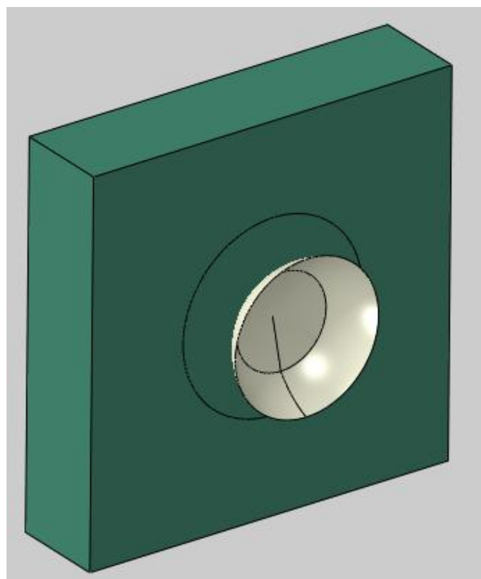
Exercise 31

In this exercise we will simulate the erosion of a plate due to impact of a projectile using the general contact approach in Abaqus/Explicit.

The material definition for the plate contains multiple failure models. So the Abaqus removes elements from the mesh as they fail due to the impact of the projectile.

Pick **File > Set Work Directory** and set the work directory to Projectile folder

Open the model database **RigidProjectile.cae**. It will appear as shown below.



The model consists of two three-dimensional parts: plate and projectile. The projectile is modeled as an analytical rigid part. The plate is assumed to be made from an alloy with a Young's modulus of 70 GPa and a Poisson's ratio of 0.33. The material definition also includes ductile and shear damage models. Abaqus/Explicit removes the elements from the mesh as they fail. The plate is meshed with 8-node, brick elements with reduced integration (C3D8R).

⇨ Analysis Steps

The analysis will be performed in one step under dynamic loading conditions. In this step the projectile moves into the plate with an initial velocity. A time period of 0.000015 s has been specified for this step.

⇨ Boundary Conditions

Boundary conditions required for the analysis have already been defined. The motion of projectile is restrained along all directions except the x-axis. A predefined field has been defined to specify an initial velocity of 700 m/s for the projectile along x-axis. Top and bottom faces of the plate are restrained to move in any direction.

Contact Interaction

We will use an element-based surface as slave to model surface erosion. This surface will include all faces of the elements which could come into contact with the projectile. This is necessary so that the surface topology evolves to match the exterior of elements that have not failed. For element-based surface, exterior faces are initially active, and interior faces are initially inactive. Once an element fails, its faces are removed from the contact domain, and any interior faces that have been exposed are activated. Contact pair algorithm does not allow element-based surface so we will define a general contact interaction to model surface erosion.

An element set "ElemErode" has already been defined which contains all the elements in the plate that could come into contact with the impacting projectile. We will create a surface containing all the interior faces of these elements.

Change to **Interaction** module and open the Interaction Manager by picking .

Pick **Create** to define a new interaction.

Enter **All_General** as the name of the interaction.

Pick **Initial** in the Step field.

Select **General contact** and pick **Continue** to proceed.

Pick **NoFric** as the global contact interaction property.

Notice that **All* with self** is selected by default. This is not sufficient because we also need to include the interior faces of the elements in the plate to define the contact domain. Due to limitations of Abaqus/CAE, we will do so by editing the input file.

Pick **OK** and it completes the definition of interaction.

Pick **Dismiss** to close the manager.

Field Output Request

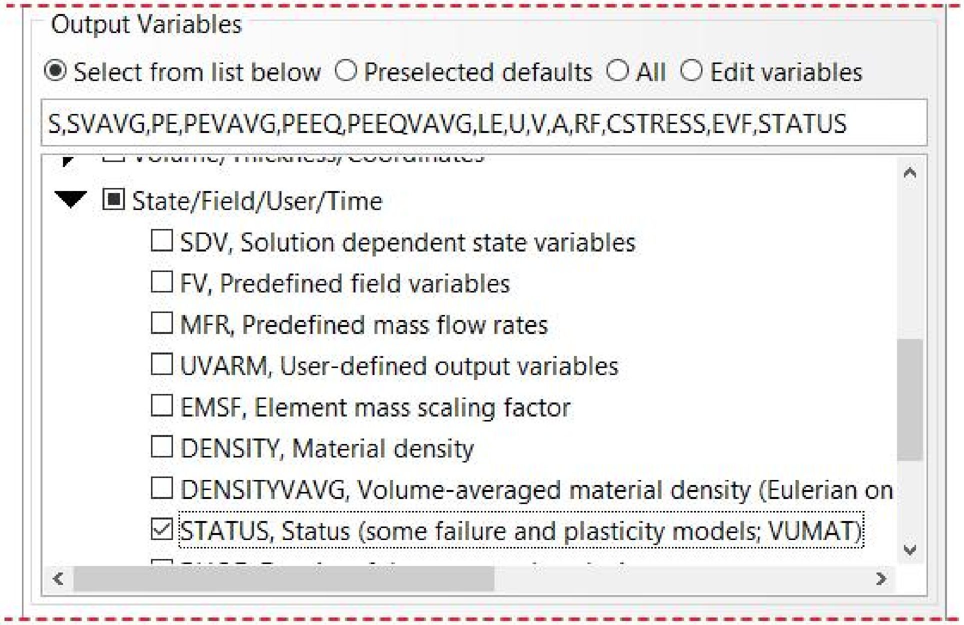
We will request the STATUS output variable so that failed elements can be excluded from the display.

Change to **Step** module.

Open the Field Output Manager by picking  .

Select the F-Output-1 field and pick **Edit**.

Check the **STATUS** variable (located under the State/Field/User/Time container).



Pick **OK** to apply and exit.

Pick **Dismiss** to close the manager.

Job Submission

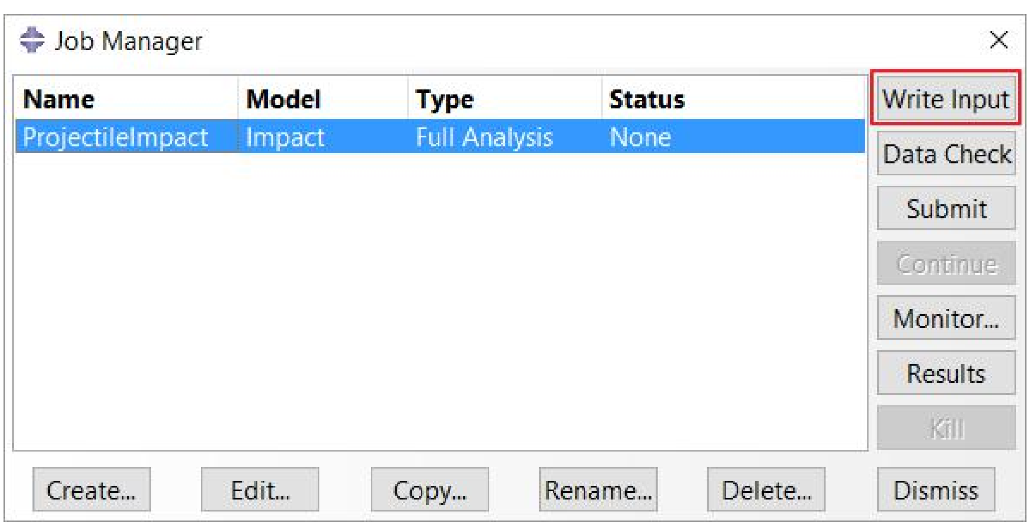
Now we will create a new job and write input file for the analysis.

So change to **Job** module and open the Job Manager by picking .

Pick **Create** and create a job named ProjectileImpact.

Pick **Continue** and then **OK**.

Pick **Write Input** to create the job input file for the analysis.



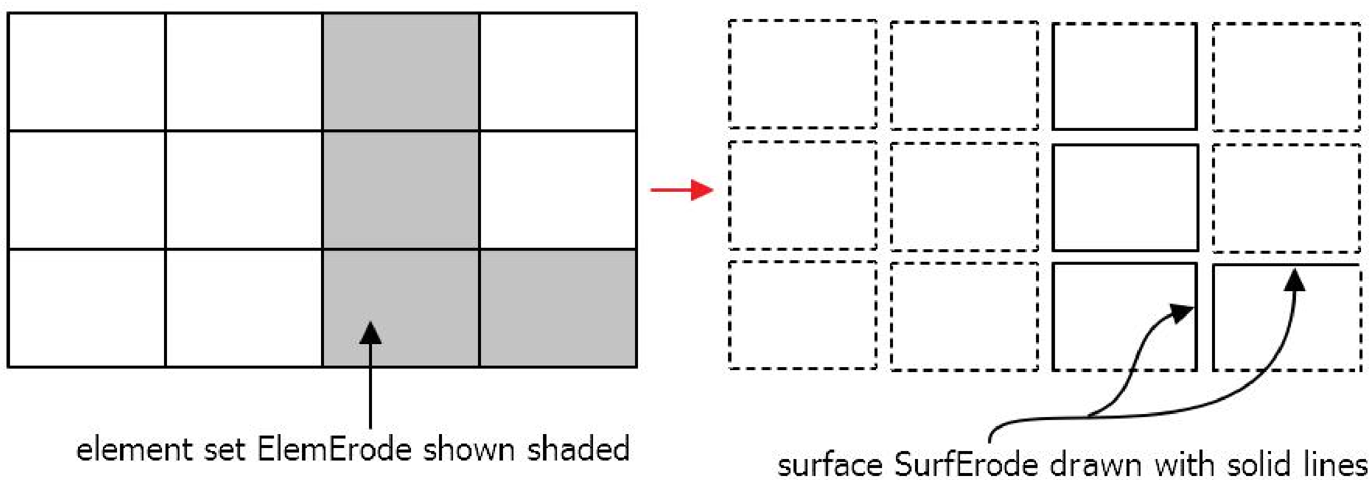
Editing the Input File

Now we will edit the input file using Notepad++ although any other editor can be used for that purpose. Go to the working directory and open the ProjectileImpact.inp

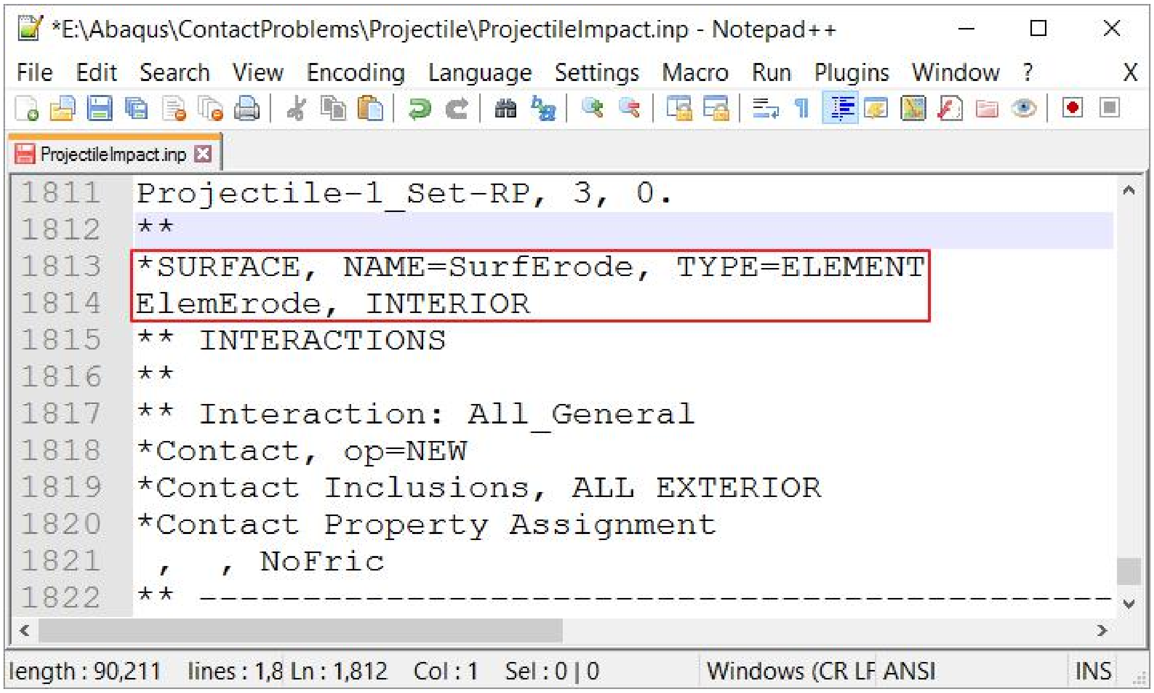
Add the following lines before the definition of contact interaction.

```
*SURFACE, NAME=SurfErode, TYPE=ELEMENT
ElemErode, INTERIOR
```

These lines create a new surface, named "SurfErode", containing all the interior faces of the ElemErode element set. So the faces of the specified elements that are not on the exterior surface of the model will be included in the surface definition. This is shown schematically in the following figure.



After the addition of these lines, input file will appear as shown below.

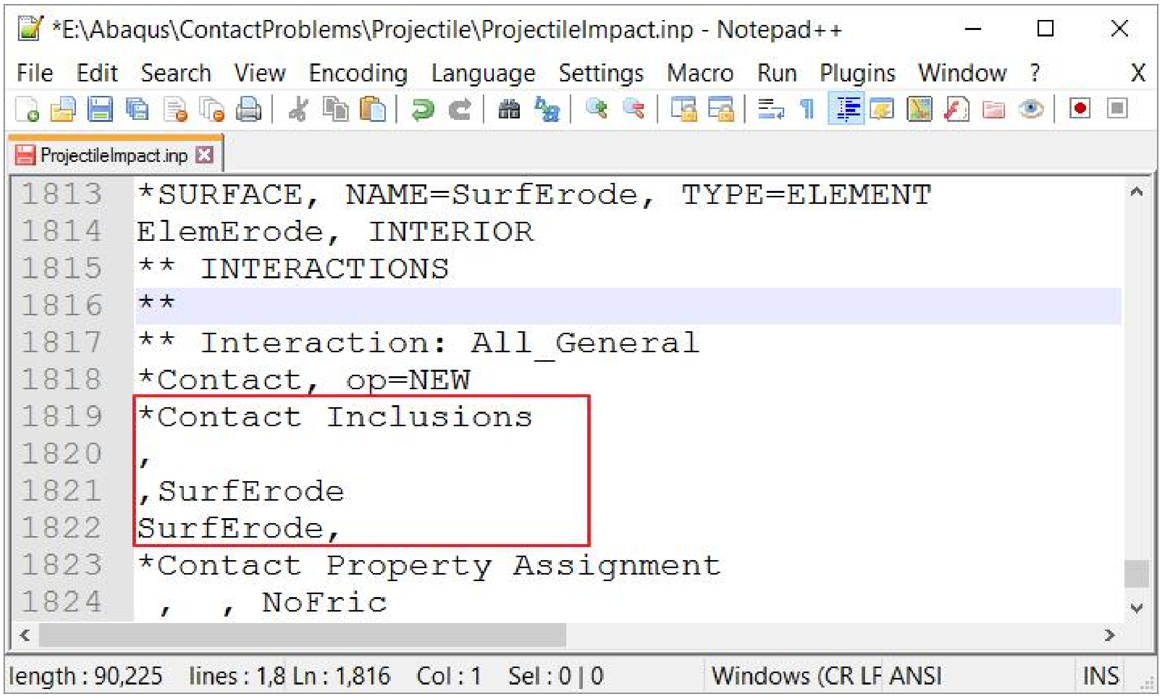


Now we will add data lines for the contact interaction but it is not allowed after the ALL EXTERIOR. So we will delete this keyword and add the following three lines for the contact inclusions

```
,
,SurfErode
SurfErode,
```

First line defines the self-contact for the default all-inclusive surface. In other words, it specifies contact between every exterior face in the model. The second line defines contact between the default all-inclusive surface and SurfErode. In other words, it specifies contact between every exterior face and SurfErode. The third line defines self-contact between the interior faces defined by SurfErode surface.

After the editing and addition of these lines, input file will appear as shown below.



Pick **File > Save** to save the changes in the input file.

Now open the Abaqus command window and set the directory to the folder where input file is located.

Enter following to run the analysis.

abaqus job=ProjectileImpact interactive

Notice that job completes successfully.

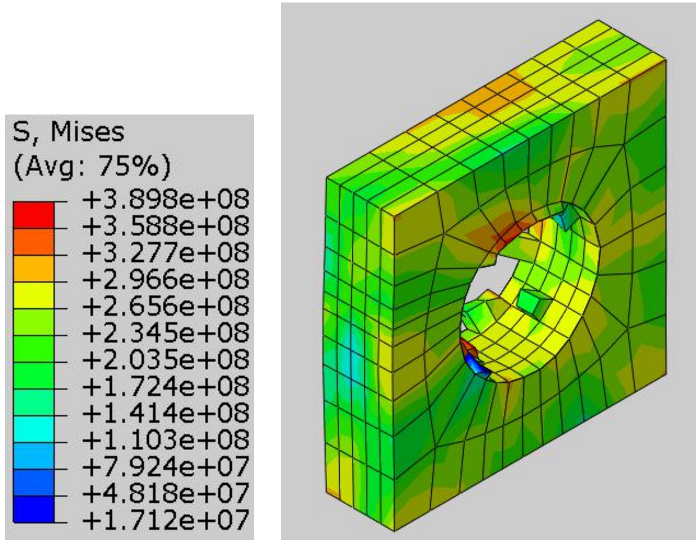
Postprocessing

Select ProjectileImpact job in the manager and pick **Results**

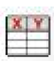
Pick  to plot the contours on deformed shape.

Select Mises in the Field Output toolbar if not selected by default.

The following figure shows the distribution of von Mises stresses after the impact of the projectile.

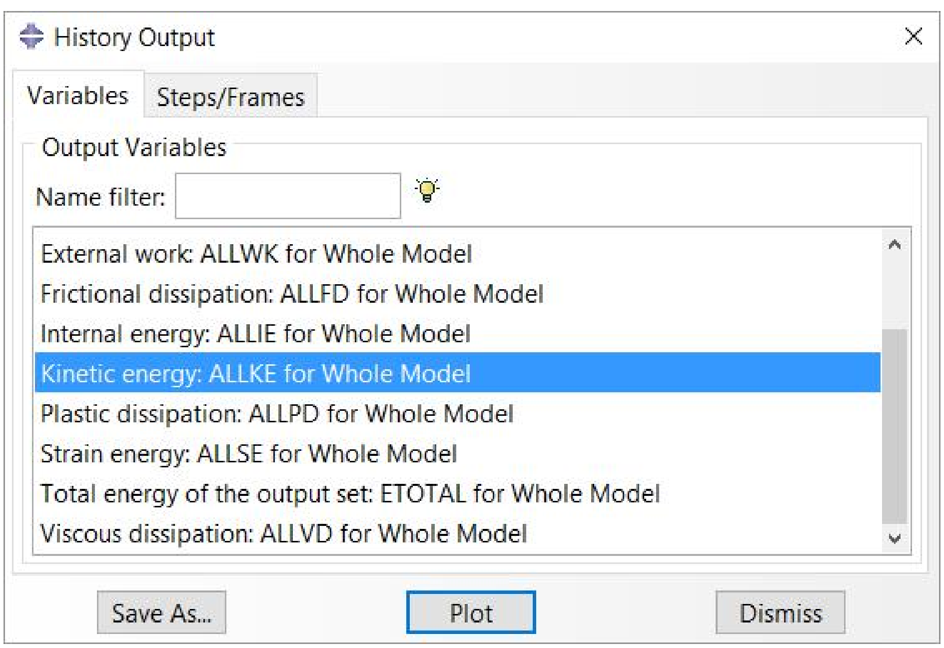


It is of interest to know the projectile's loss of kinetic energy in the impact analysis. So we will plot the history of kinetic energy.

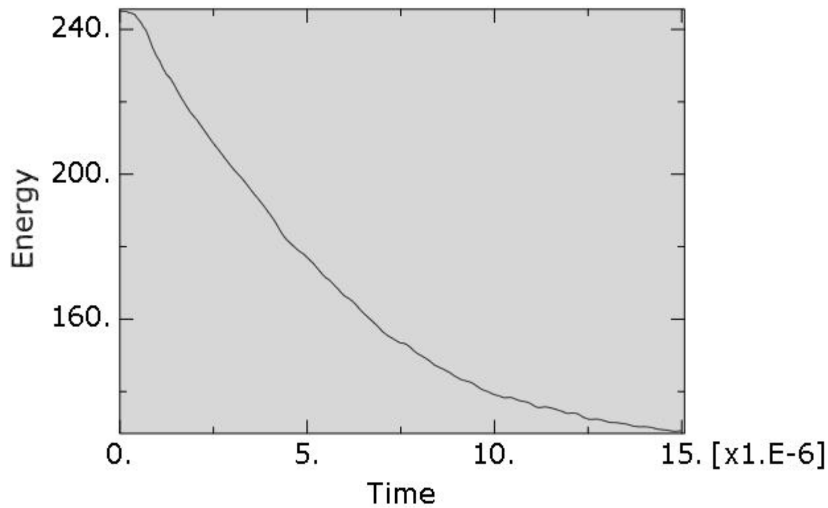
So pick  and Create XY Data dialog box will appear.

Pick **ODB history output > Continue**

In the History Output dialog box, select **ALLKE**.



Pick **Plot** and the graph will appear as shown below.



Approximately 47% of the initial kinetic is absorbed by the impact.

Select **File > Save** to save the changes we made so for.