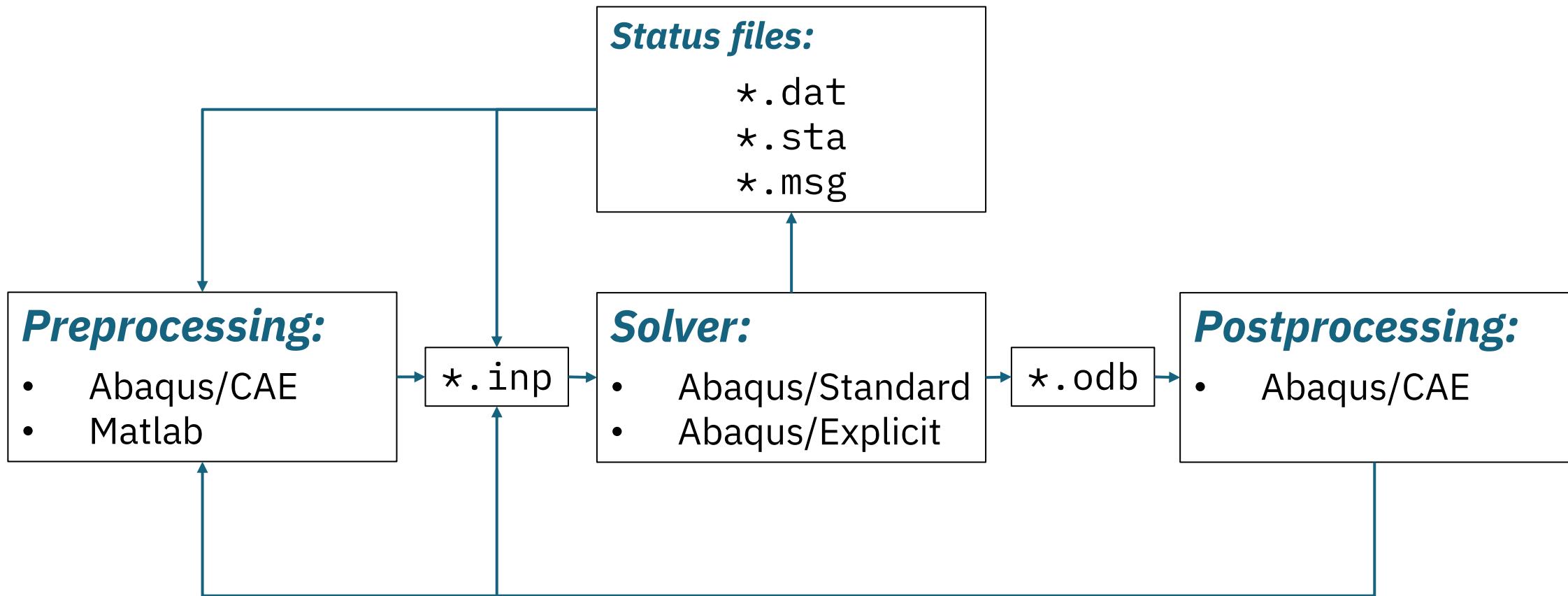


Brief introduction to Abaqus

Marco Riva

1 – Abaqus workflow

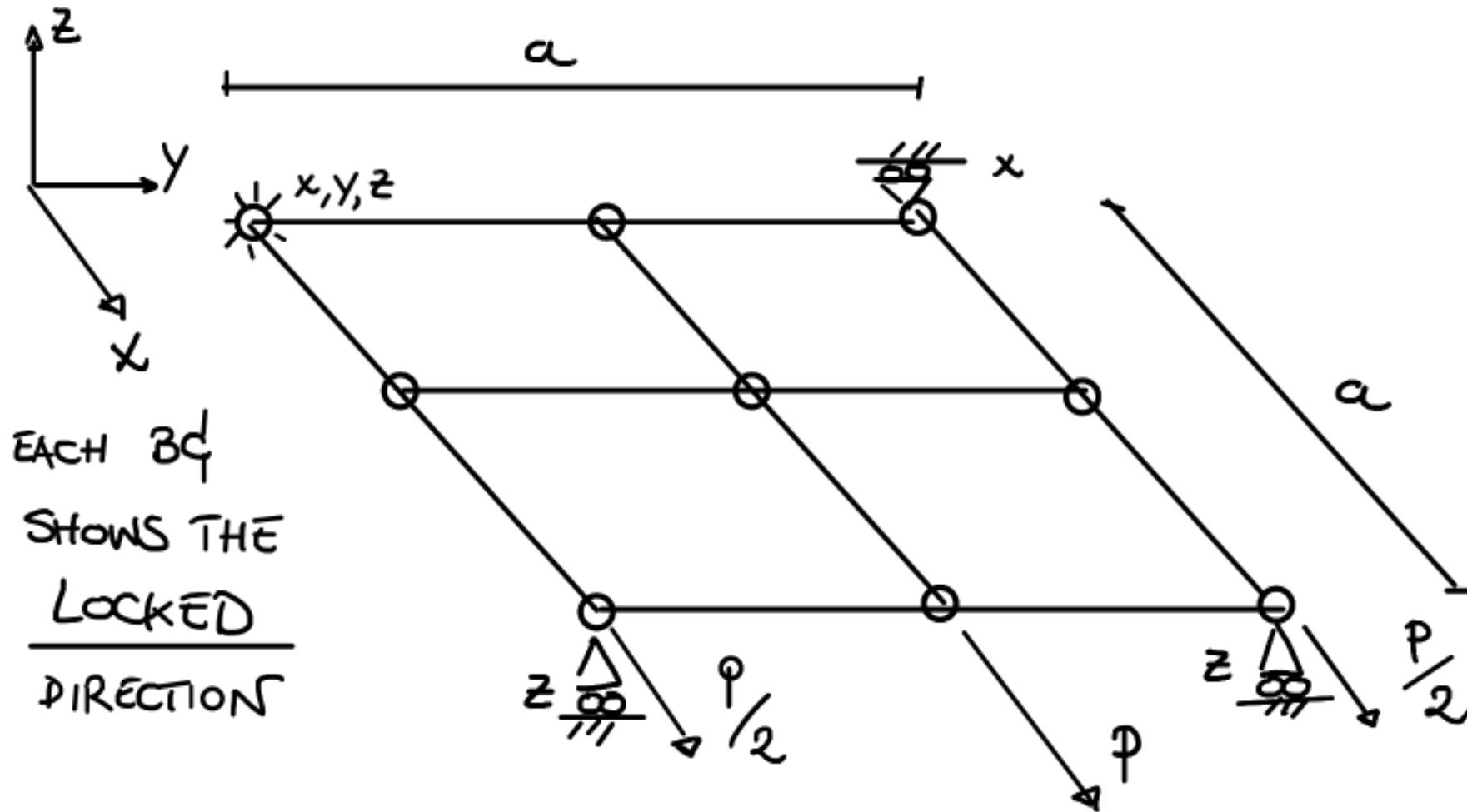


2 – Information contained in the *.inp file

The input file must contain all the information required by the solver:

- Discretized geometry: nodal position, element type and nodal connectivity
- Elements sections: associate the constitutive law to the element (material, thickness, shape, ...)
- Material: constitutive laws
- Loads and BC: introduce constraints and load (of different types) in the model
- Analysis type: define the required type of analysis to be performed
- Output: selected the desired output

3 – Problem under study



Properties:
 $a = 100 \text{ mm}$
 $t = 1 \text{ mm}$
 $E = 72 \text{ GPa}$
 $\nu = 0.3$
 $P = 10 \text{ N}$

4 – Abaqus units of measure

In Abaqus it is not implemented any system of units of measure.

This means that it is up to the user to use **consistent units of measure** during the modeling and the interpretation of the results.

Abaqus guide has a table to summarize the different consistent units of measure.



Consistent units				
Quantity	SI	SI (mm)	US Unit (ft)	US Unit (inch)
Length	m	mm	ft	in
Force	N	N	lbf	lbf
Mass	kg	tonne (10^3 kg)	slug	$\text{lbf s}^2/\text{in}$
Time	s	s	s	s
Stress	$\text{Pa (N/m}^2\text{)}$	$\text{MPa (N/mm}^2\text{)}$	lbf/ft^2	$\text{psi (lbf/in}^2\text{)}$
Energy	$\text{J (N} \times \text{m)}$	$\text{mJ (10}^{-3}\text{ J)}$	ft lbf	in lbf
Density	kg/m^3	tonne/mm^3	slug/ft^3	$\text{lbf s}^2/\text{in}^4$

We will use this set

5 – Input file

```
*****  
**      PART  
*****  
**  
*NODE, NSET=ALL_NODES  
1,    0.0,    0.0,    0.0  
2,   50.0,    0.0,    0.0  
3,  100.0,    0.0,    0.0  
4,    0.0,   50.0,    0.0  
5,   50.0,   50.0,    0.0  
6,  100.0,   50.0,    0.0  
7,    0.0,  100.0,    0.0  
8,   50.0,  100.0,    0.0  
9,  100.0,  100.0,    0.0  
**  
*ELEMENT, TYPE=54, ELSET=ALL_ELEMENTS  
1, 1, 2, 5, 4  
2, 2, 3, 6, 5  
3, 4, 5, 8, 7  
4, 5, 6, 9, 8  
**  
*SHELL SECTION, ELSET=ALL_ELEMENTS, MATERIAL=ALUMINIUM  
1.0  
**
```

Add to SET (store them for recalling)

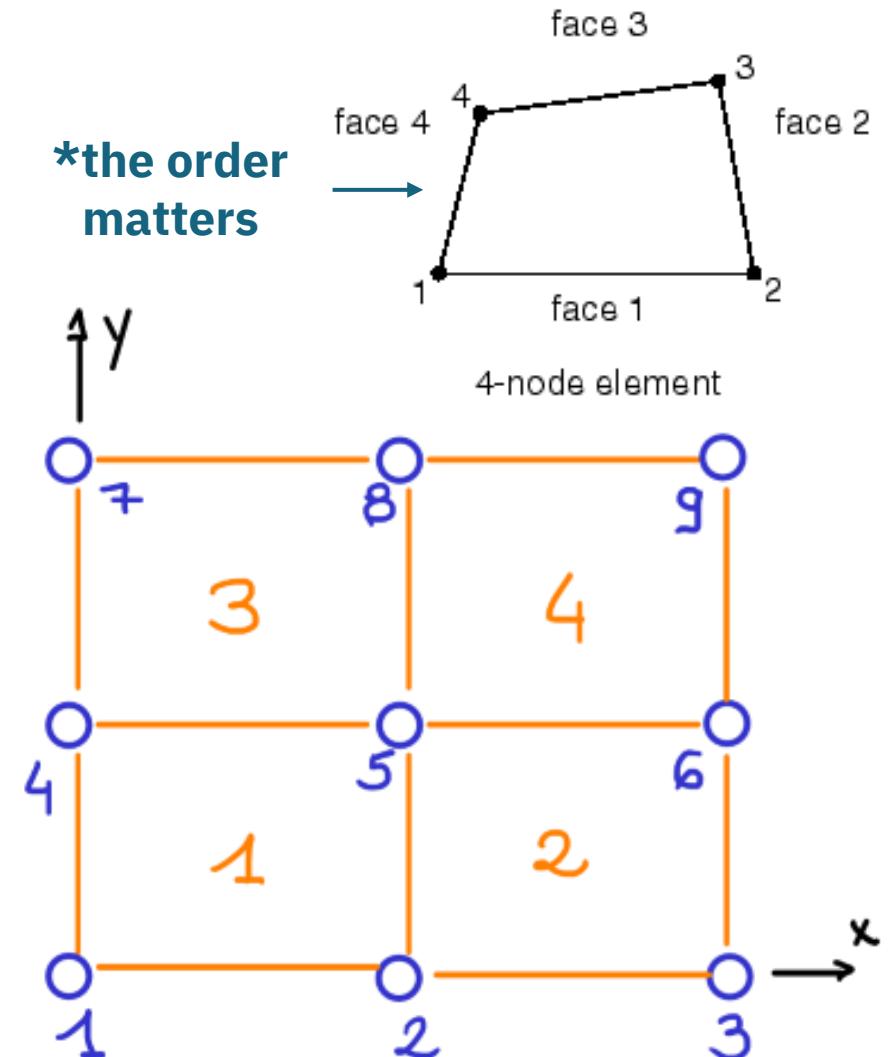
Node position ID, x, y, z

El. type (4 nodes shell)

Elements connectivity ID, N1, N2, N3, N4 *the order matters

Section definition with Element Set (ELSET) to which it is referred to and the material used. thickness

*the order matters



5 – Input file

```
*****  
** MATERIAL  
*****  
**  
*MATERIAL, NAME=ALUMINIUM  
*ELASTIC  
72000., 0.3,  
**  
*****  
** STEP  
*****  
**  
*STEP, NAME=STEP-1  
*STATIC  
** BC  
*BOUNDARY  
1, 1, 3, 0  
3, 3, 3, 0  
7, 1, 1, 0  
9, 3, 3, 0  
** LOAD  
*CLOAD  
3, 1, 5  
6, 1, 10  
9, 1, 5  
** OUTPUT  
*OUTPUT, FIELD, VARIABLE=PRESELECT  
*OUTPUT, HISTORY, VARIABLE=PRESELECT  
*END STEP
```

Material definition (*MATERIAL) and type of constitutive law (*ELASTIC) E, ν

Step definition (type of analysis)

Boundary conditions definition:
NODE_ID, FirstDoF, LastDoF, Value
Line one means that the node 1 has the DoF 1, 2, 3 fixed to zero. The ordered DoF are the three translations on x, y, z and the three rotations on x, y, z

Loads definition:
NODE_ID, Direction, Value
Line one means that the node 3 has a force of 5 on direction 1 (x)

Output request (predefined outputs)

Remarks:

The input file is based on its own syntax, which work as:

*KEYWORD [, OPTIONS]
[LINES OF DATA]

Note that this was just a simplified example, there are many keywords which perform different functions (different load types, different constraints, interactions, material laws, ...)

For completeness always refer to the Abaqus manual at:

[Abaqus manual](#)

(register with your POLIMI email address)



POLITECNICO
MILANO 1863

Virtual desktop guide

1 Virtual Desktop set up

To be able to use Virtual Desktop follow the instruction that you find at [Virtual Desktop Polimi](#).

Note that if you want to use your personal PC in classroom you can avoid the cable and your access will be considered a remote one.

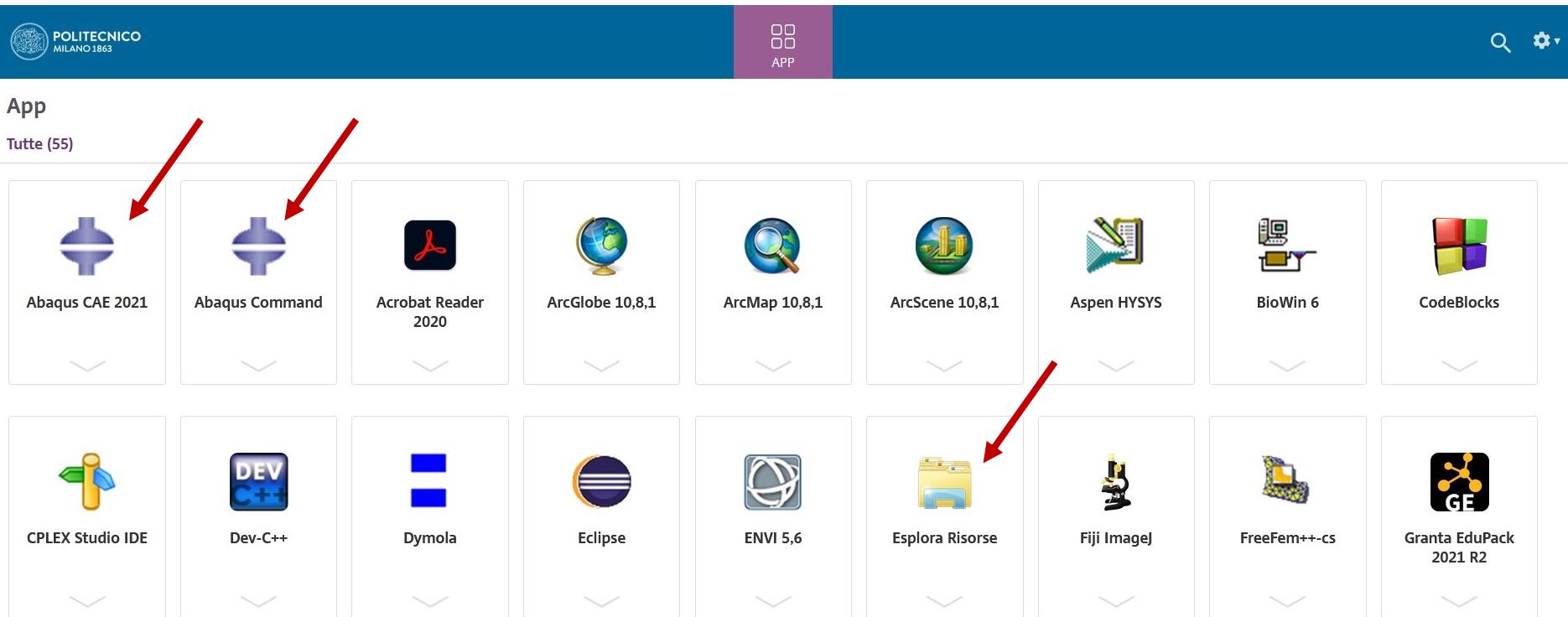
The screenshot shows a web page from the Politecnico di Milano website. At the top, there's a navigation bar with links for 'Online Services' (containing icons for Italian and English flags), 'Services', 'How to...', and 'Welcome kit'. Below the navigation, there's a section for 'ICT SERVICES' and 'Students, professors, temporary research fellows and researchers'. A 'Activation' section is shown, stating that the service does not require activation and is available at once for all recipients. A large central box contains instructions for 'Access in classroom' (with a blue circular icon containing a white arrow pointing right) and 'Via personal Pc'. The 'Via personal Pc' section lists four steps: 1. connect the cable to a network socket (with a note about permanent connection), 2. download and install the Citrix client (with a link to help and setup), 3. open a browser page and connect to the address <https://virtualdesktop.polimi.it/>, and 4. log in with University login credentials. Below this, another section titled 'Via PC supplied to the classroom' lists three steps: 1. Switch on the PC and open Internet Explorer, 2. Connect to <https://virtualdesktop.polimi.it/>, and 3. log in with the University login credentials.



1 Virtual Desktop set up

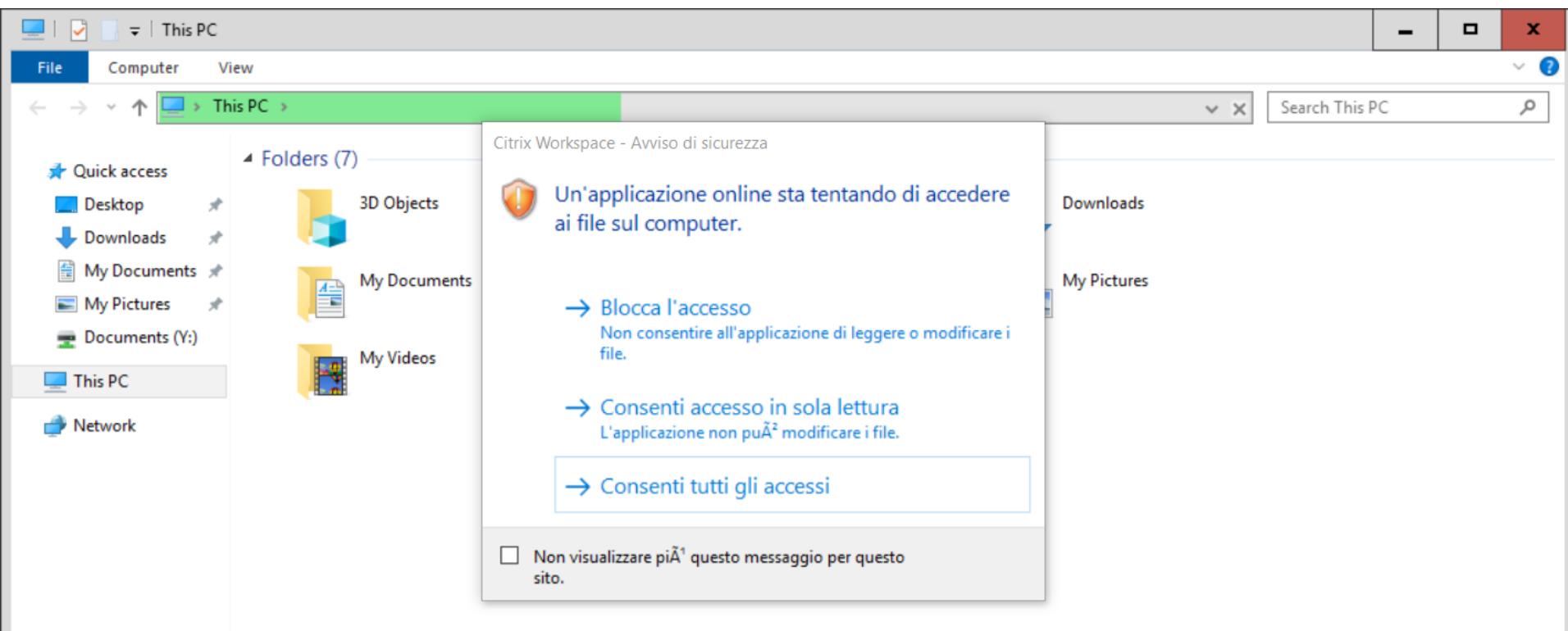
The three applications needed are the one highlighted:

1. Abaqus/CAE
2. Abaqus Command
3. File explorer



2 File explorer set up

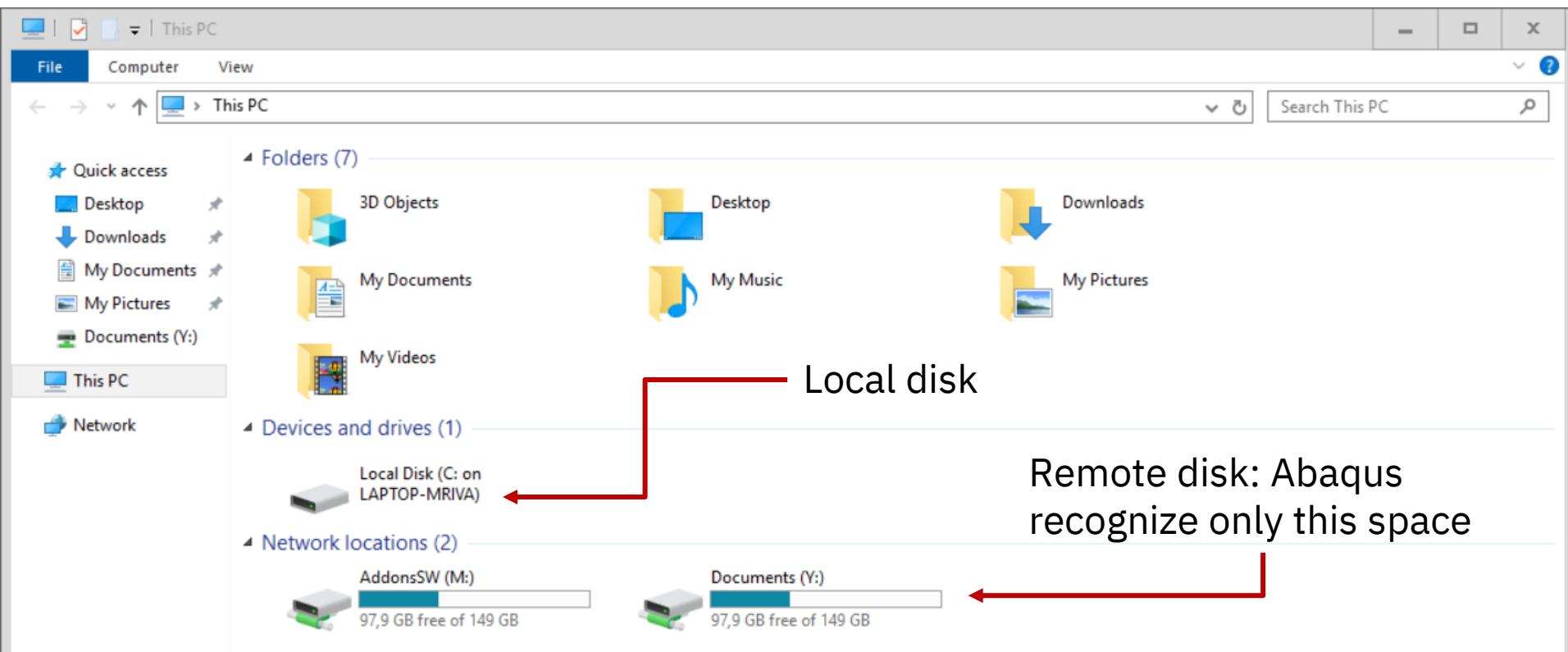
If file explorer asks to have access to your personal file give it the full access for both reading and writing on your disk.



2 File explorer set up

Every input or output of Abaqus is on the remote disk.

You must transfer the needed files between the local disk and the remote one to work on them (simple copy-paste and check by refreshing the folder if everything worked).



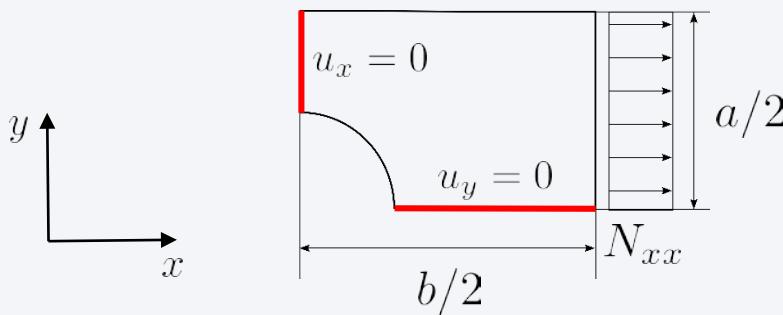
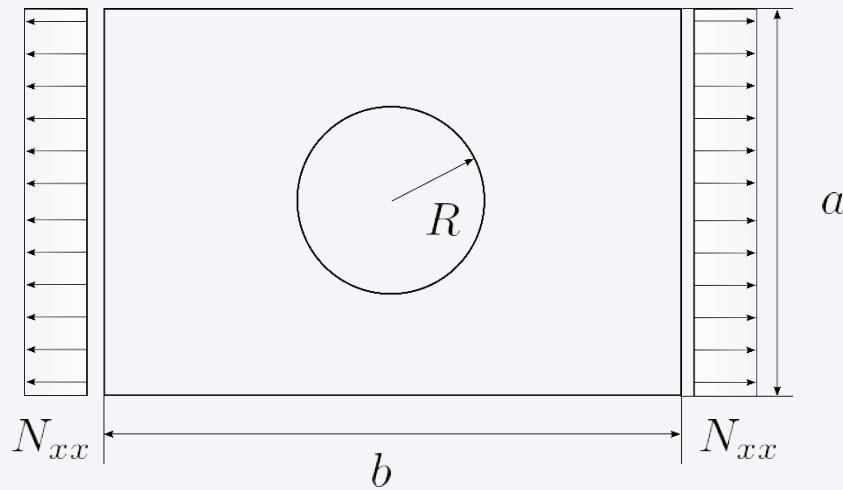
A brief introduction to the use of Abaqus using the CAE

Course of Spacecraft Structures
AY 2017/2018

Riccardo Vescovini
Politecnico di Milano, Department of Aerospace Science and Technology

Example

- Open-hole test: membrane with center hole, loaded in traction (1/4 of the structure due to the double symmetry of the problem)



Input data

$$a = 100 \text{ mm}$$

$$b = 150 \text{ mm}$$

$$R = 25 \text{ mm}$$

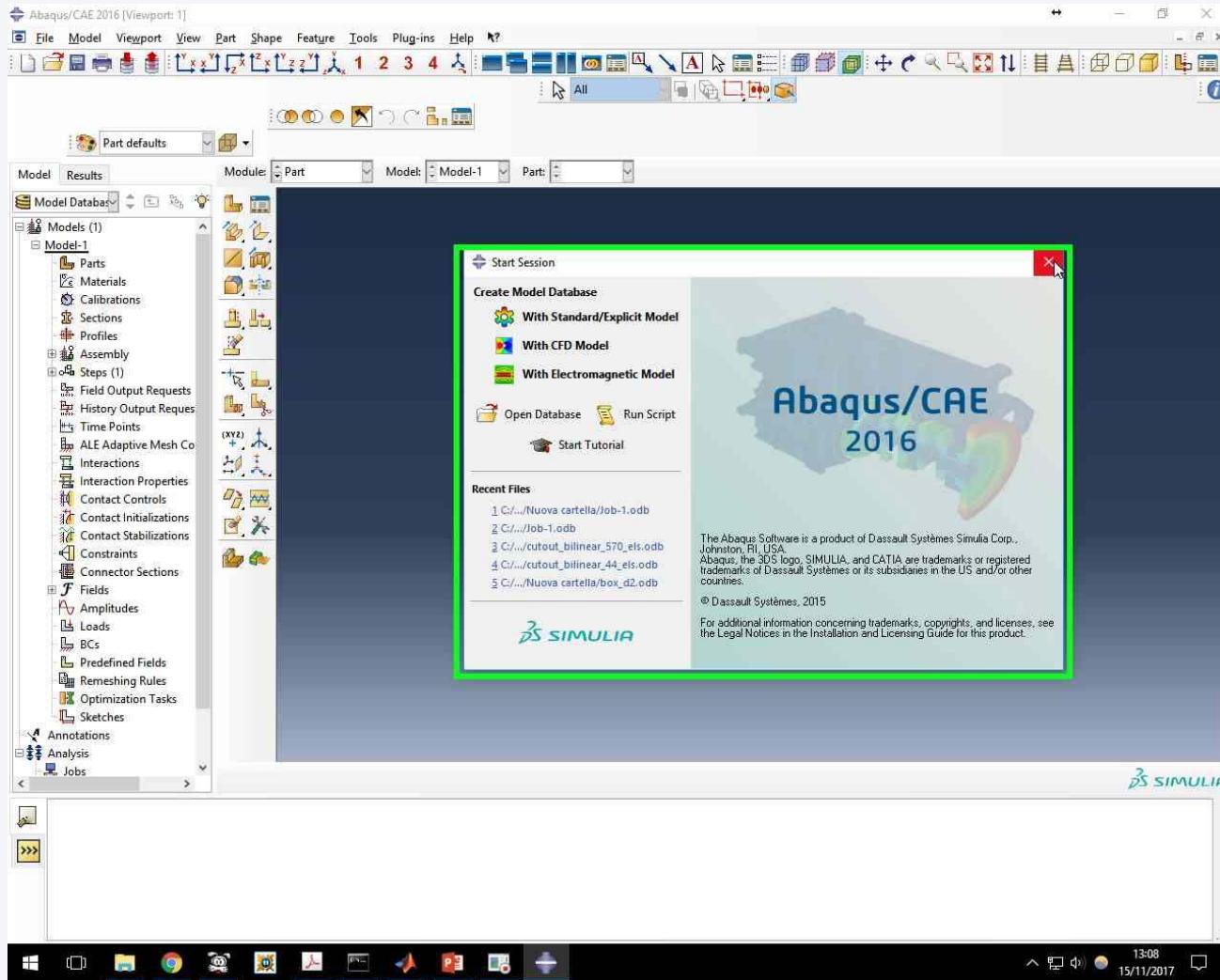
$$t = 1 \text{ mm}$$

$$E = 72 \text{ GPa}$$

$$\nu = 0.3$$

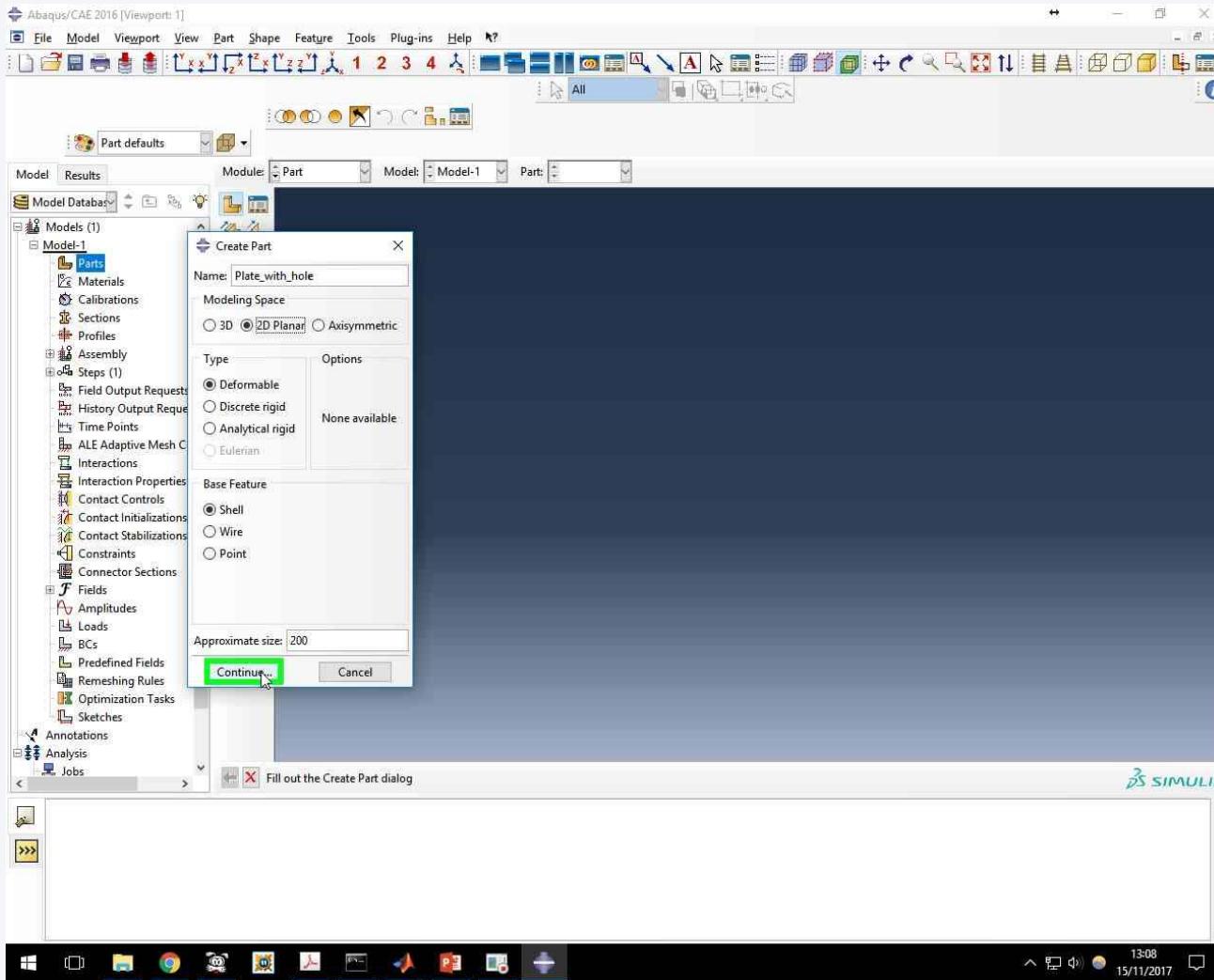
$$N_{xx} = 20 \text{ N/mm}$$

Start session



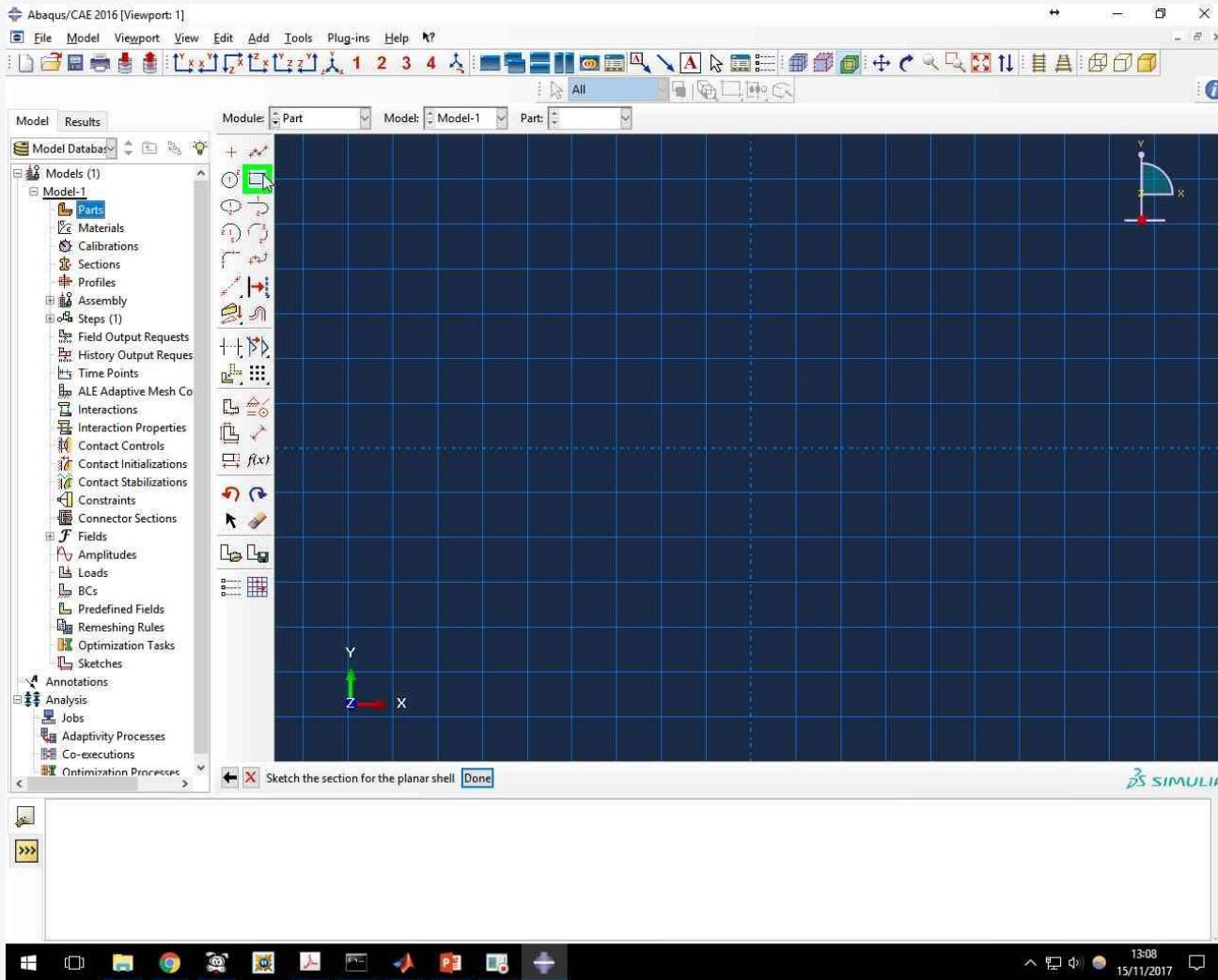
- + Create a folder
- + Open the command prompt and move to the folder (the syntax is «cd c:\my_address\....\My_folder»)
- + Launch the Abaqus command interface with the command «abaqus cae»

Geometry



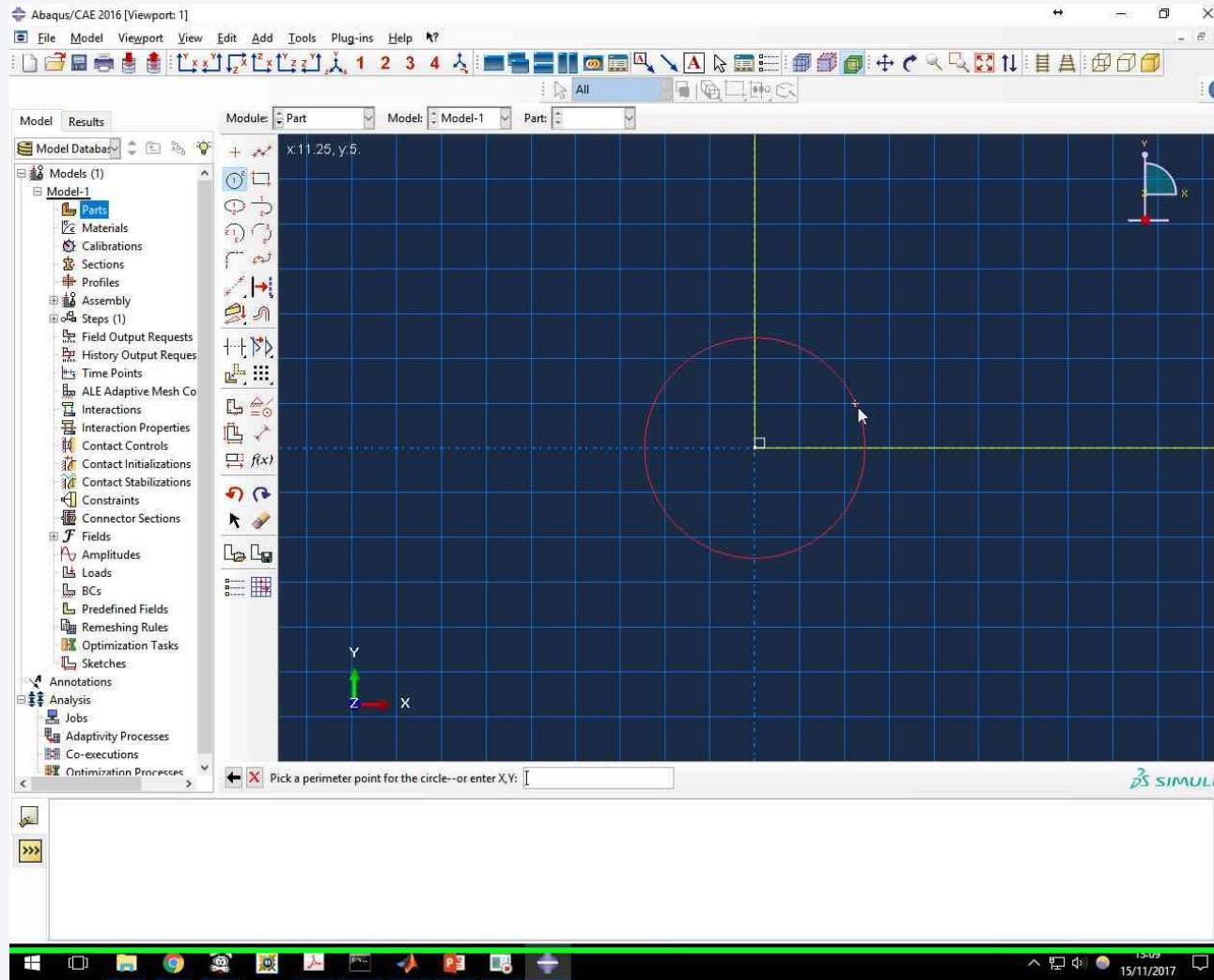
The first step consists in generating the geometry. Note that the geometry has the only role of supporting the generation of the mesh. The finite element model does not involve the geometric quantities defined at this stage.

Geometry



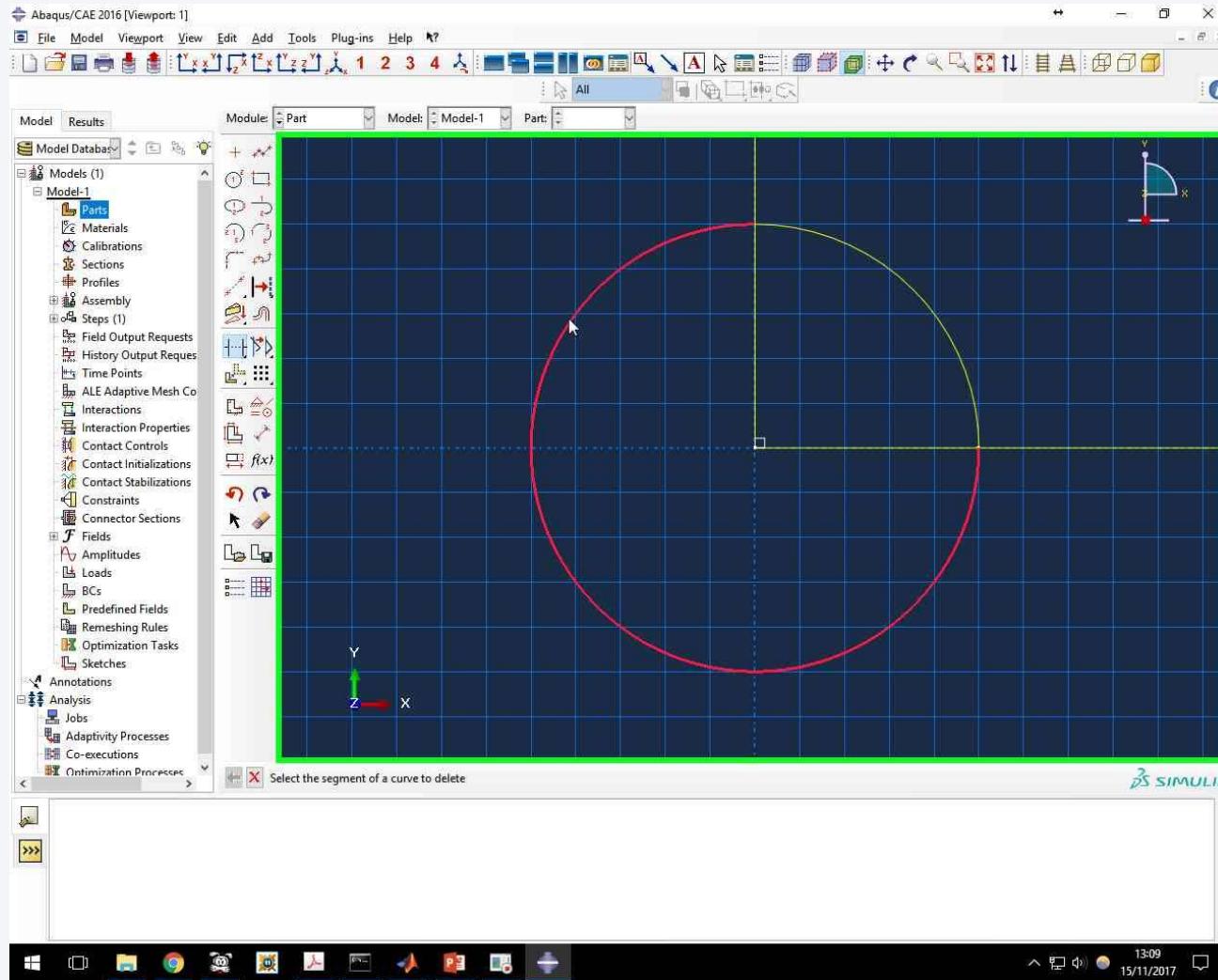
Sketch the rectangle. The corner positions can be defined by inserting the coordinates or by clicking in the corresponding positions.

Geometry



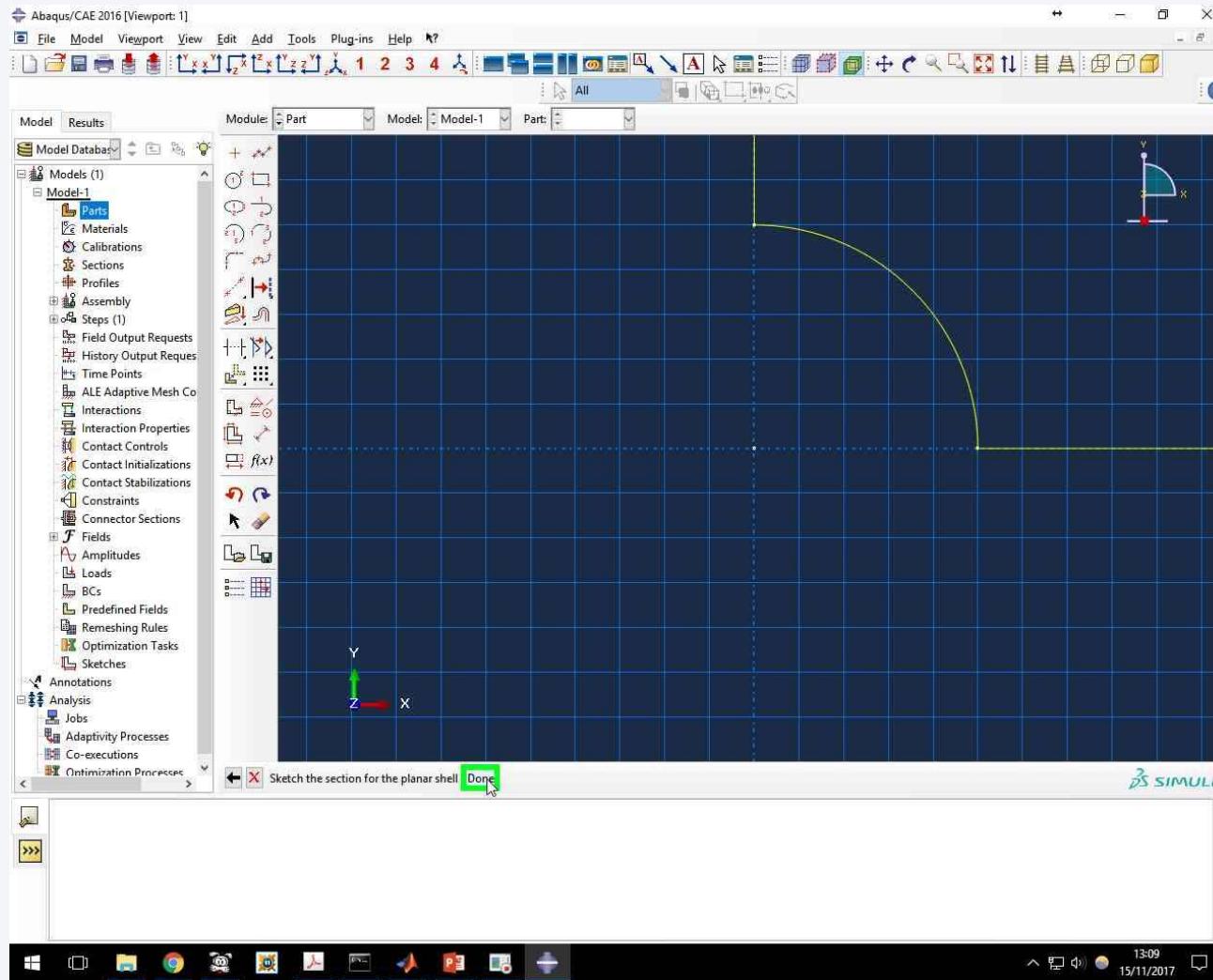
Sketch now the circle to define the hole.

Geometry



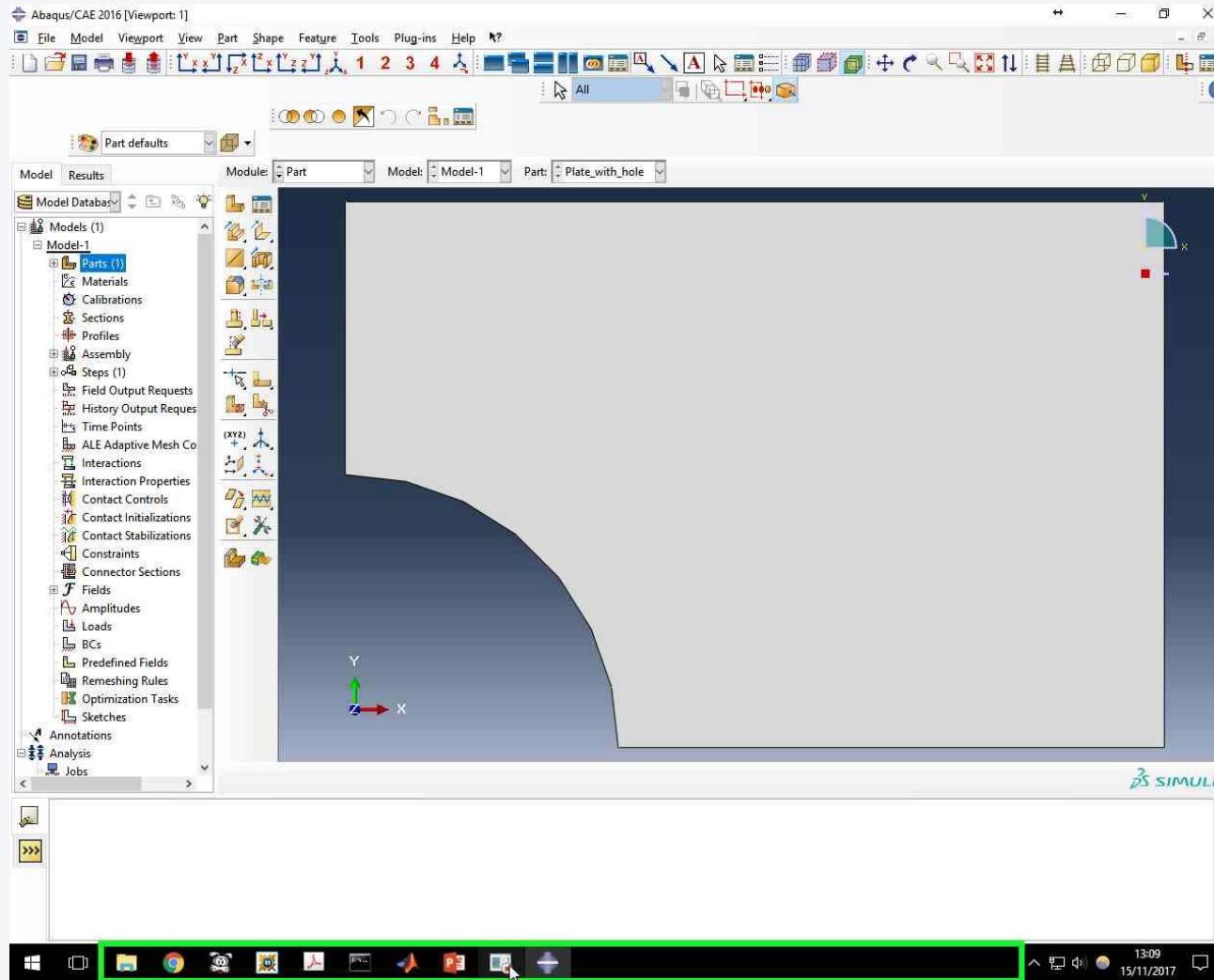
Use the auto-trim command to delete the outer portions of the circle.

Geometry



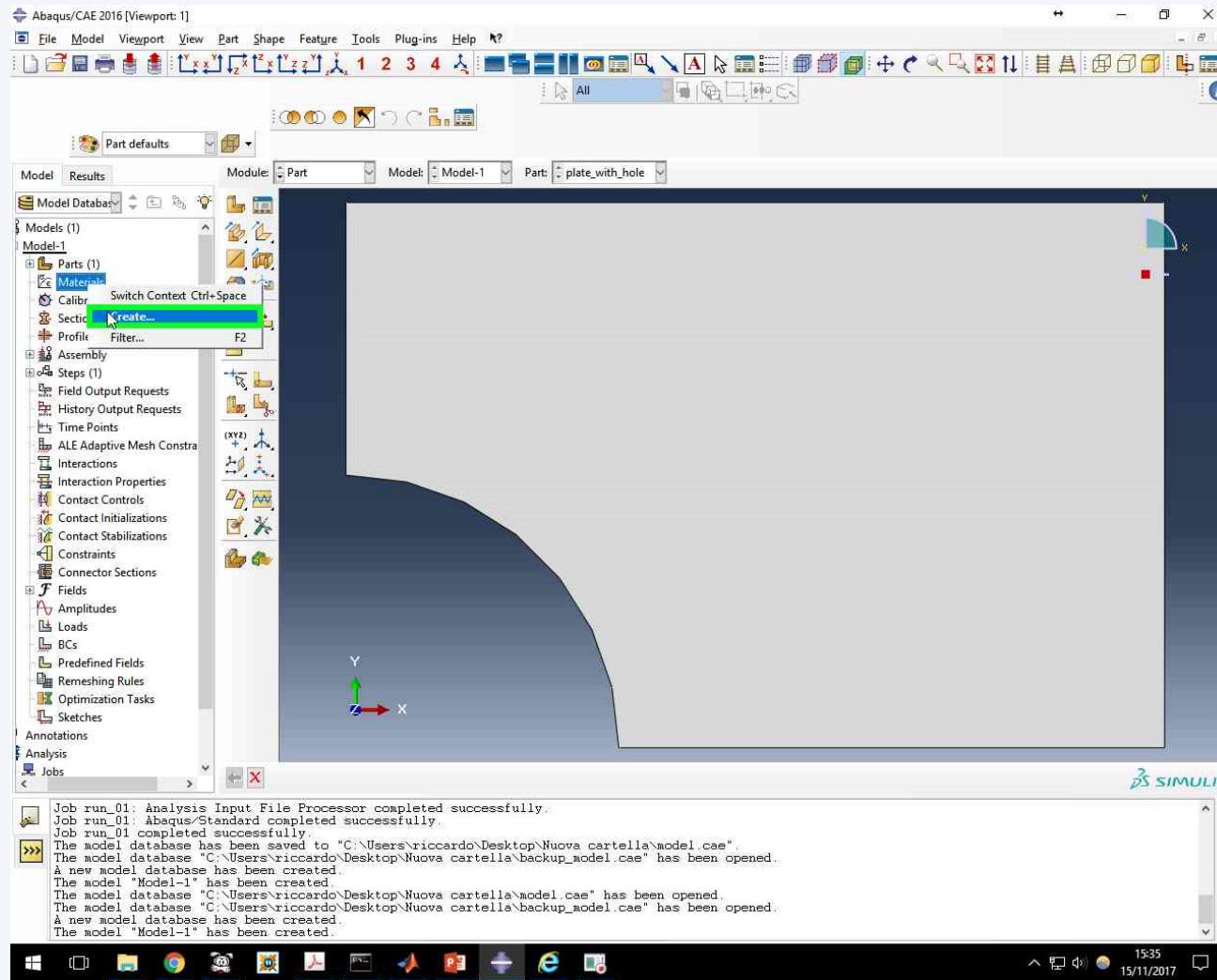
Confirm by clicking on Done.

Geometry



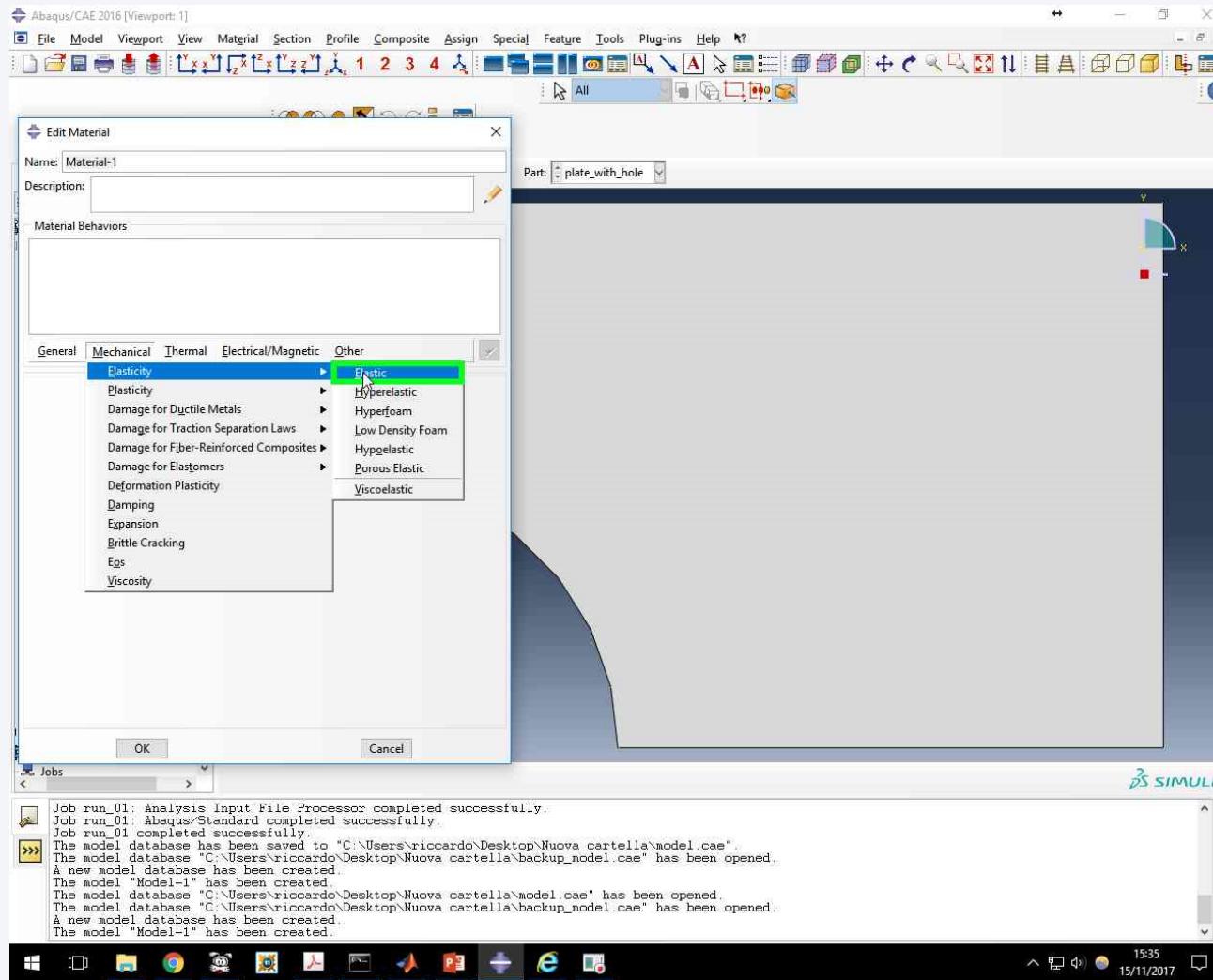
The Part should now appear as illustrated.

Material definition



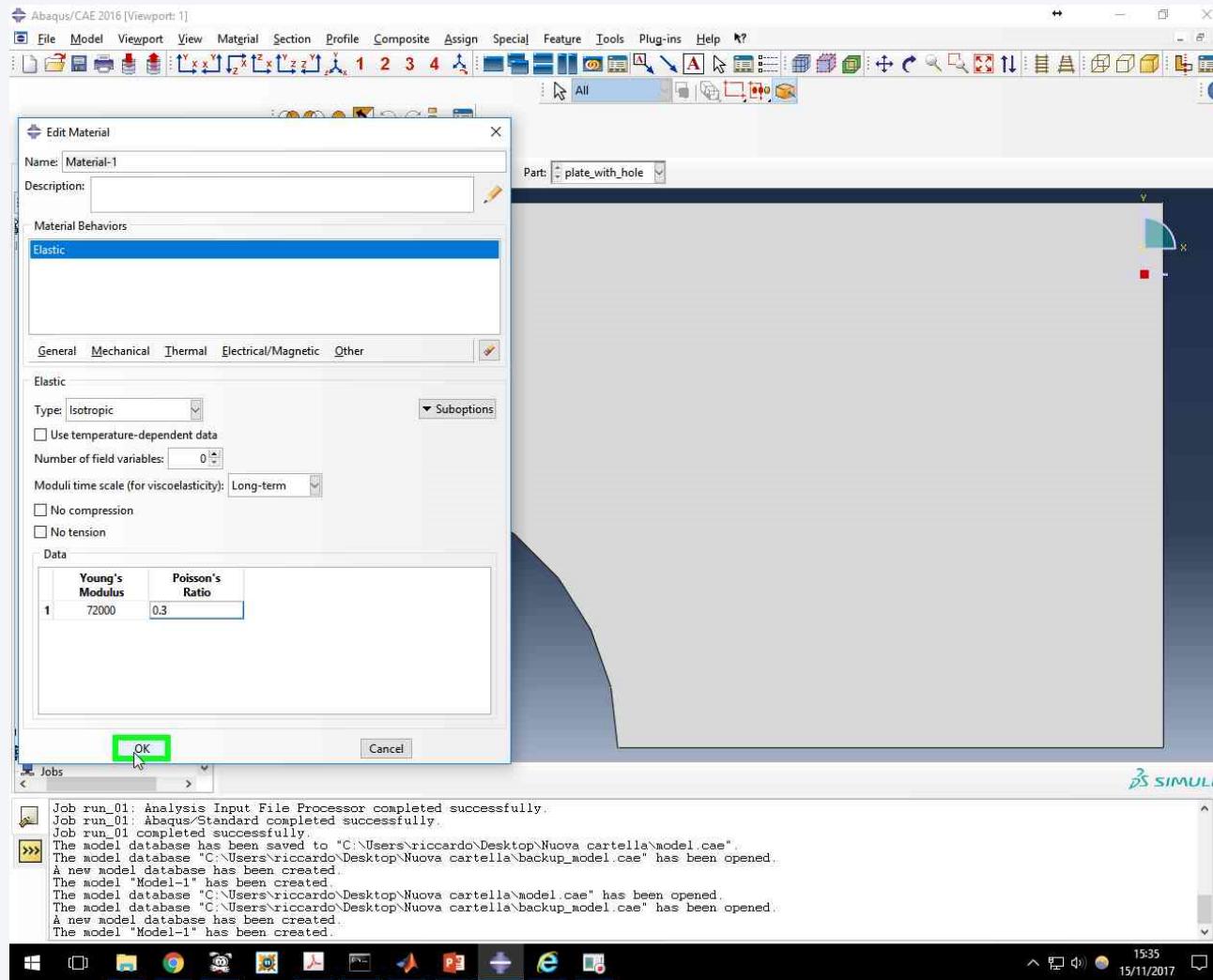
Create the material definition.

Material definition



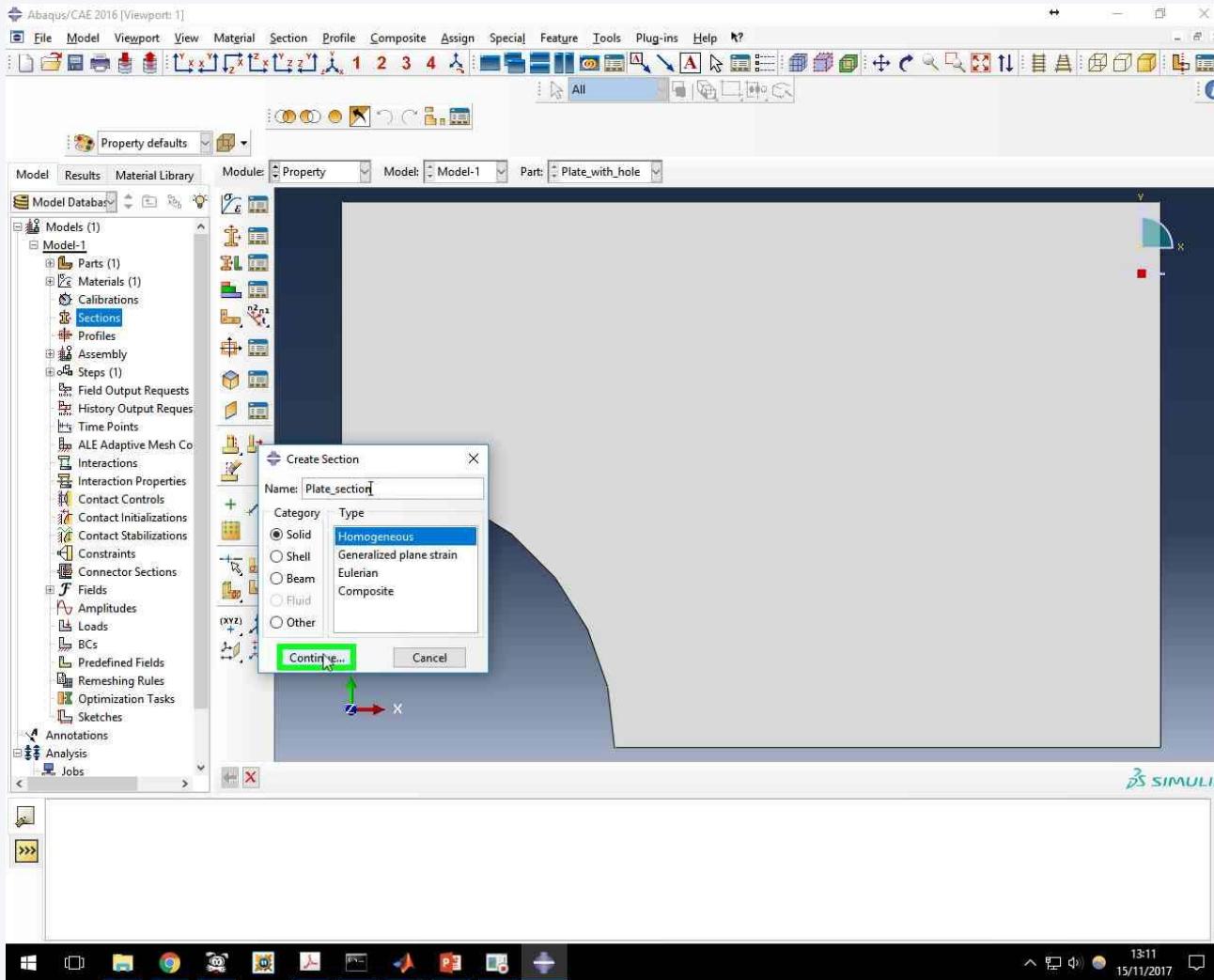
In this case a linear elastic material is considered.

Material definition



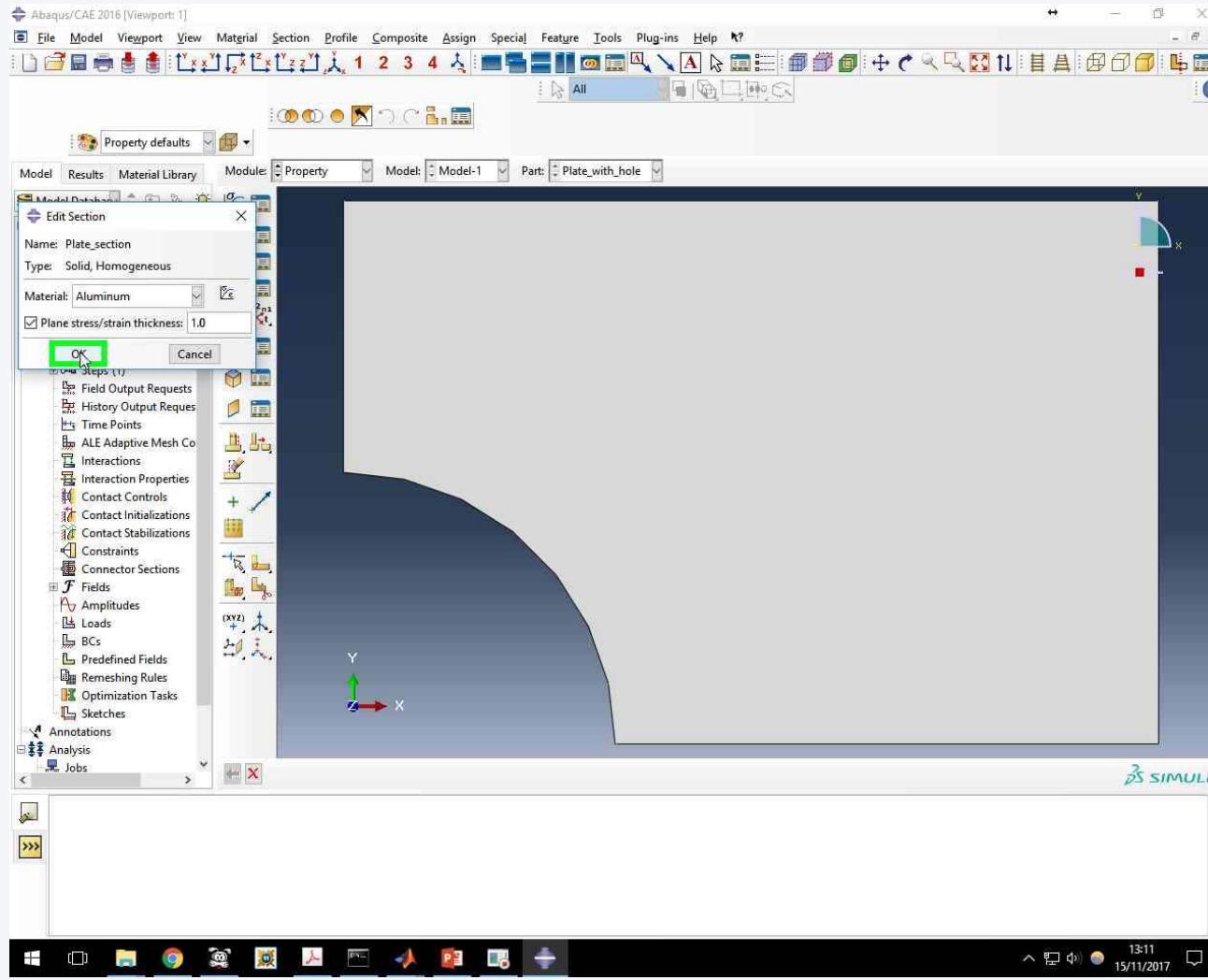
Define the Young's modulus and Poissons's ratios.

Section properties



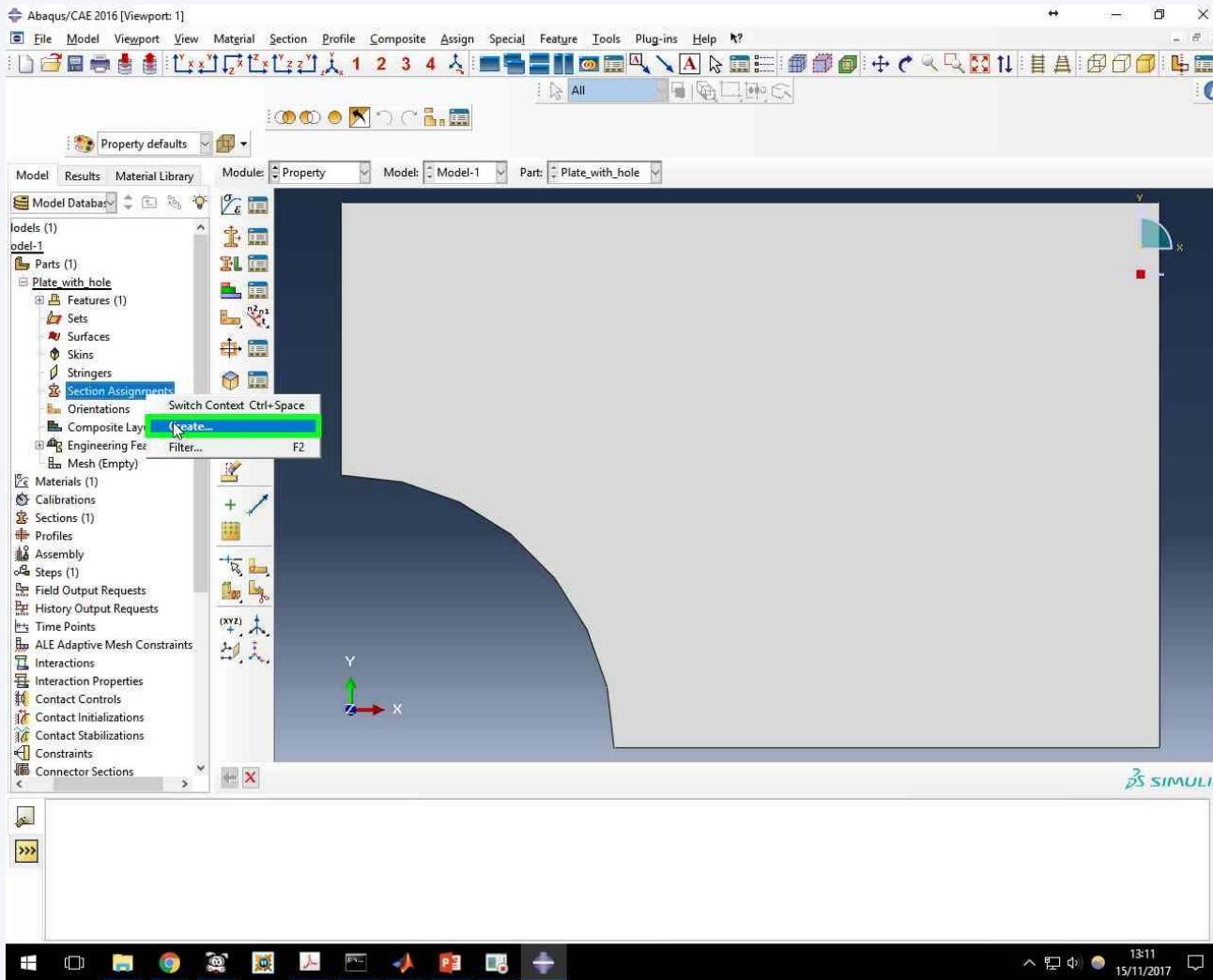
Create now the Section properties. The section, in this case, is used to specify the thickness of the membrane (recall that the membrane model is given by a 2D surface, thus the thickness should be defined to calculate the membrane stiffness).

Section properties



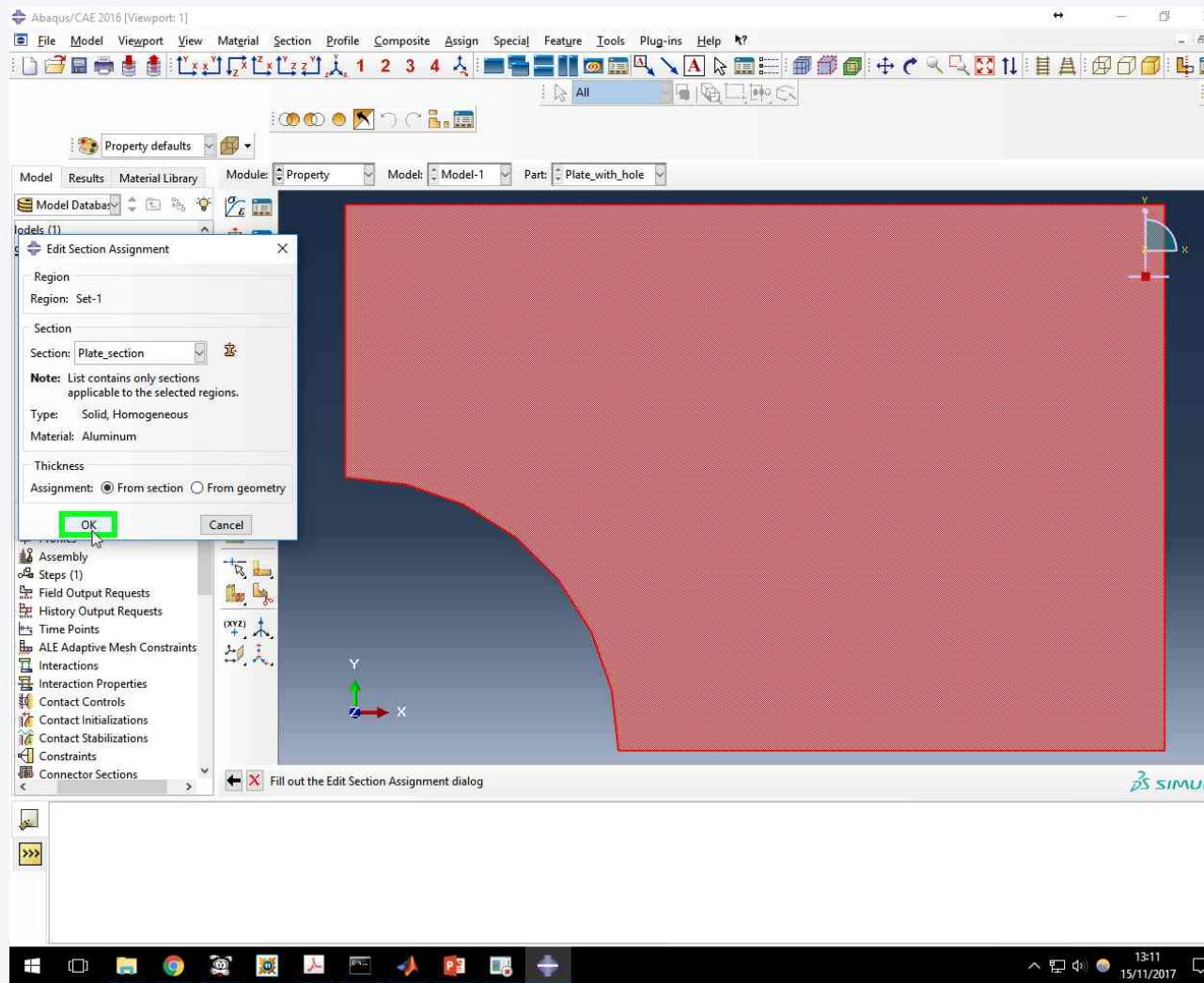
Select the material and confirm.

Section properties



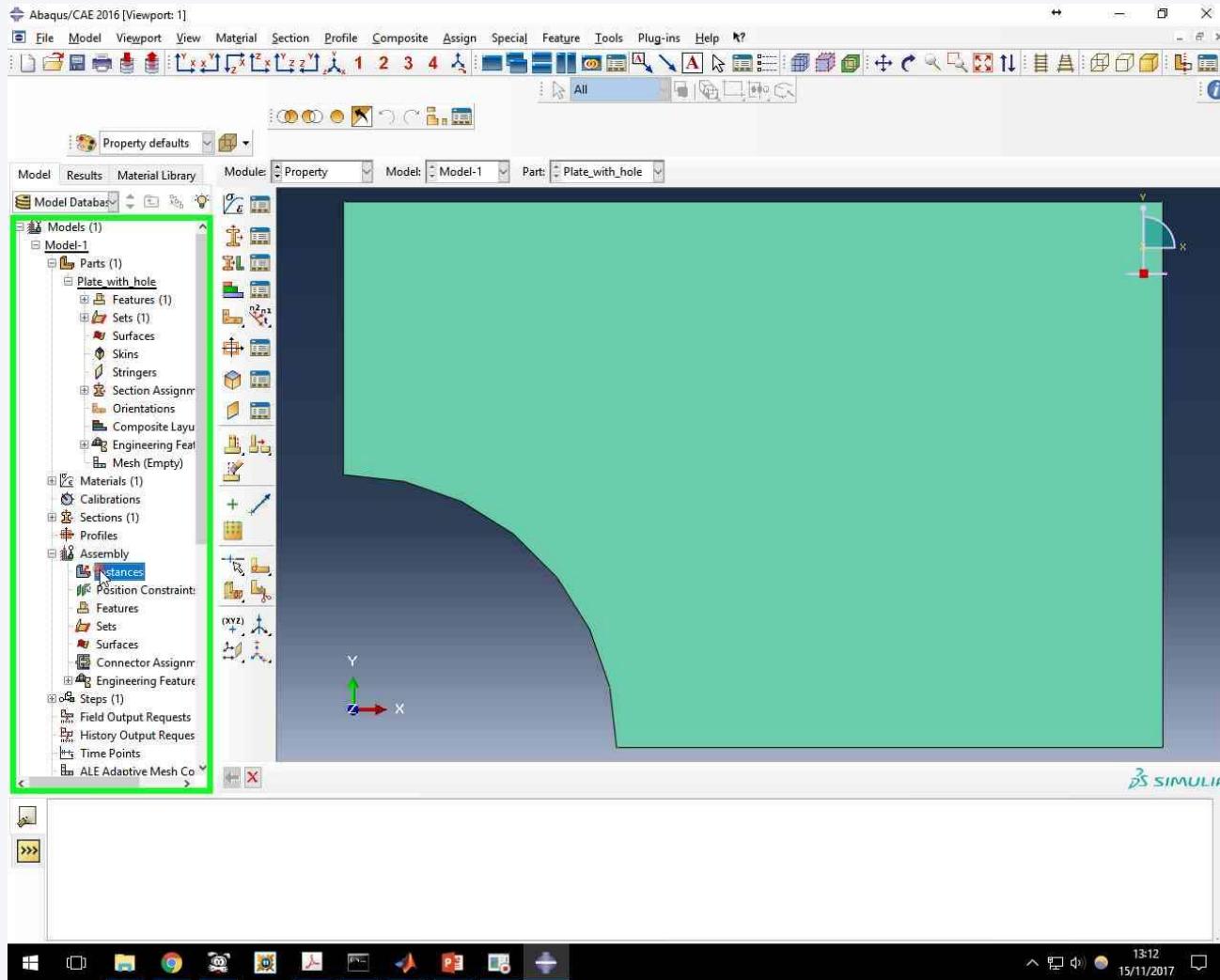
The section previously defined is now assigned to the part. In other words, it is specified that the geometric entity previously sketched is made of Aluminum and has thickness equal to 1.0.

Section properties



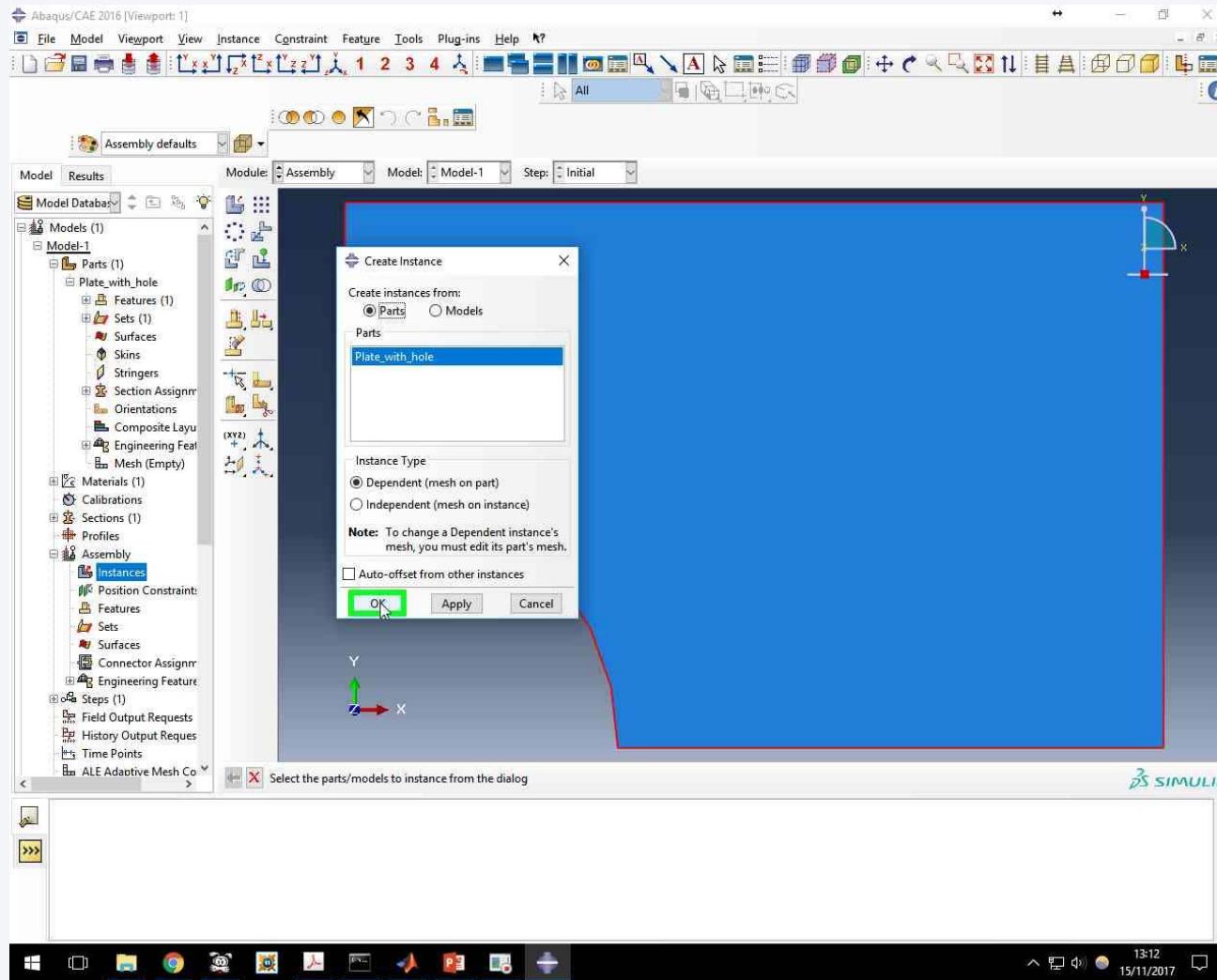
Confirm.

Creating an instance

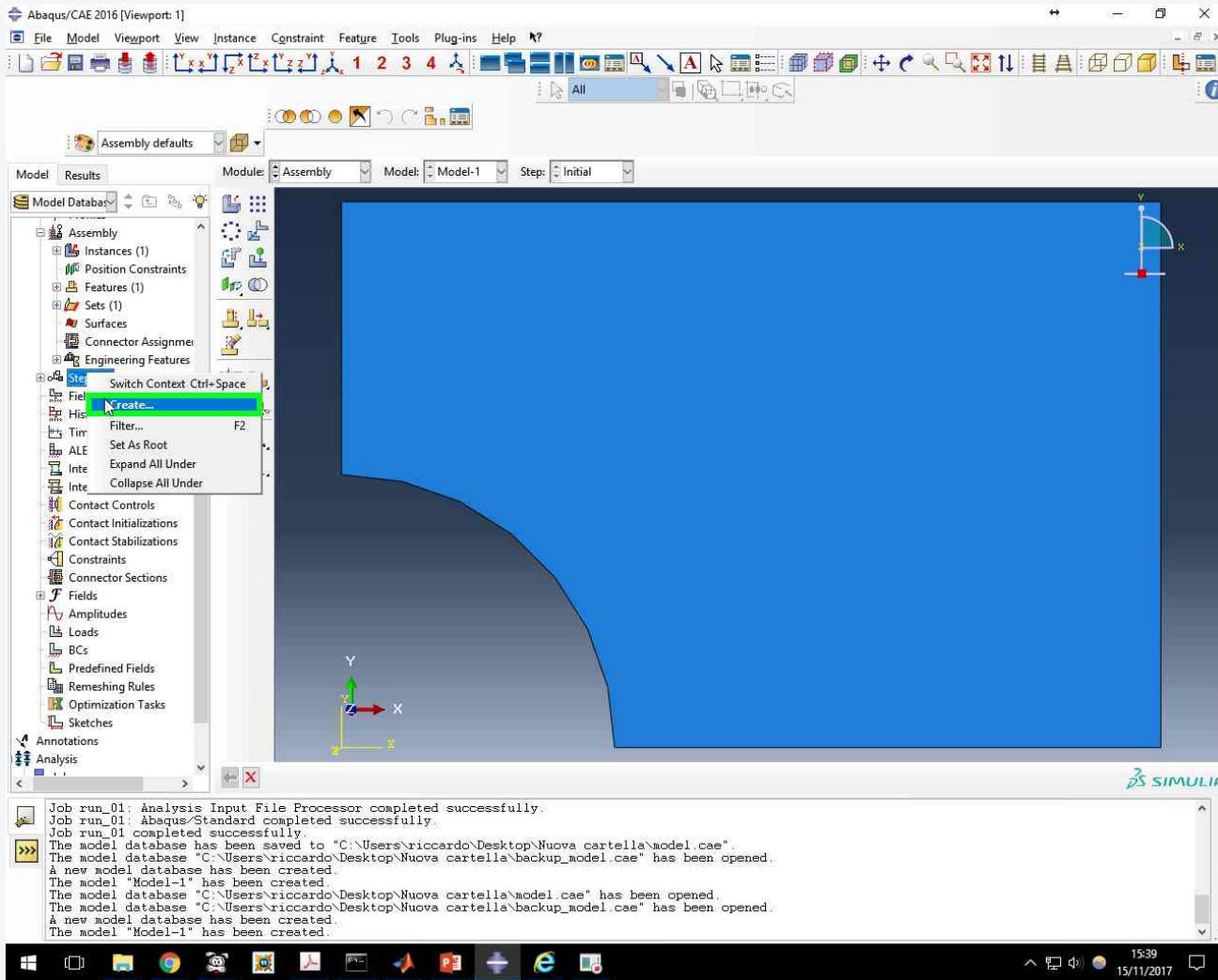


To proceed with the generation of the model it is necessary to create an instance. Right-click on Instance.

Creating an instance

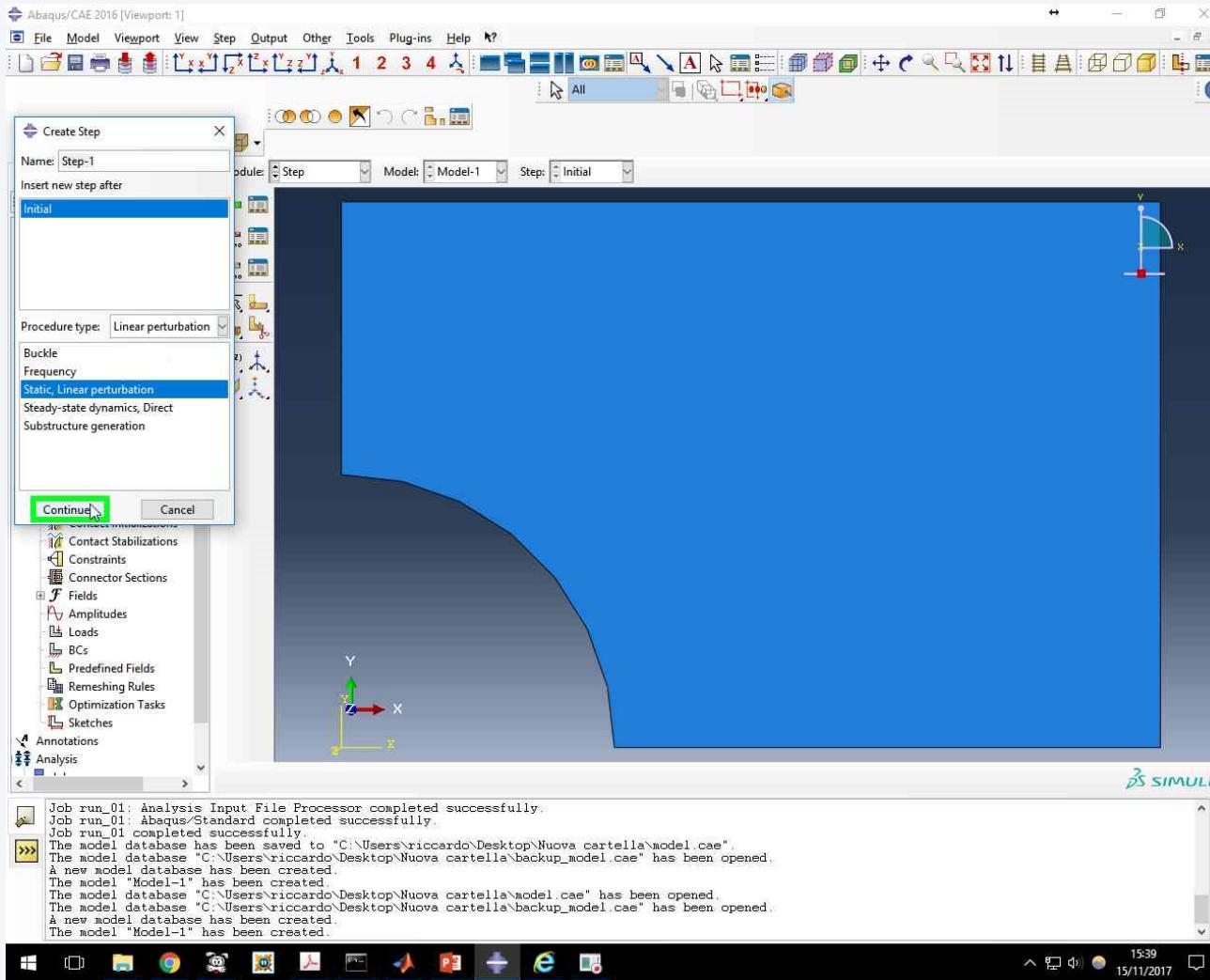


Step definition



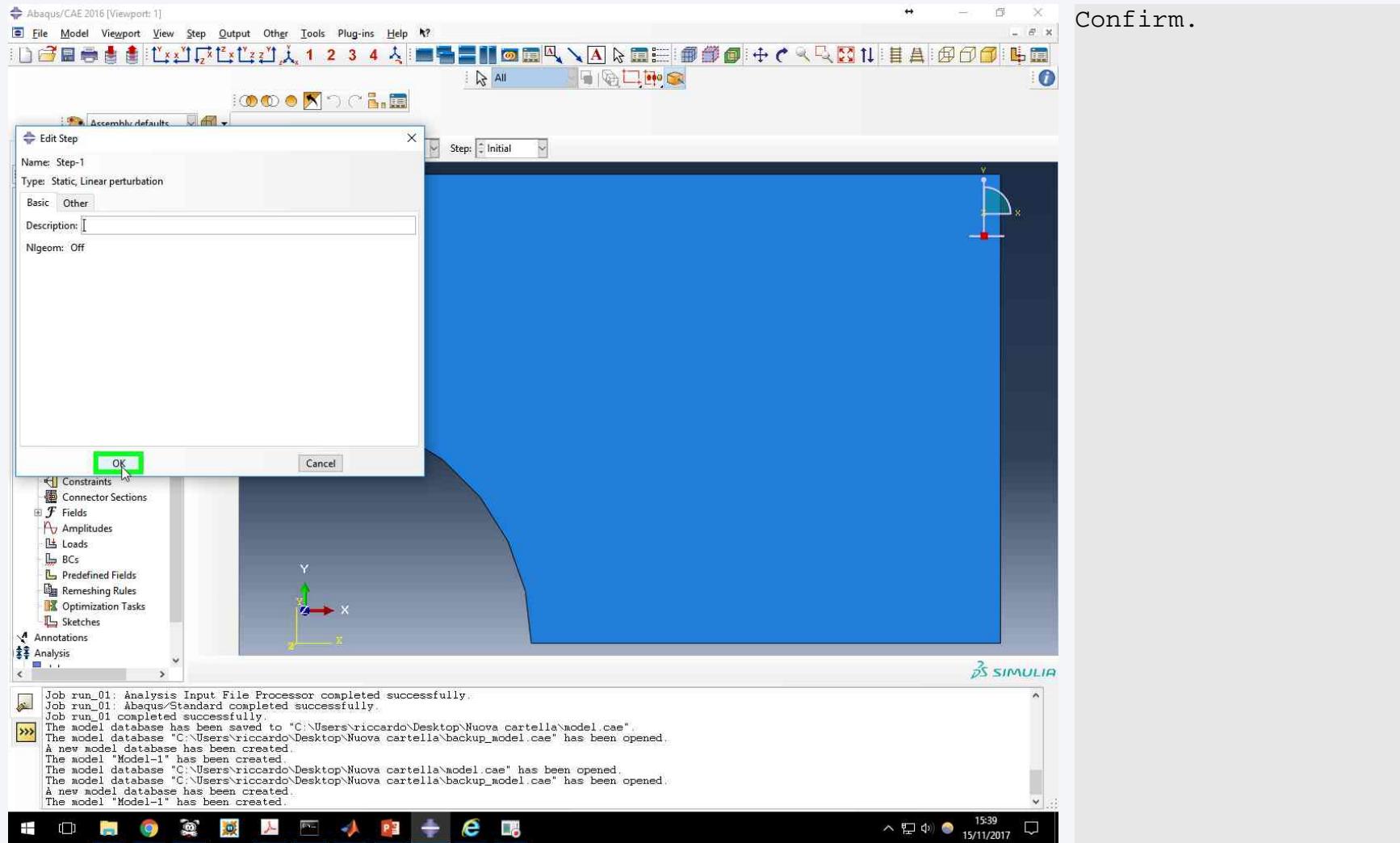
The Step defines the analysis procedure (in this case linear static). In principle, one single analysis can be composed of several steps. For instance, the initial part of the analysis could be a thermal step, and a subsequent step can be the introduction of a mechanical load. In this case one single step is considered.

Step definition

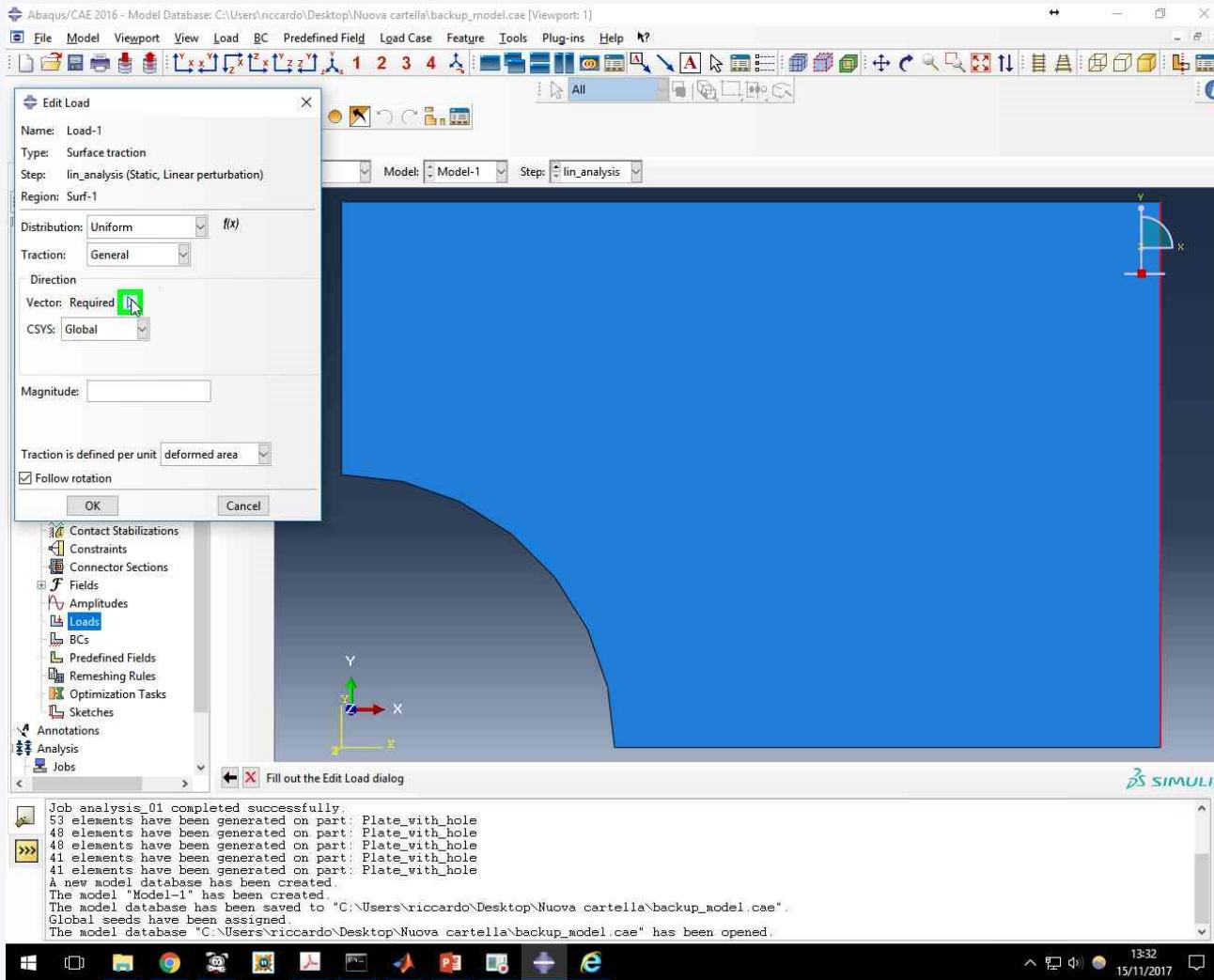


The analysis type is selected among the linear perturbation ones.

Step definition



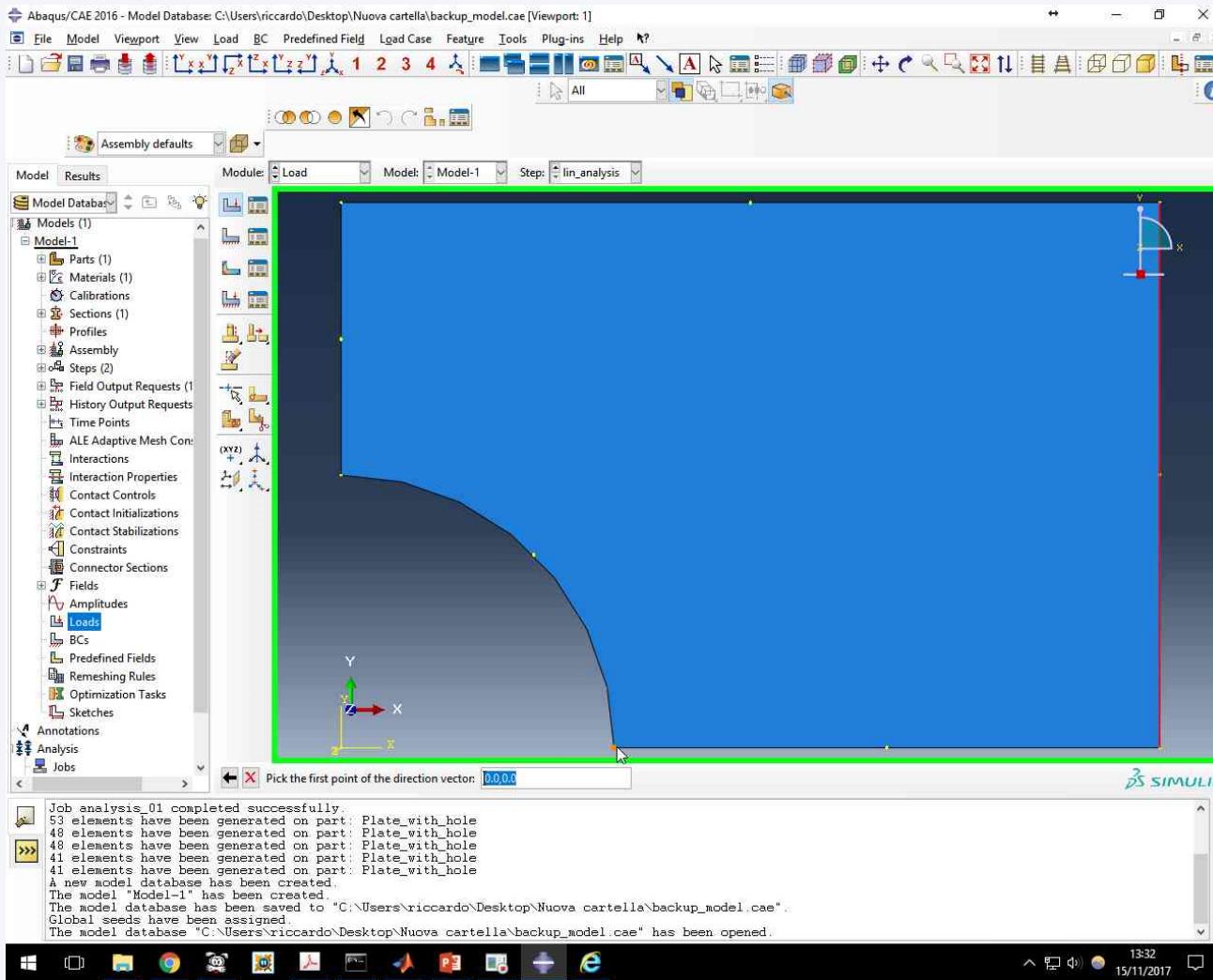
Application of loads



Right click on loads and specify the loading condition. In this case the load is introduced as a distributed force, denoted as "Surface traction".

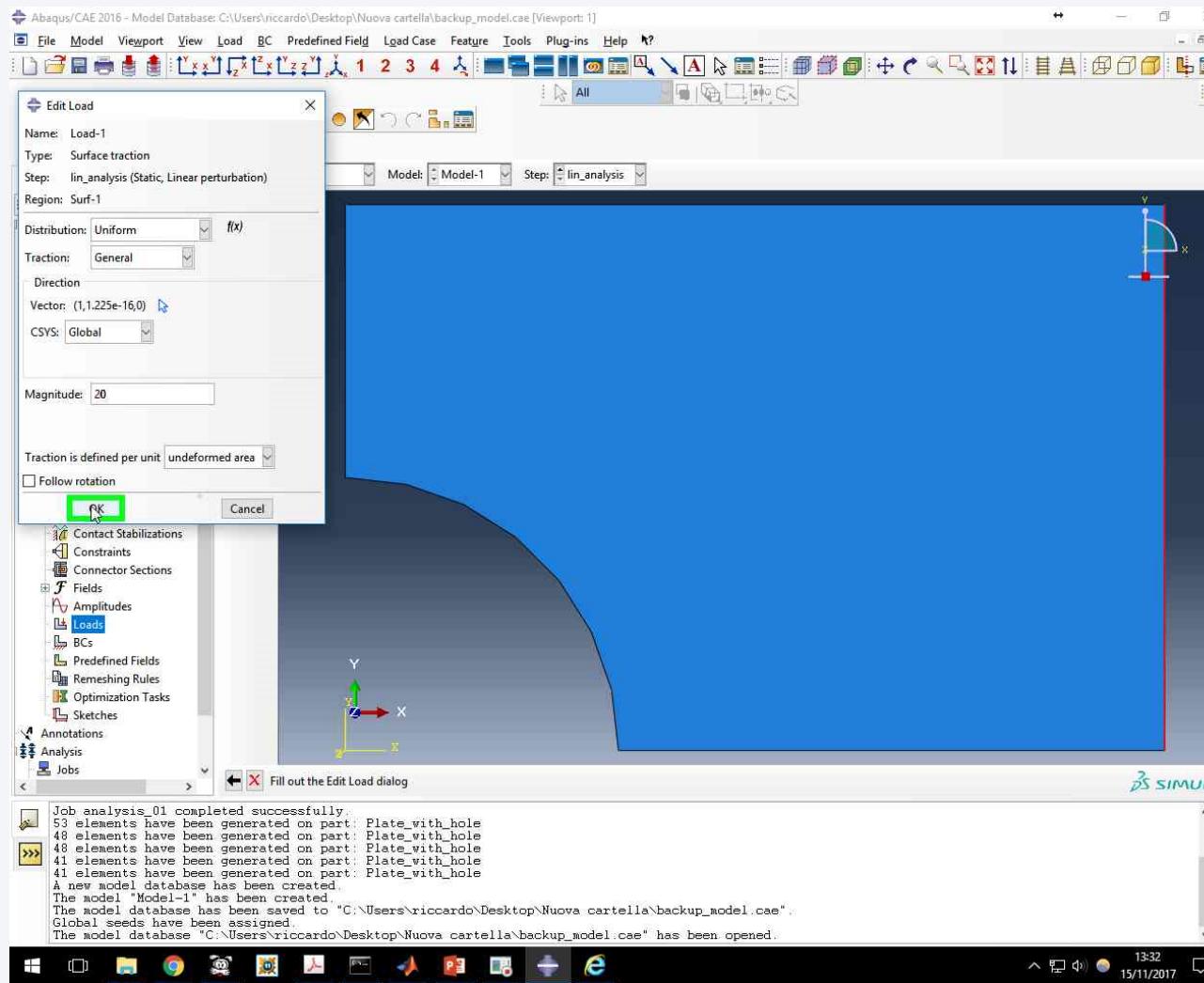
Important remark: the (many) features to introduce the distributed loads are a user-friendly way for quickly specifying the loading conditions. Indeed the user is not required to evaluate by hand the consistent nodal forces. The conversion is done by software, but the final result is always in the form of nodal forces.

Application of loads



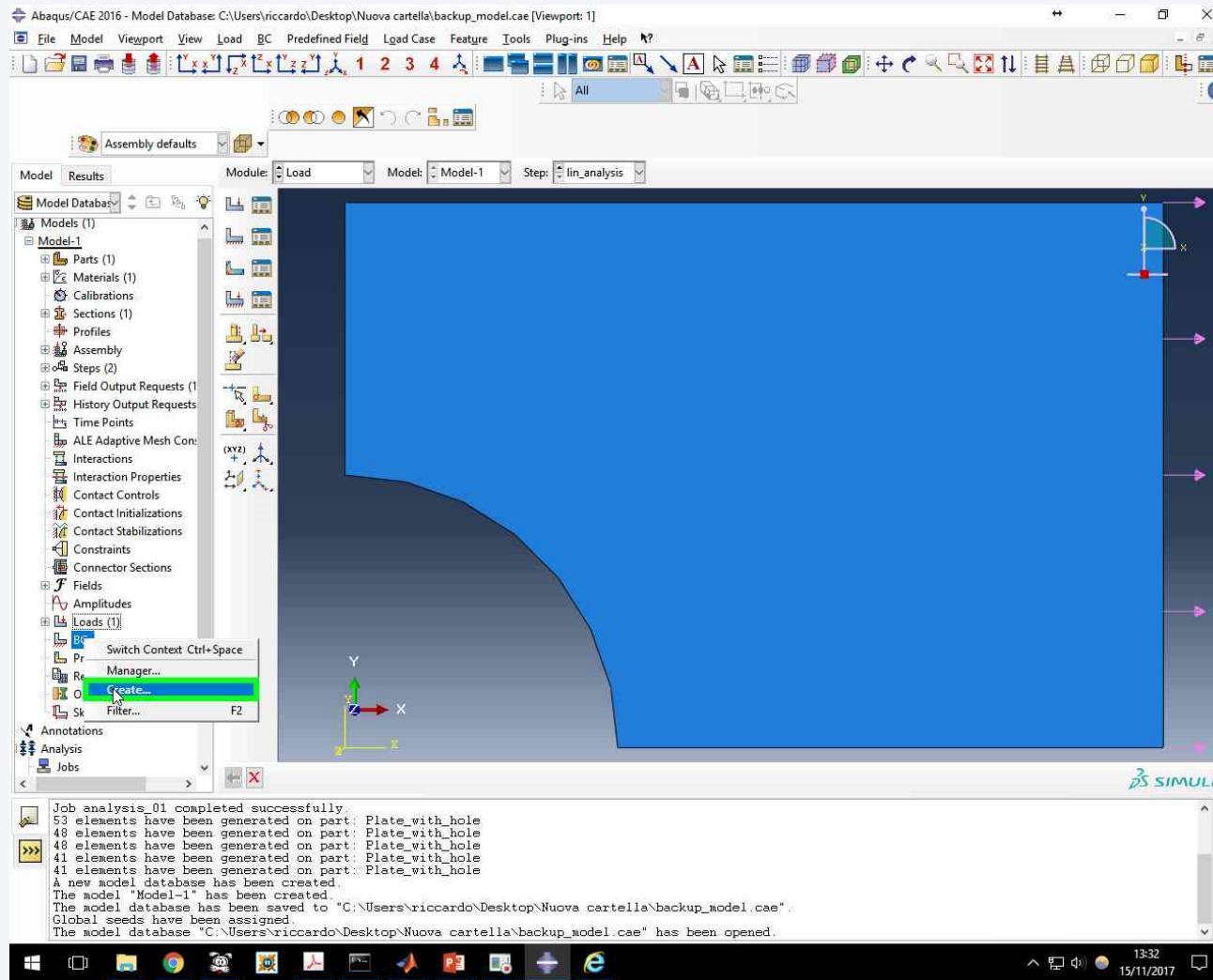
In this case, a vector describing the direction of the distributed force has to be specified. This is done by specifying the coordinates of the vector or by clicking over two points to define the vector.

Application of loads



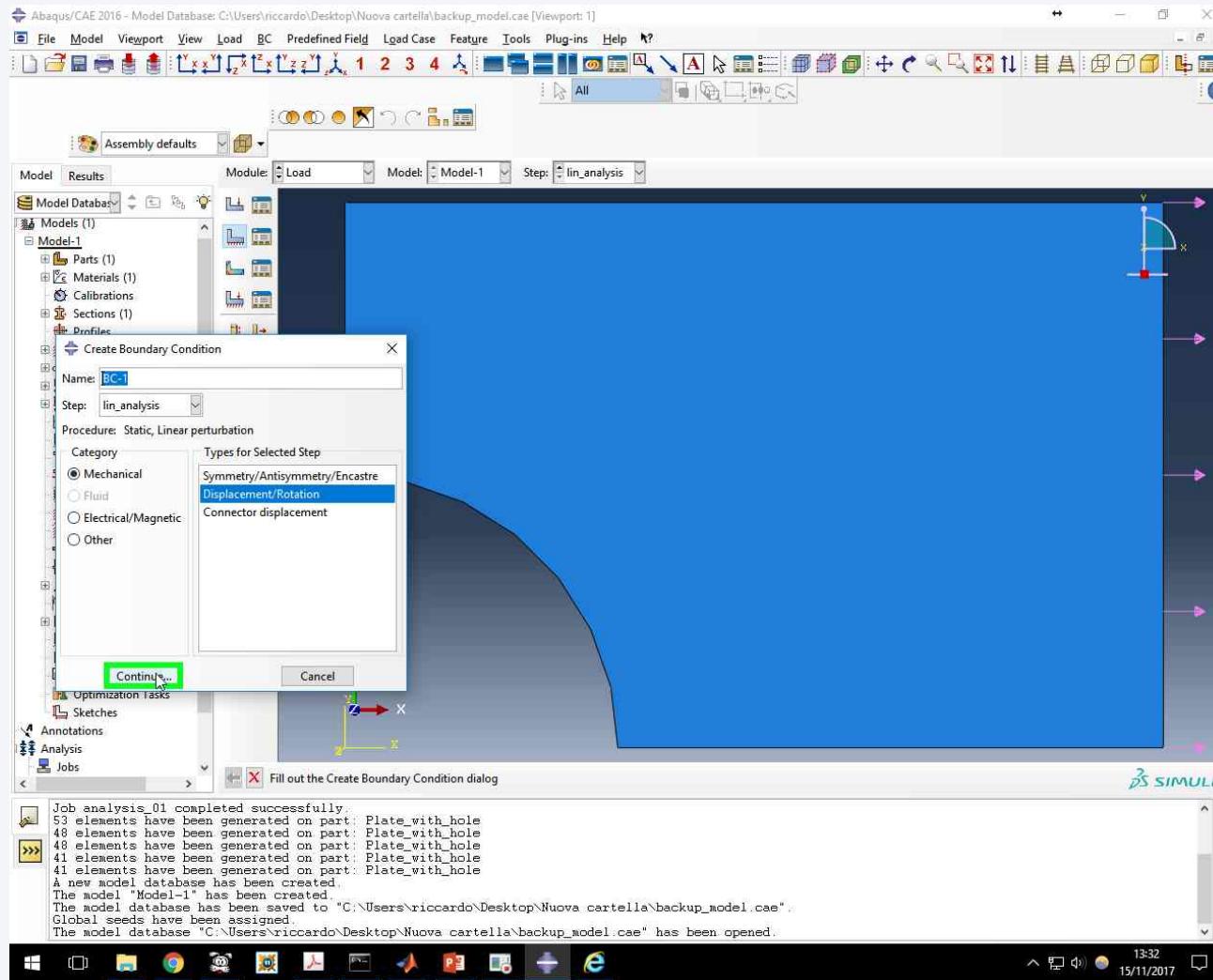
Confirm.

Definition of boundary conditions



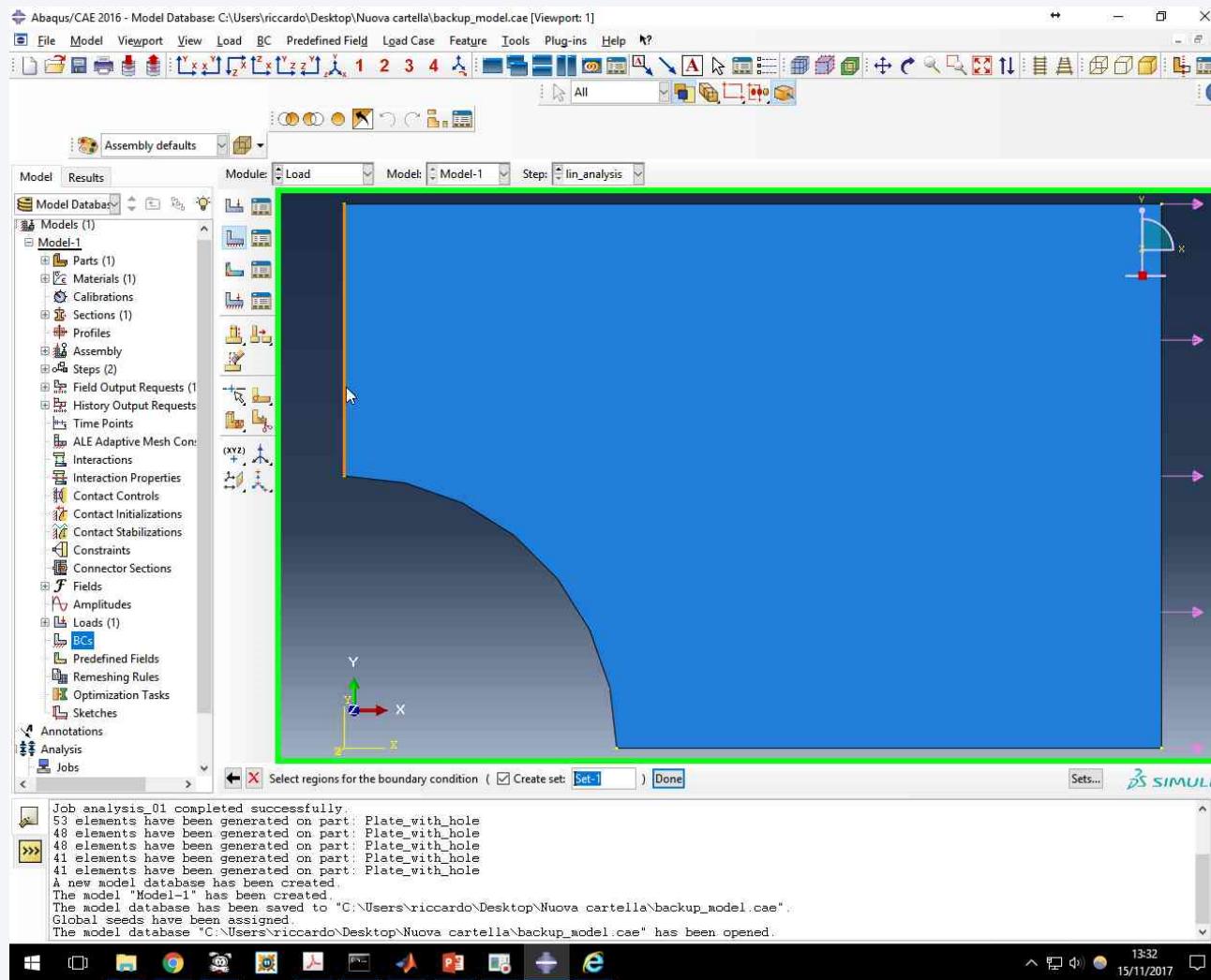
It is now necessary to specify the boundary conditions.

Definition of boundary conditions



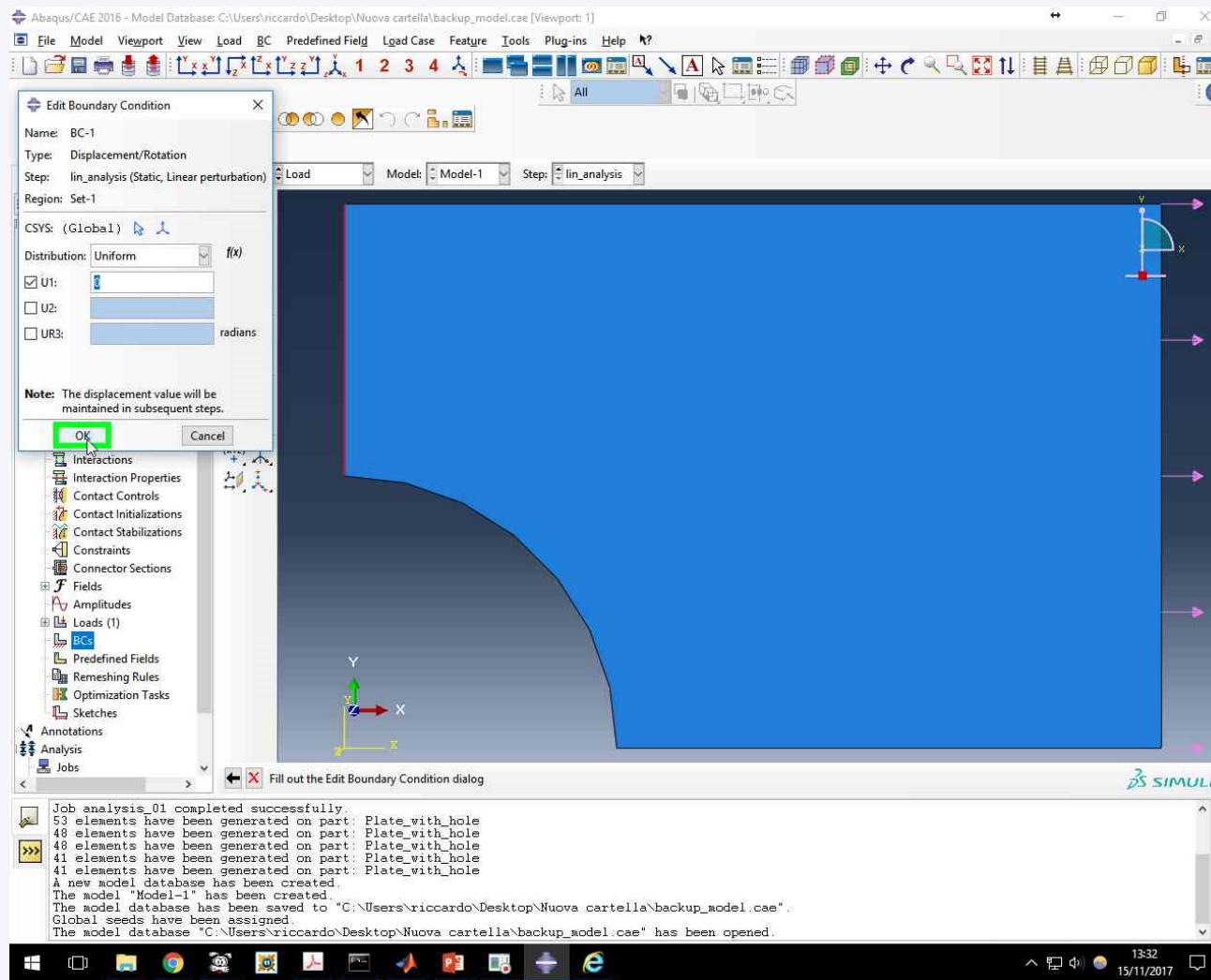
Select conditions of mechanical type and "Displacement/Rotatio n"

Definition of boundary conditions



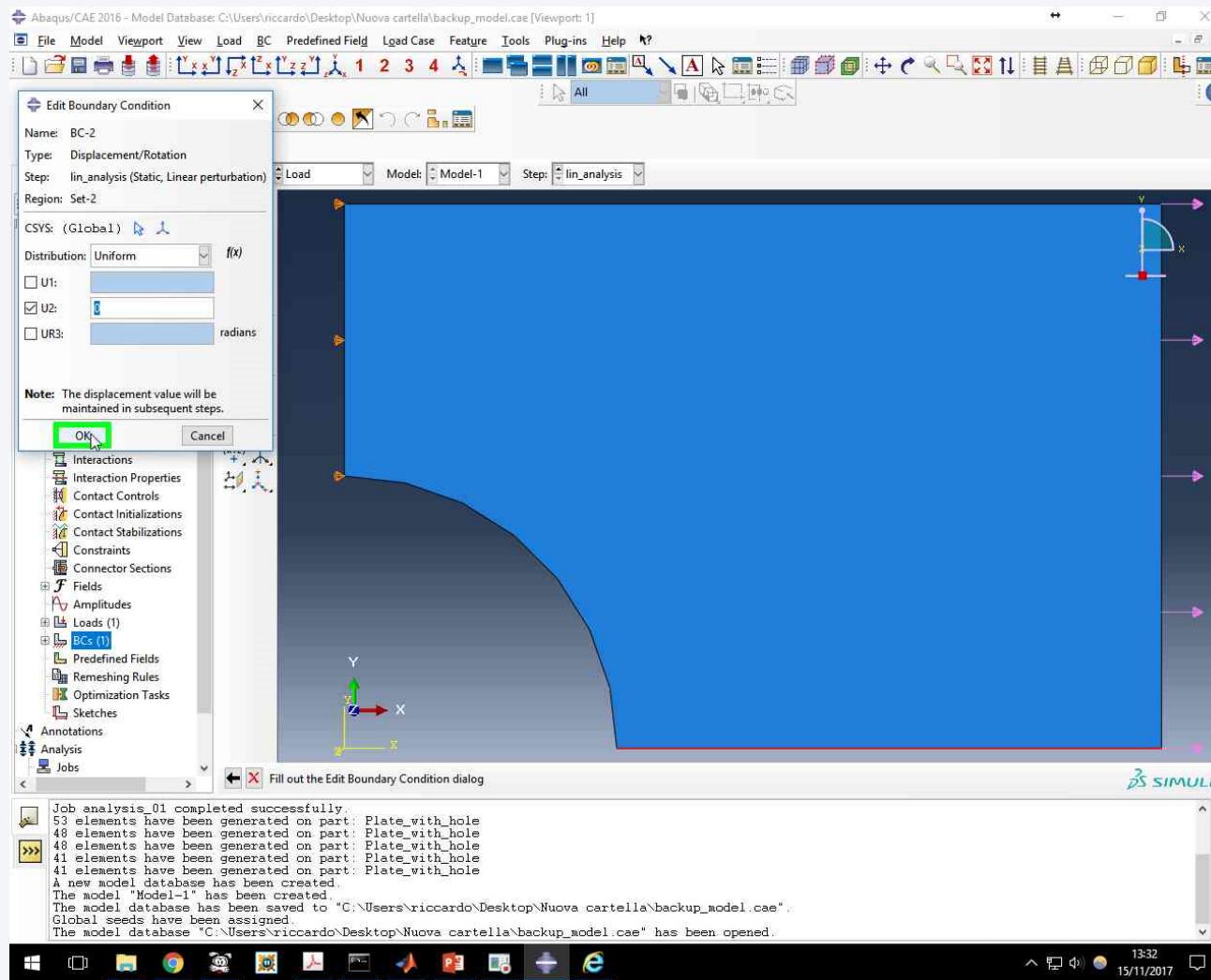
Confirm and select the first edge to be constrained.

Definition of boundary conditions



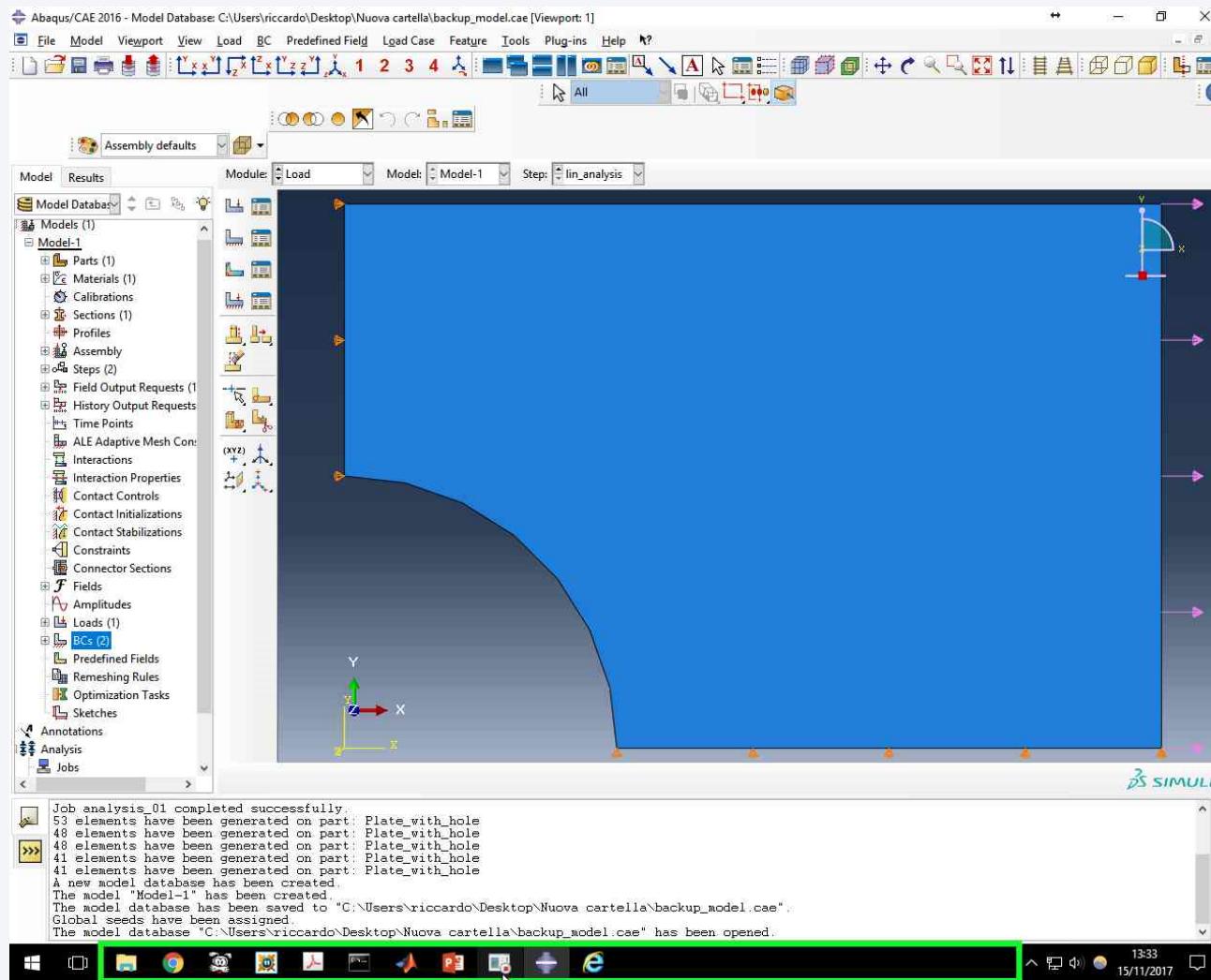
Confirm and specify the components of the displacement to be constrained.

Definition of boundary conditions



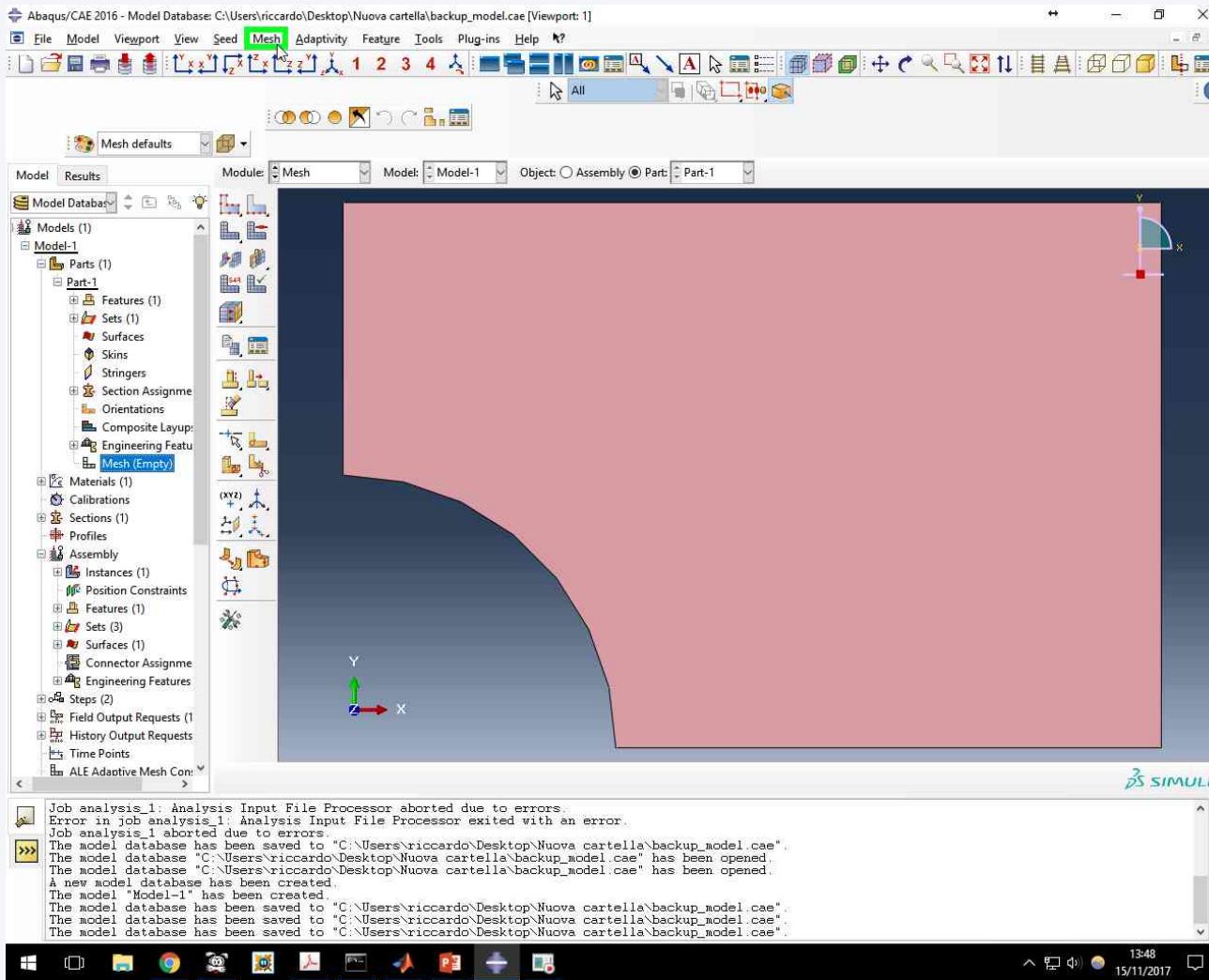
Repeat the same steps for the second edge to be constrained.

Definition of boundary conditions



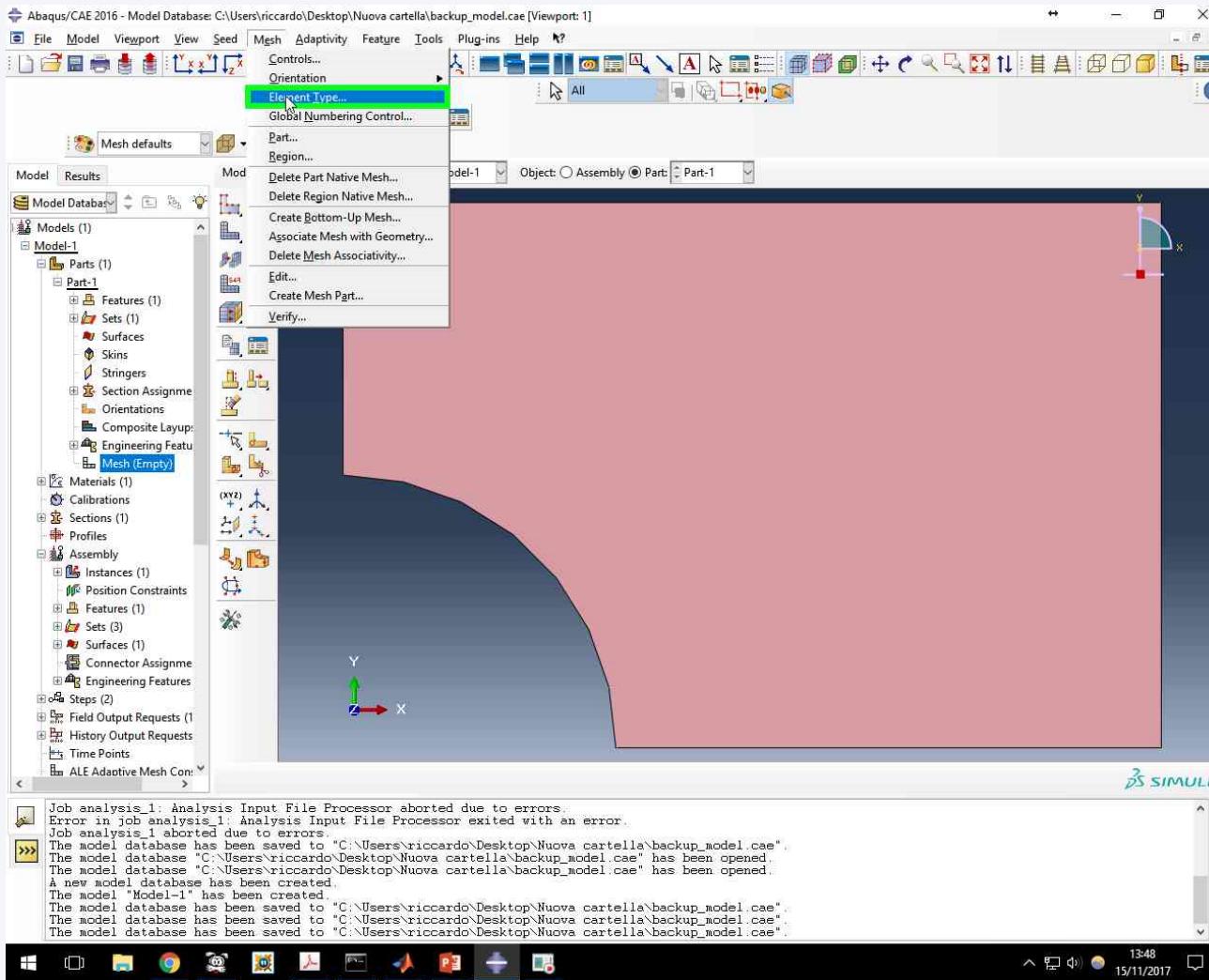
The model should now appear as reported.

Creating the mesh



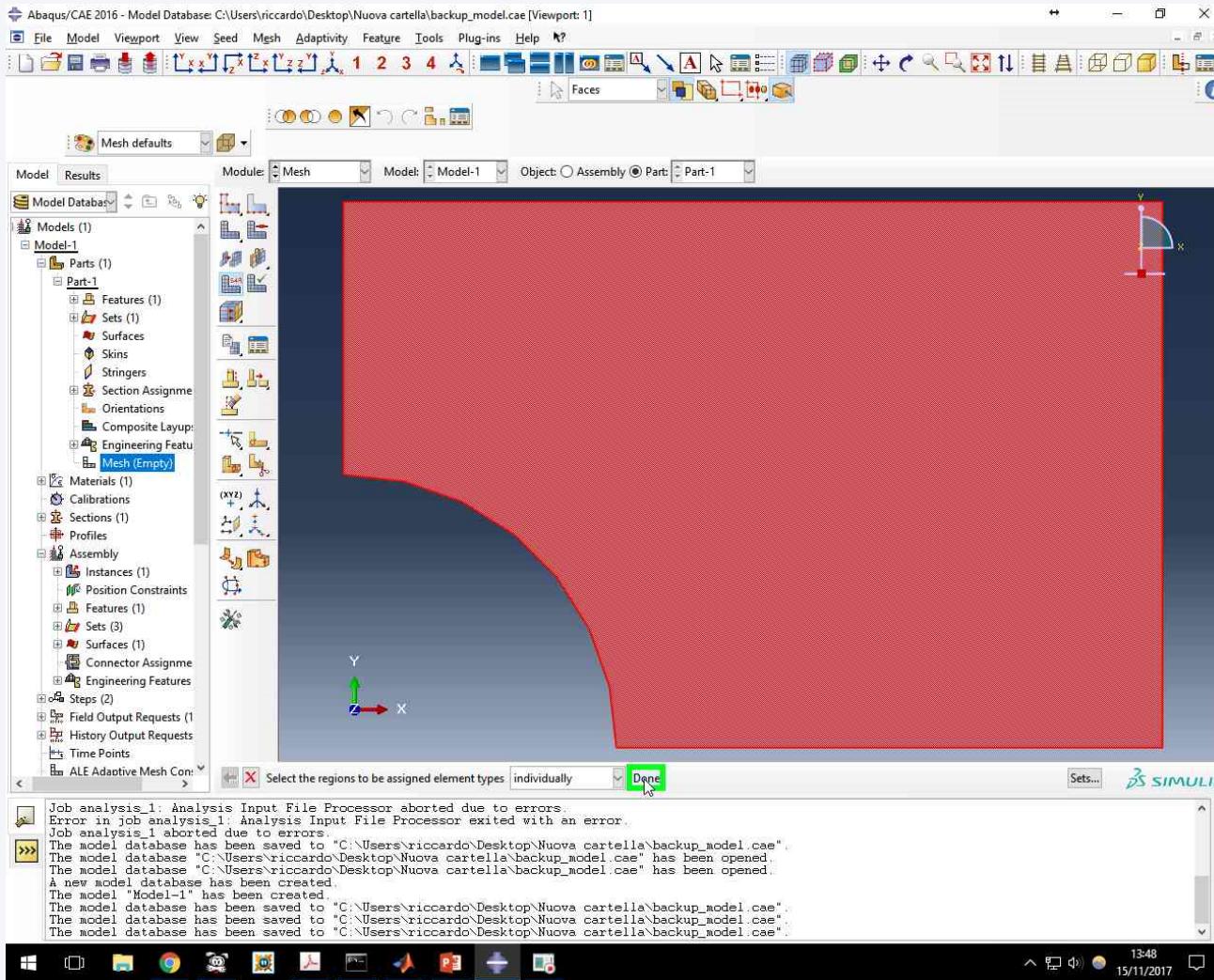
Move now to the mesh menu.

Creating the mesh



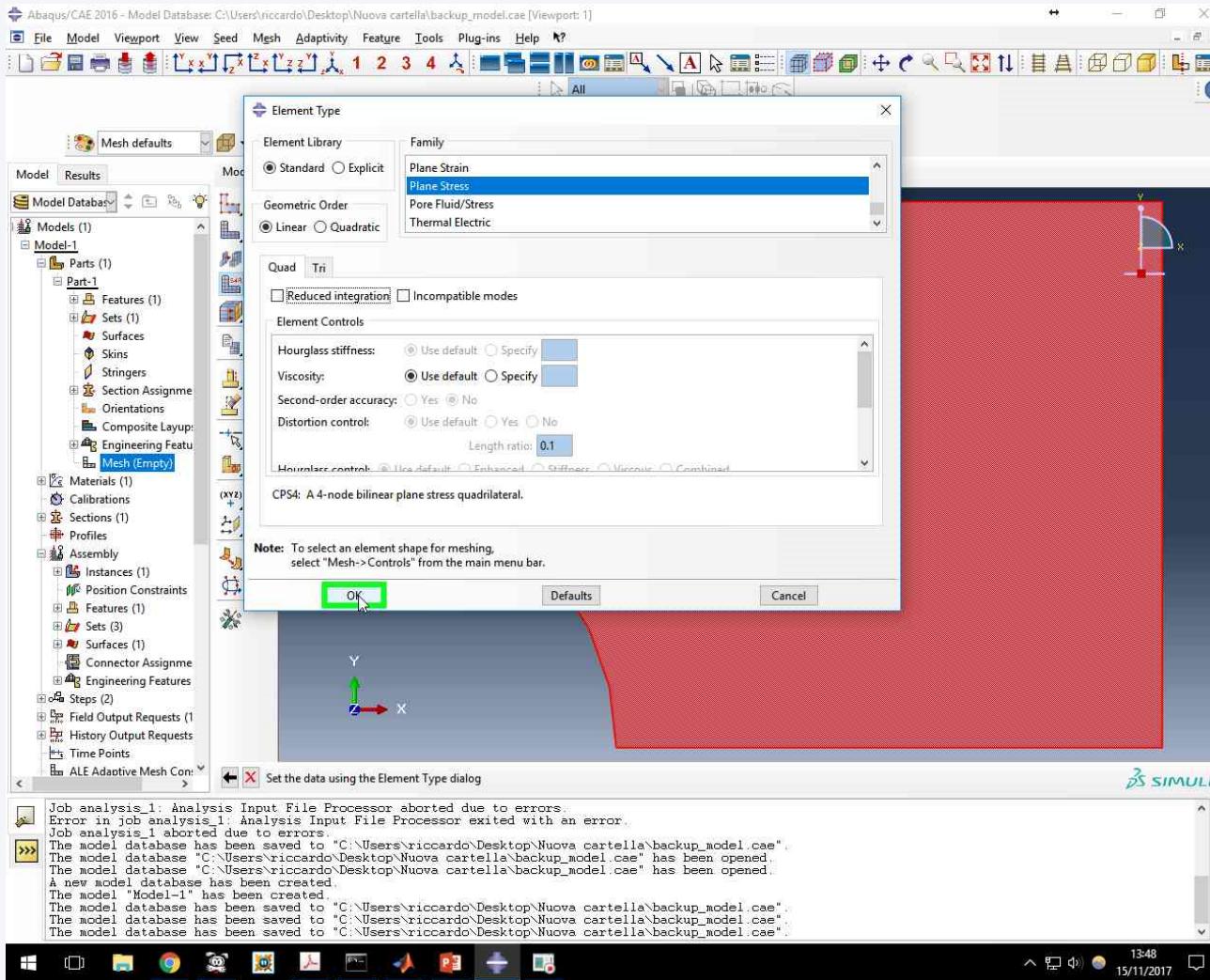
Select "Element type" to specify the kind of element to be used.

Creating the mesh



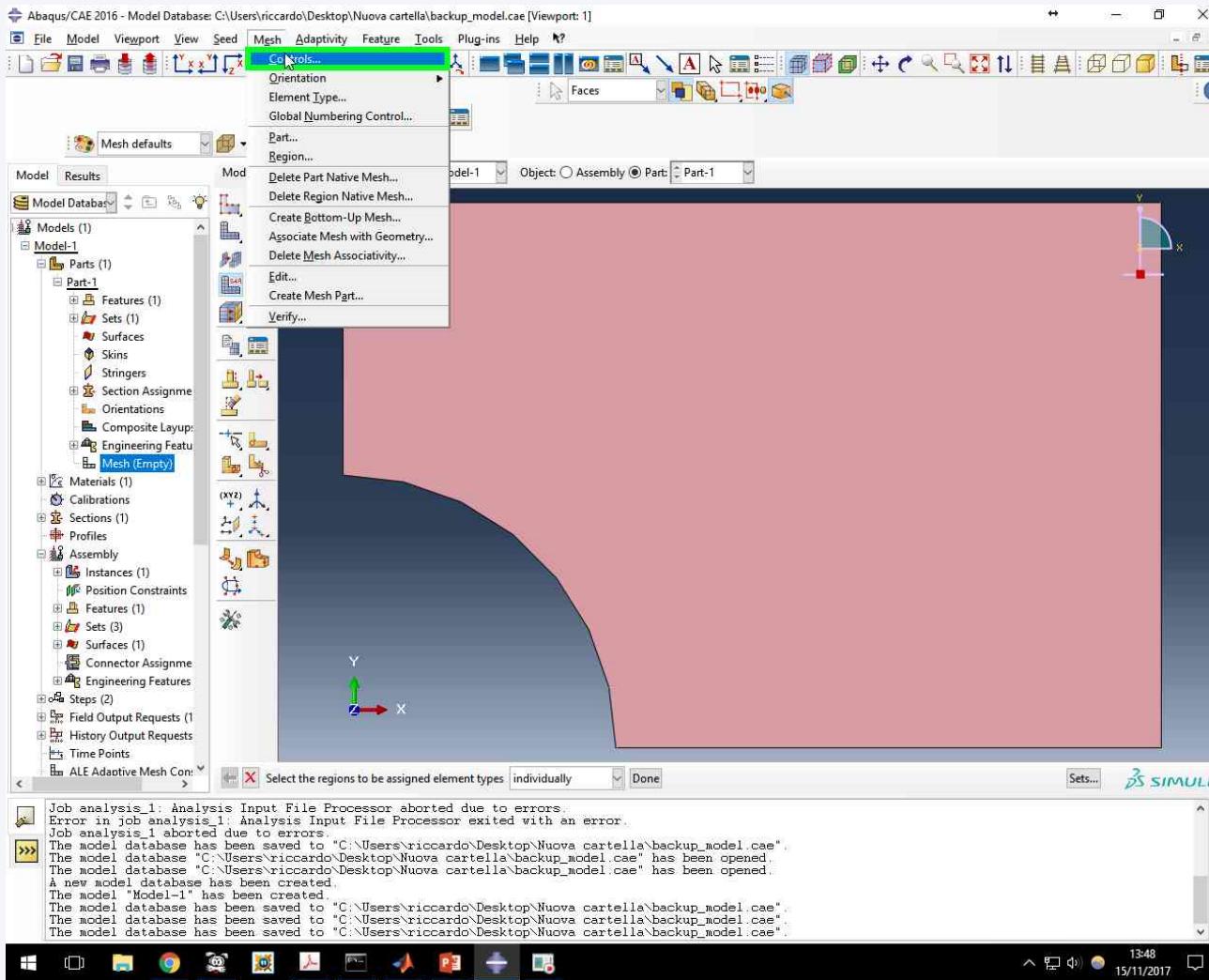
Select the part that will be associated to the specified kind of finite element.

Creating the mesh



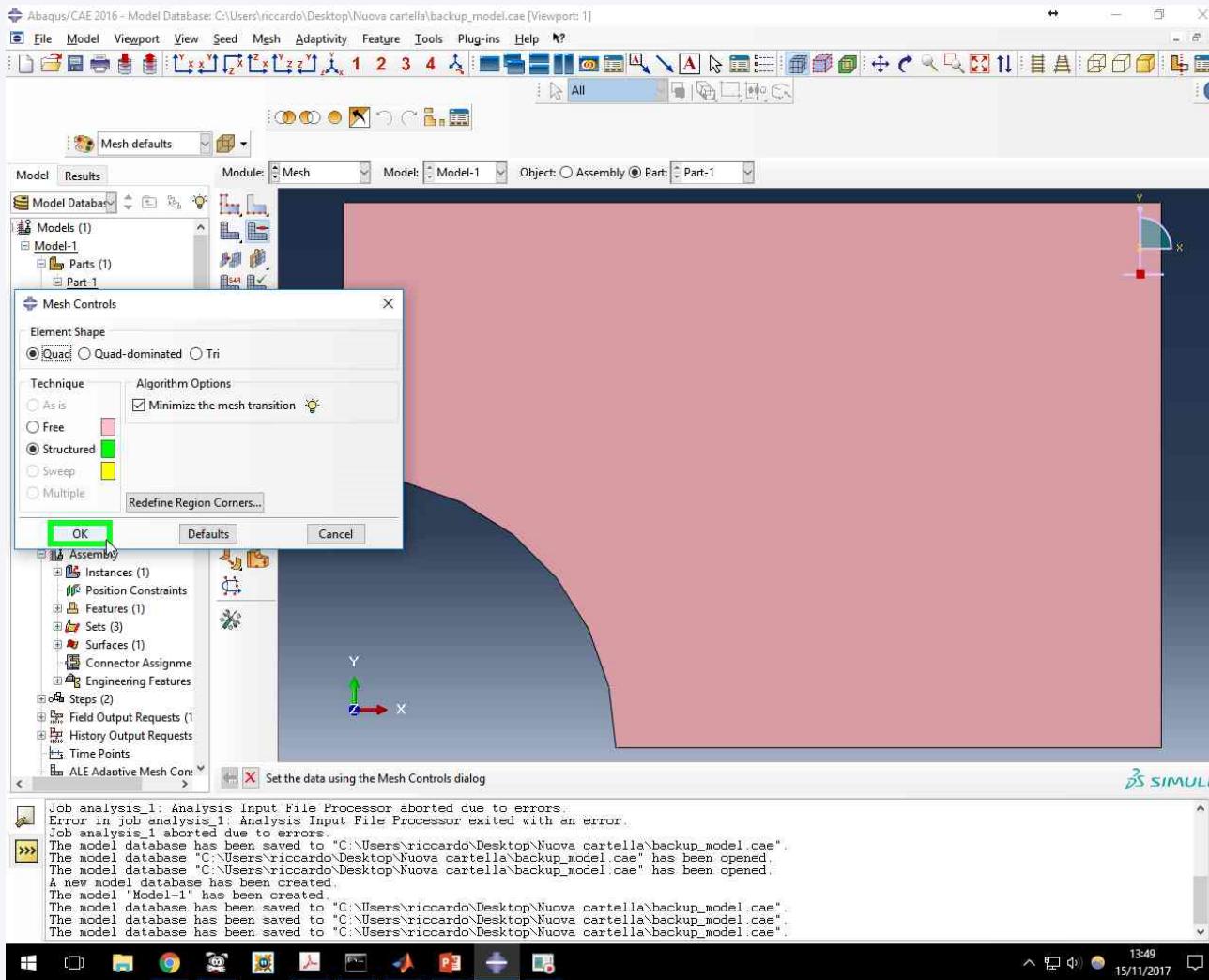
Many possibilities are available. In this case a membrane plane stress element is adopted.

Creating the mesh



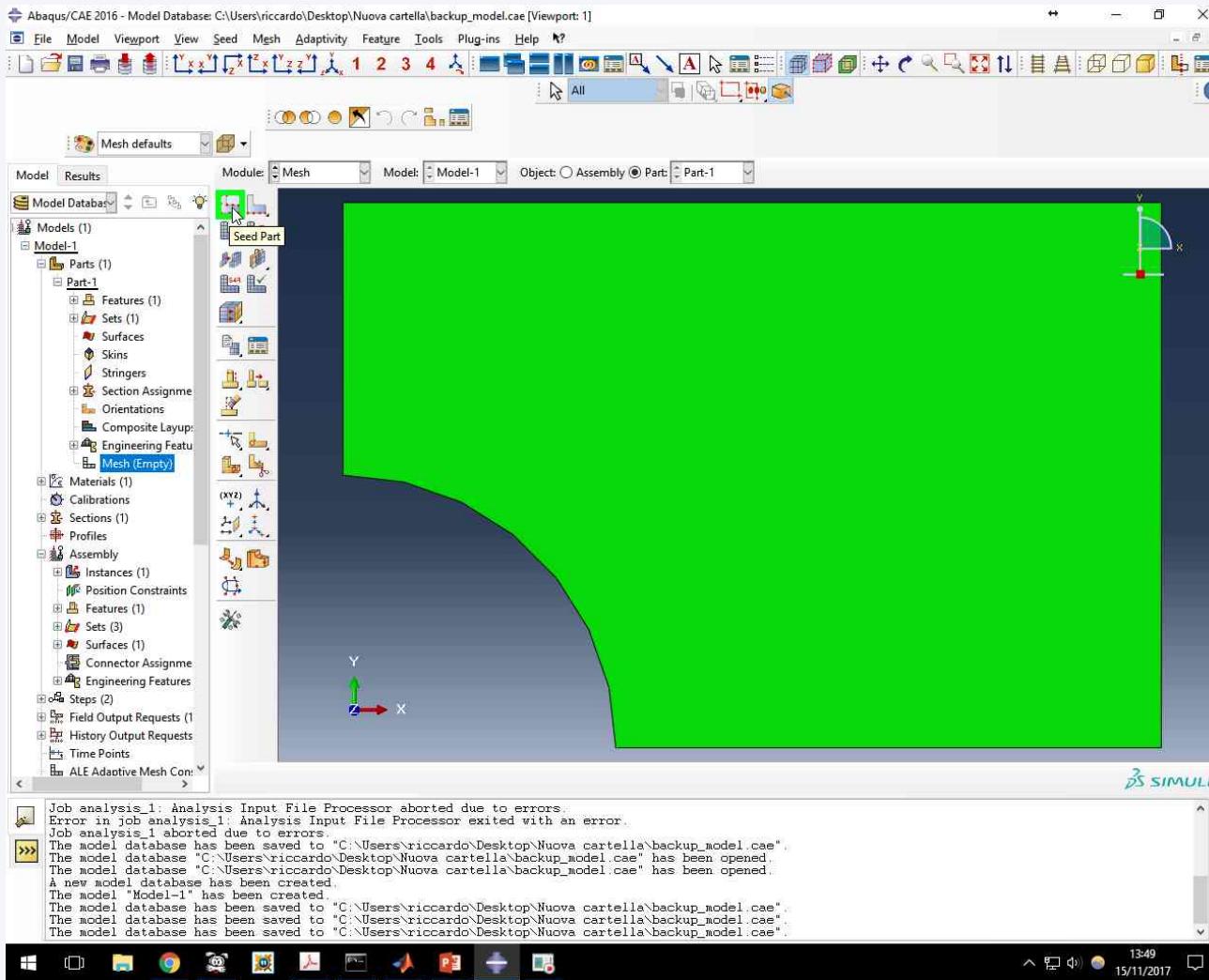
Specify some controls over the meshing rules.

Creating the mesh



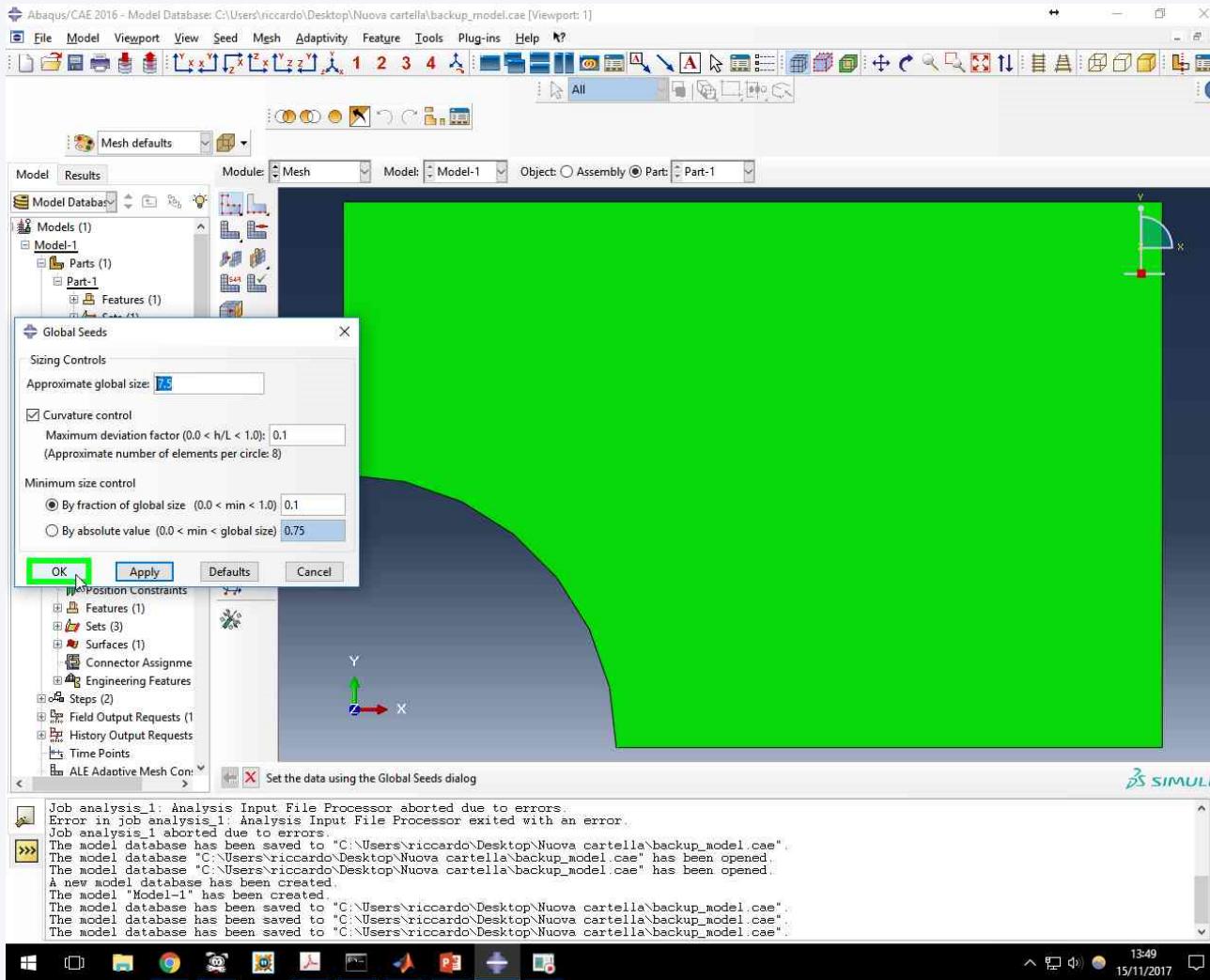
Select the options illustrated.

Creating the mesh



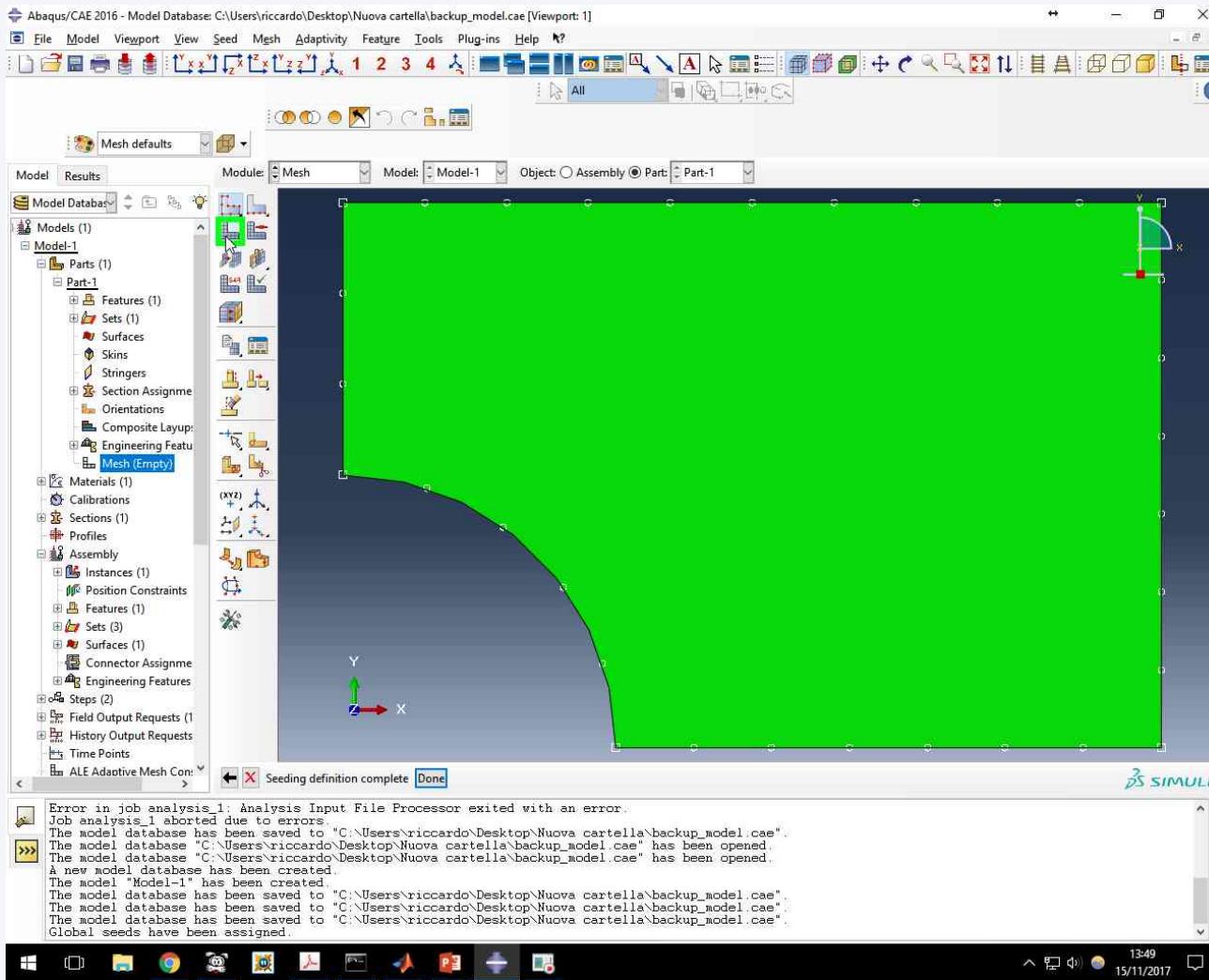
Specify the number of elements and nodes, for now, with the seed command.

Creating the mesh



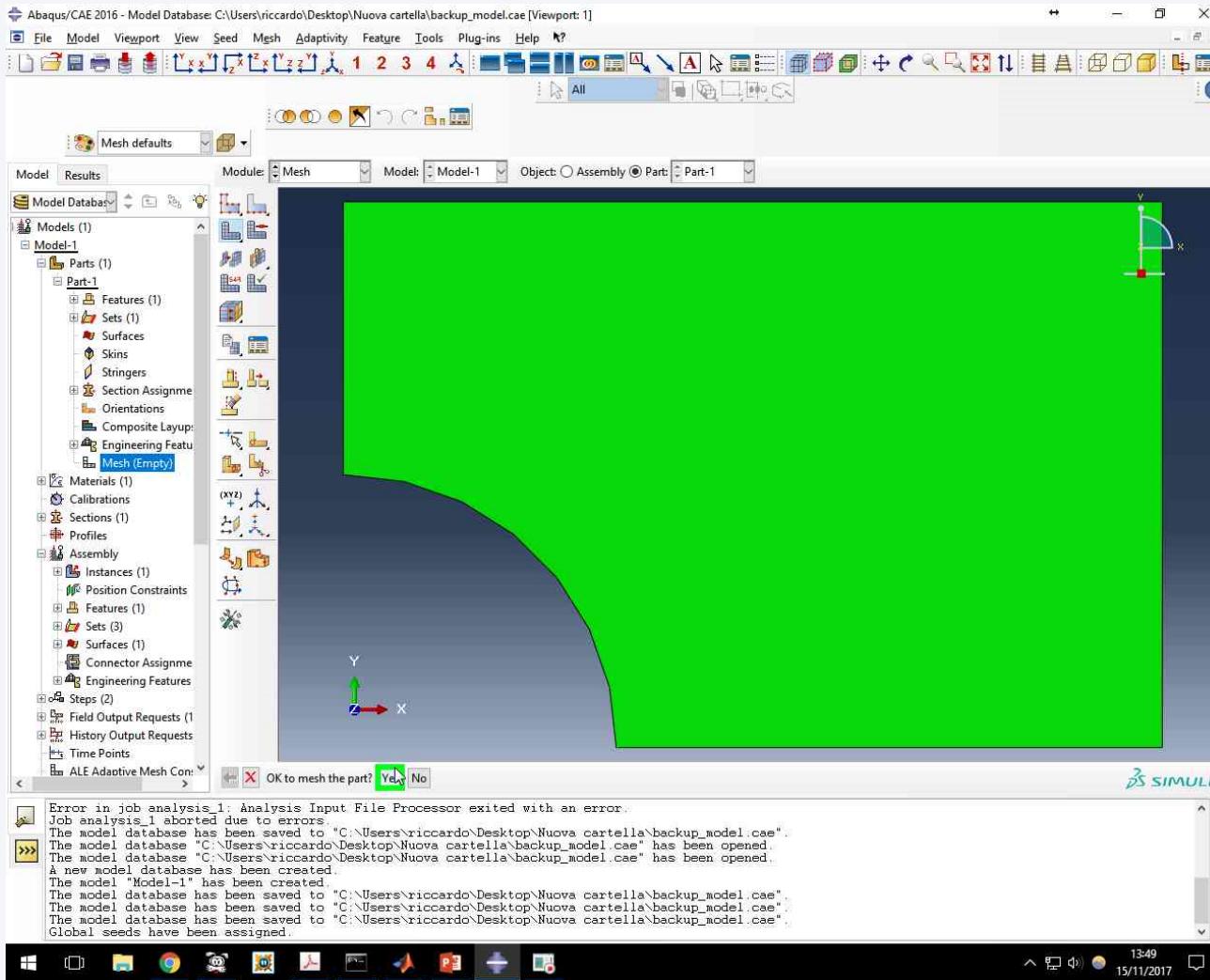
Accept the default options.

Creating the mesh



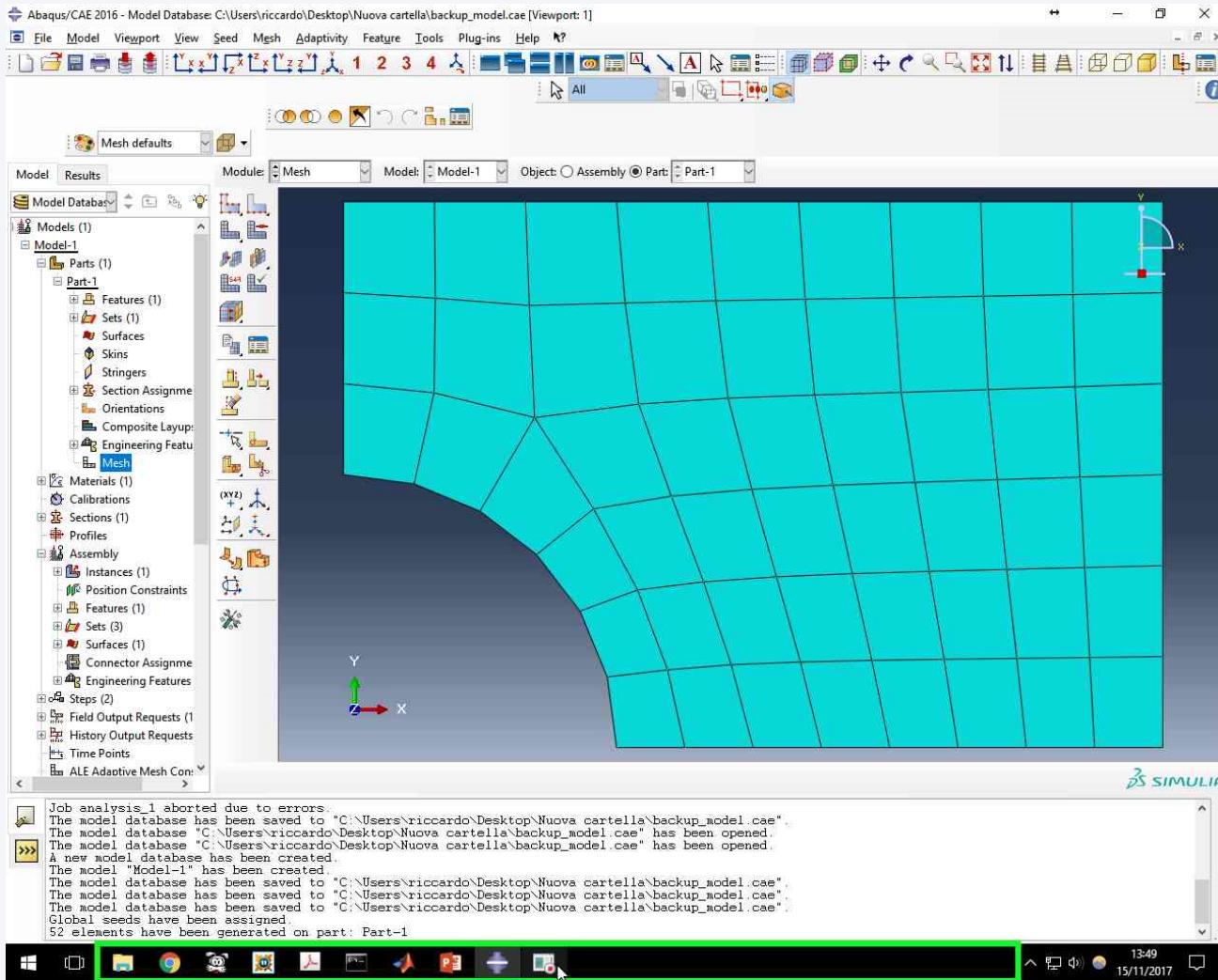
Realize now the mesh by clicking on "Mesh Part Instance".

Creating the mesh



Select the part and confirm.

Creating the mesh

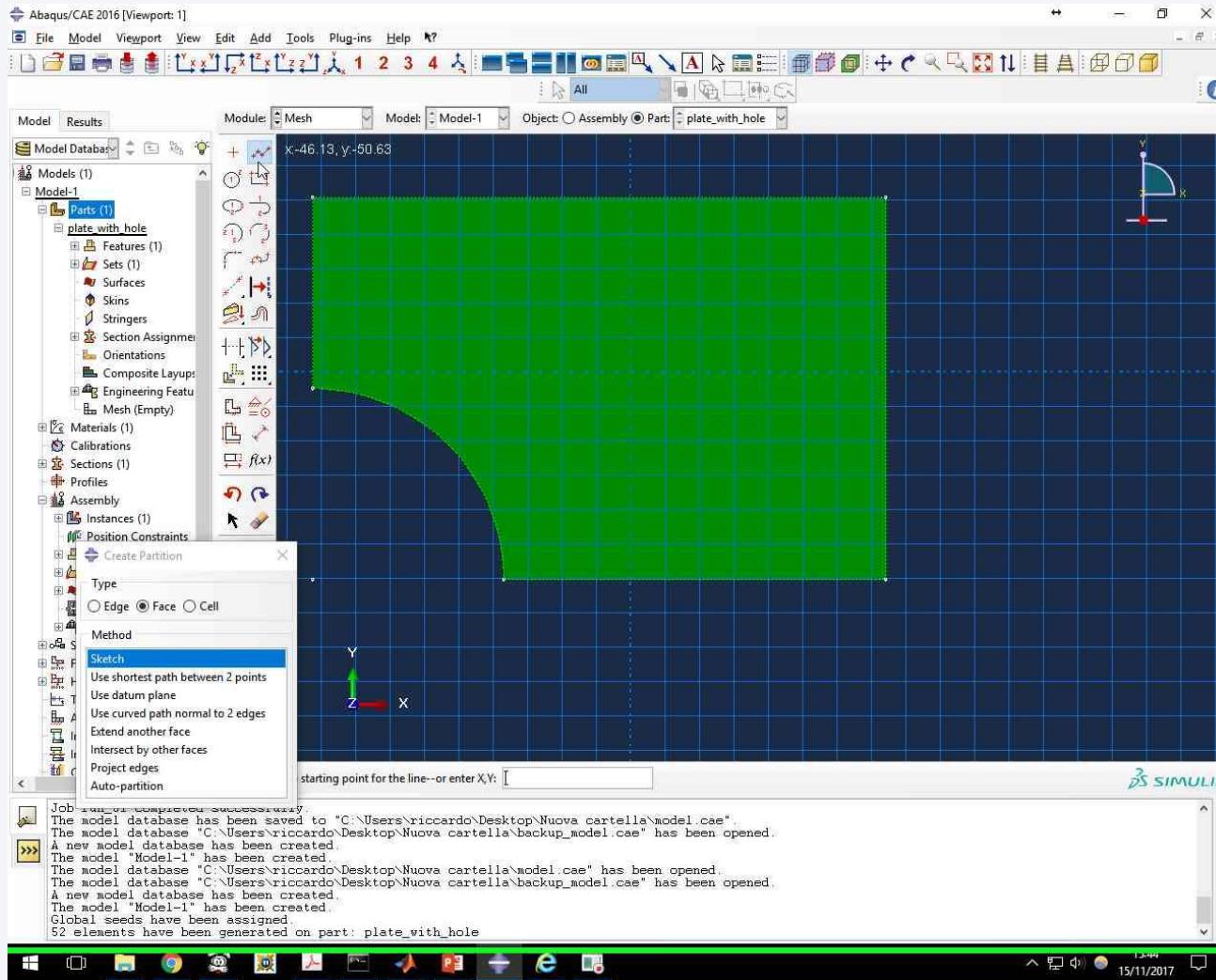


The mesh appear as reported.

In general, the direct meshing of the part may lead to unsatisfactory meshes. For instance, some elements could be excessively distorted.

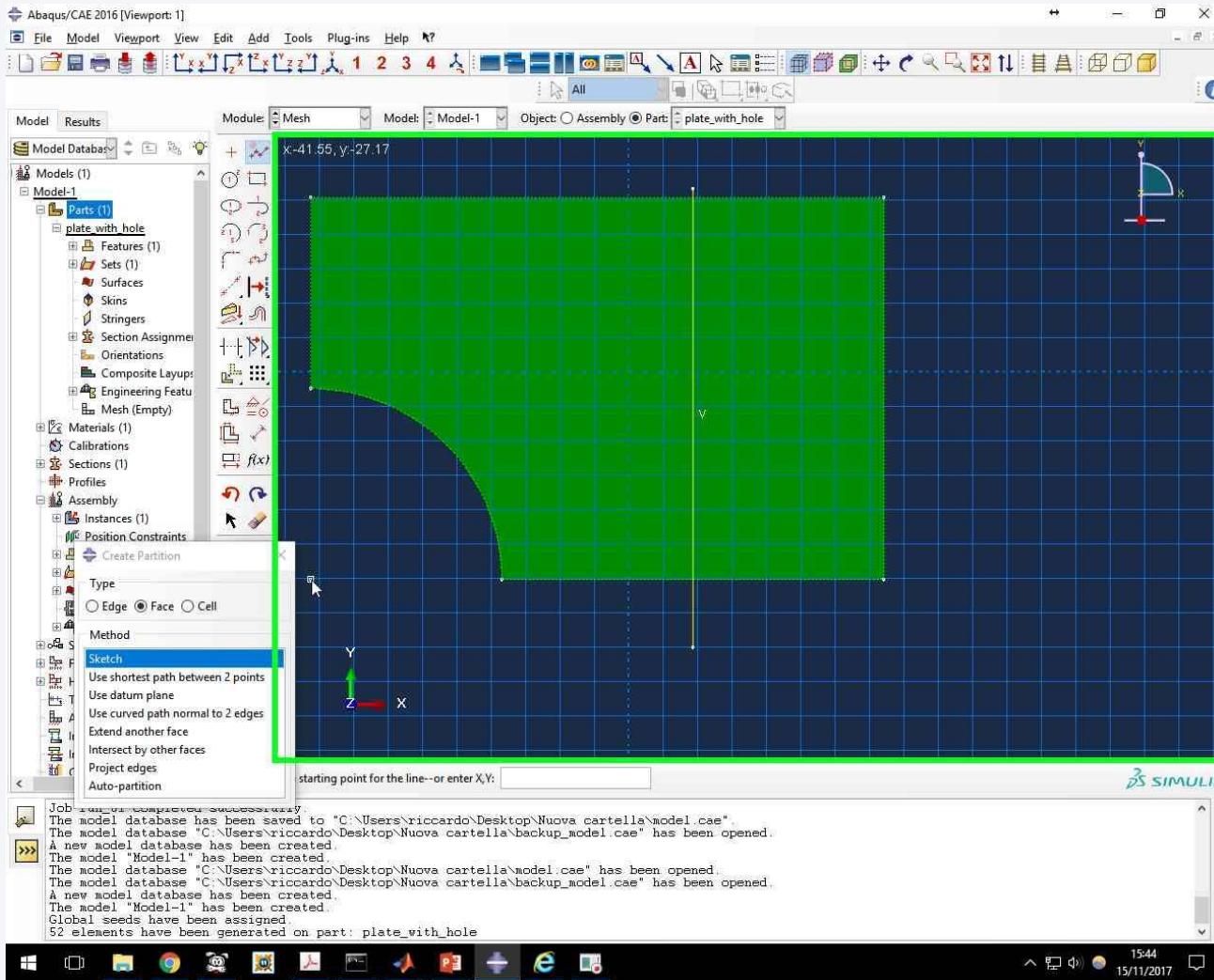
To this aim a good approach is to divide the part into a number of simpler sub-parts to be meshed independently (see later).

Remeshing using partitions



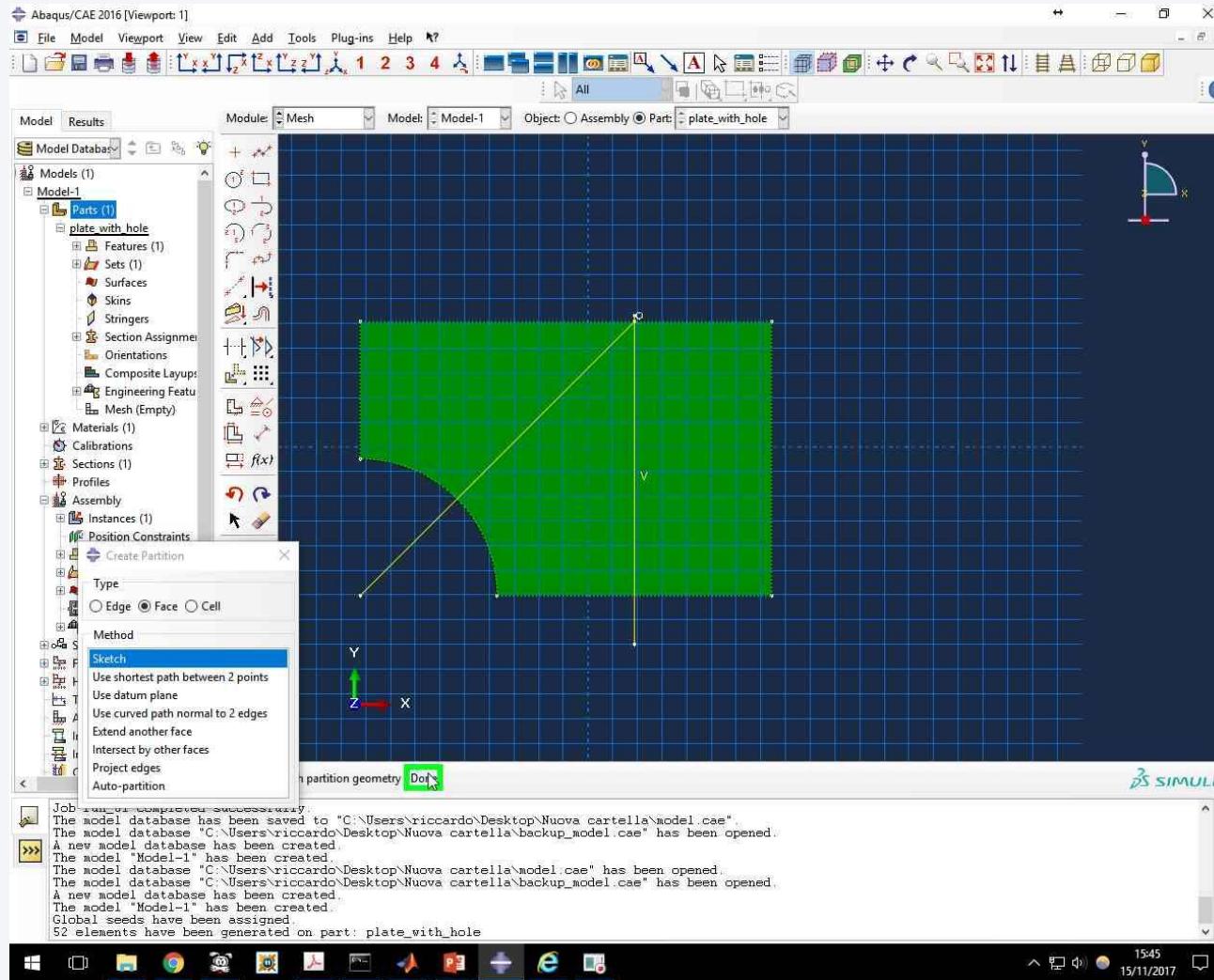
Using the command Tools → Partition the part can be divided into simpler entities.

Remeshing using partitions



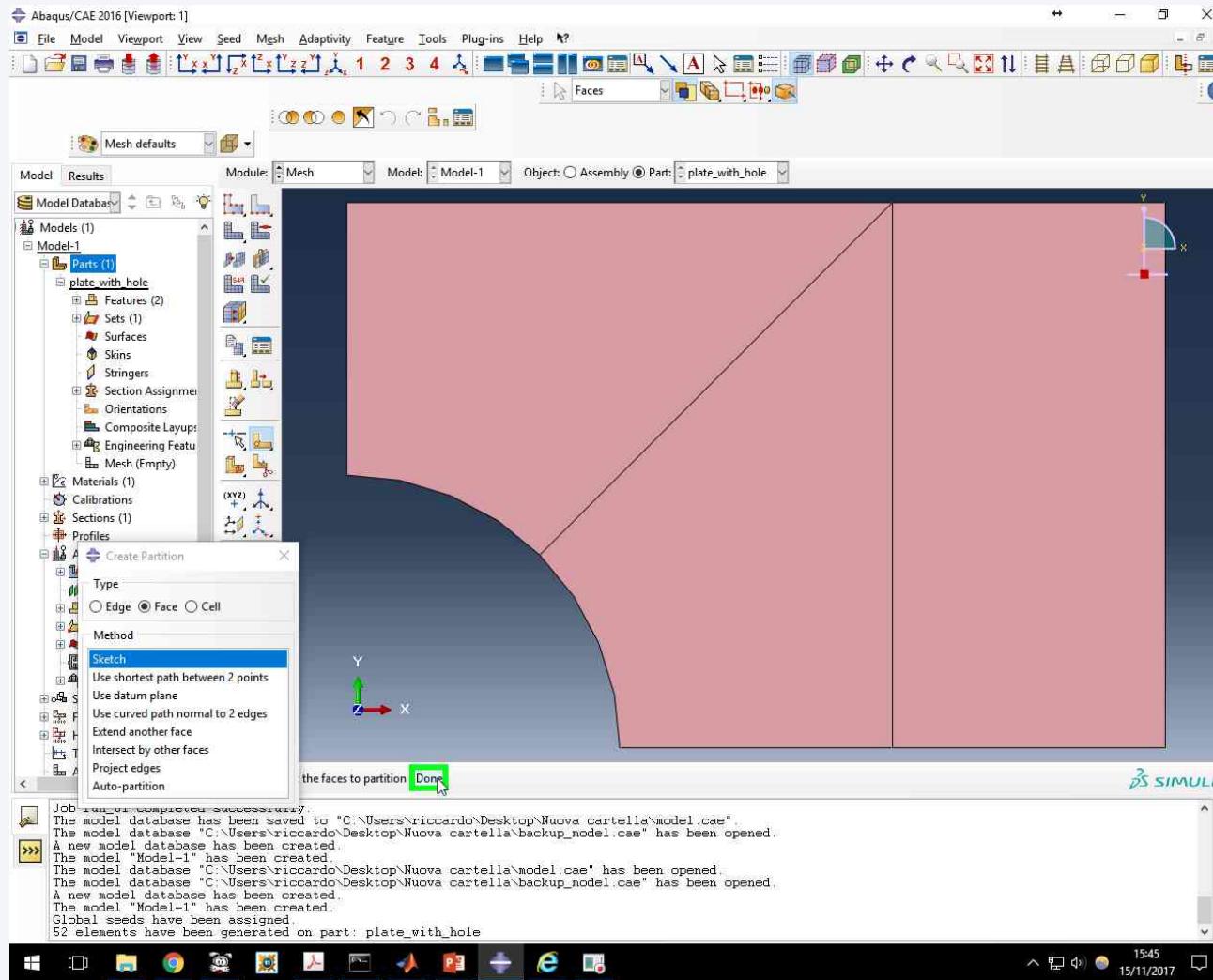
In this case a natural choice consists in dividing the plate into three subdomains.

Remeshing using partitions



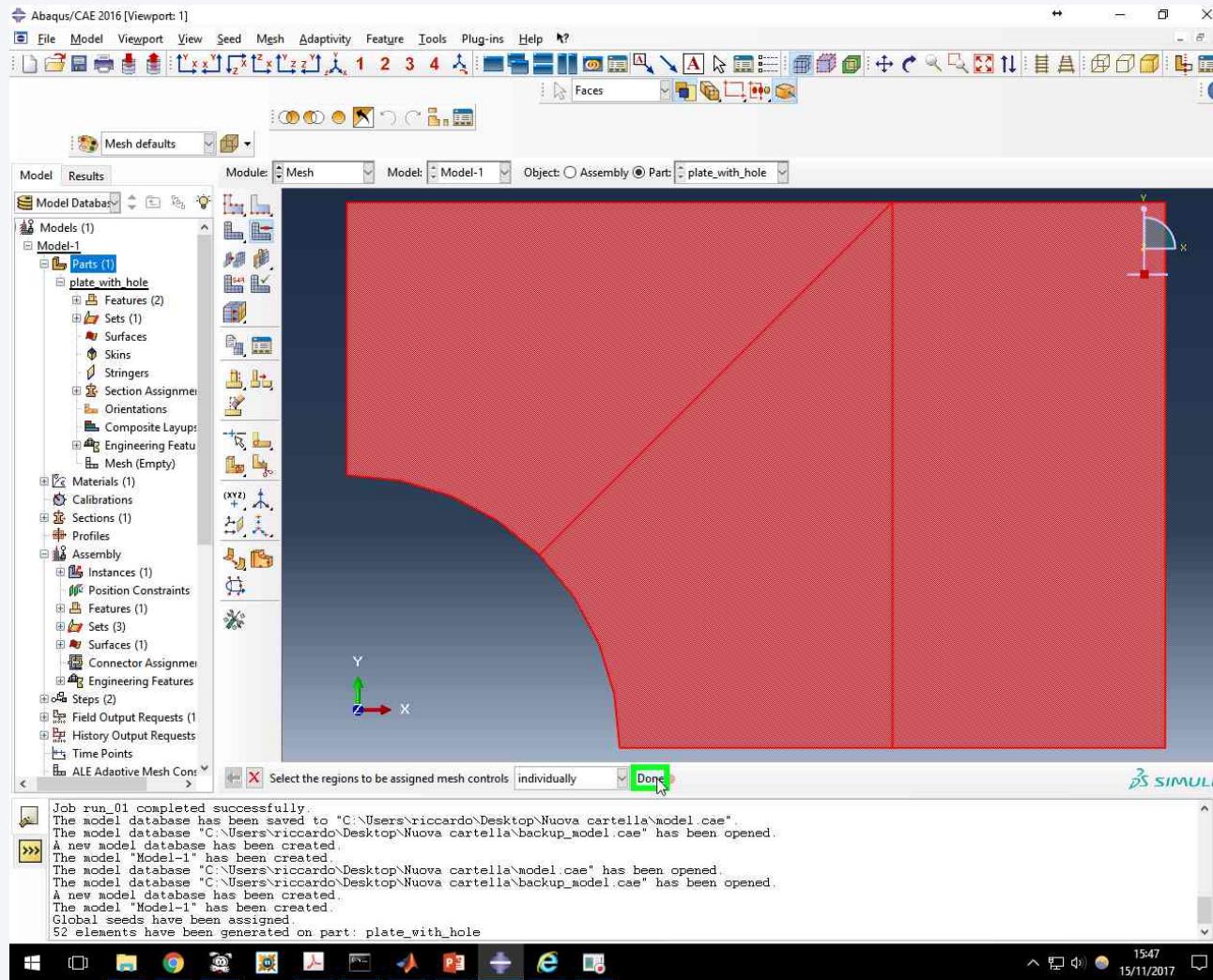
In this case a natural choice consists in dividing the plate into three subdomains.

Remeshing using partitions



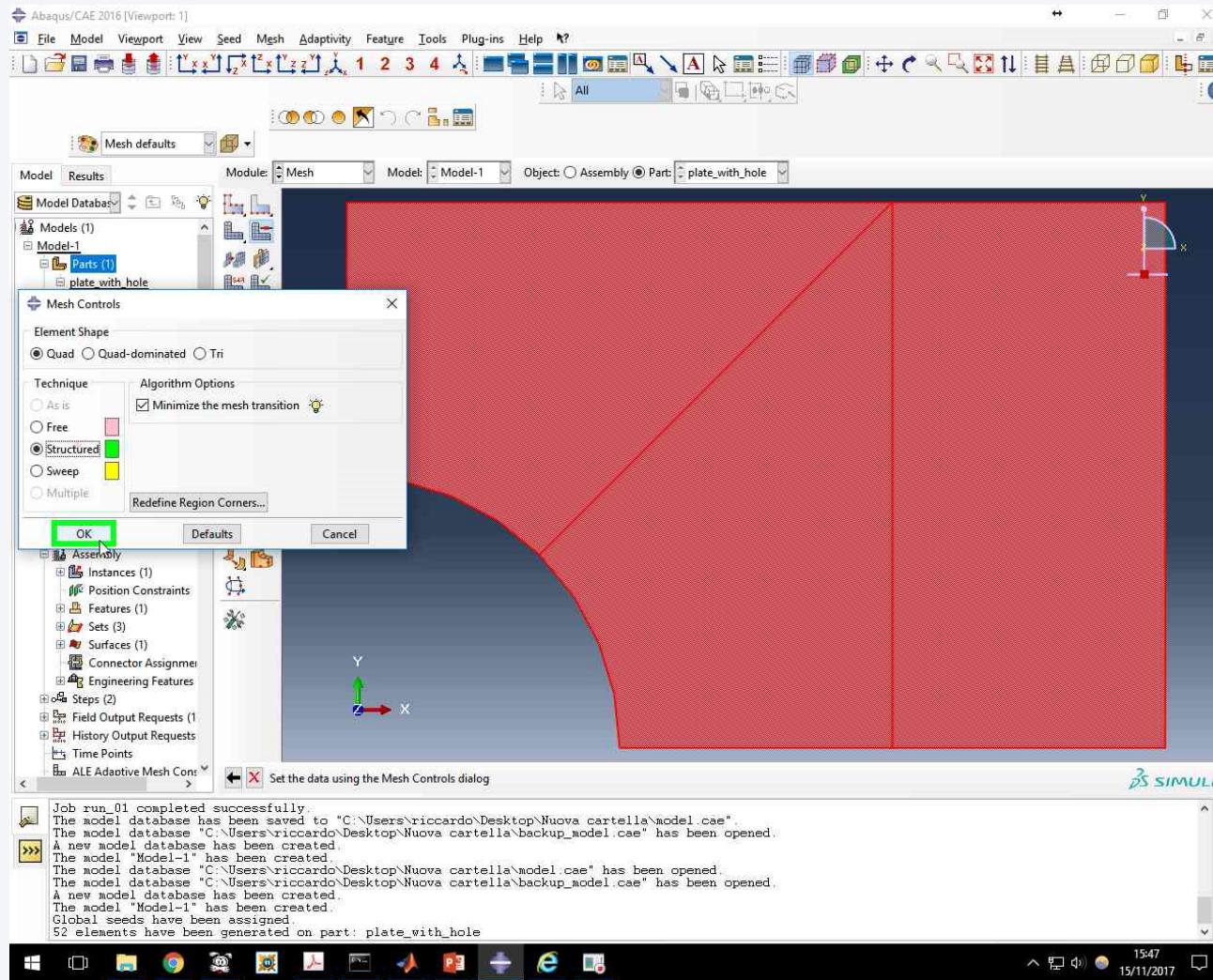
The subdivision looks now as reported.

Remeshing using partitions



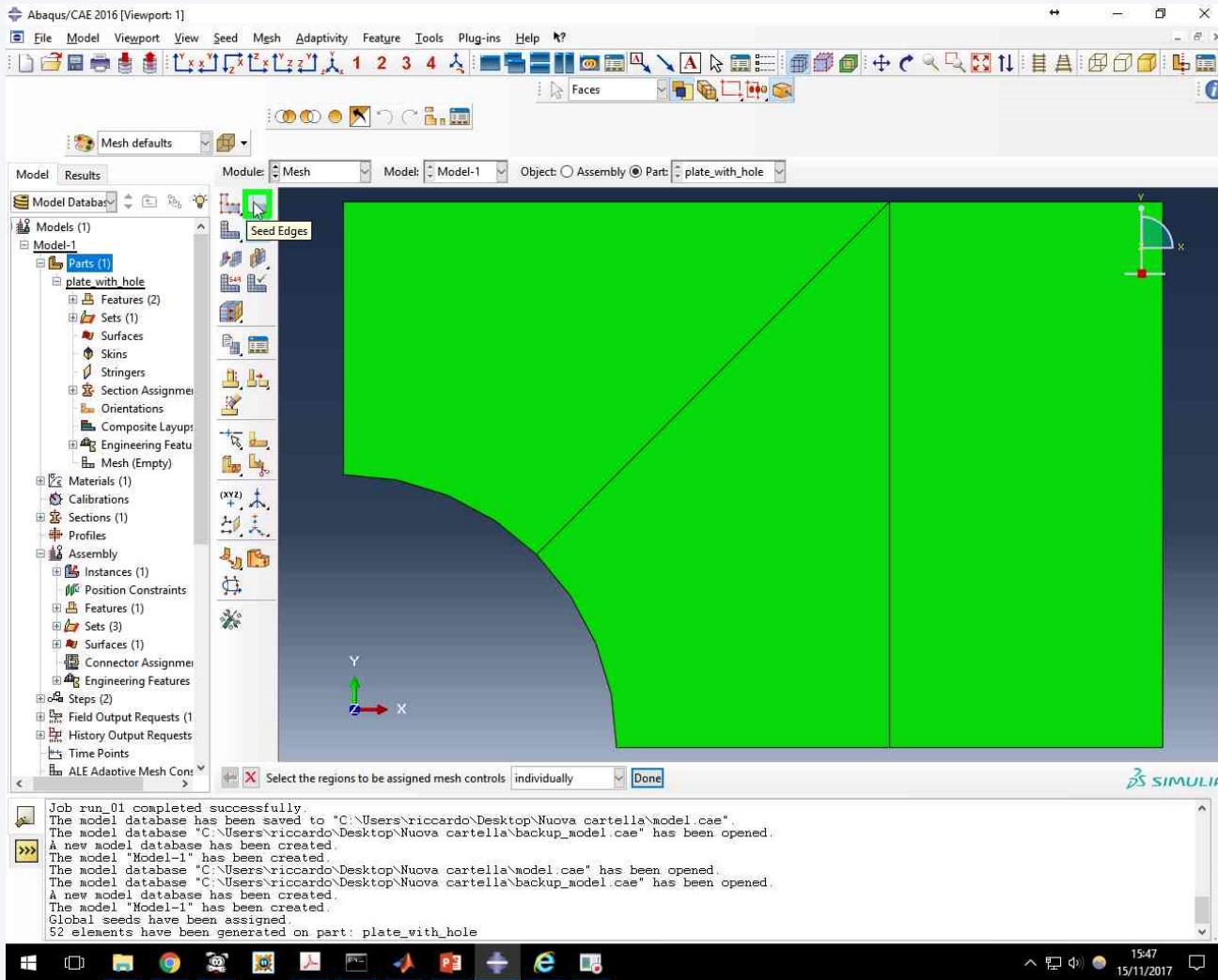
As far as the part has been modified, the mesh controls should be re-defined

Remeshing using partitions



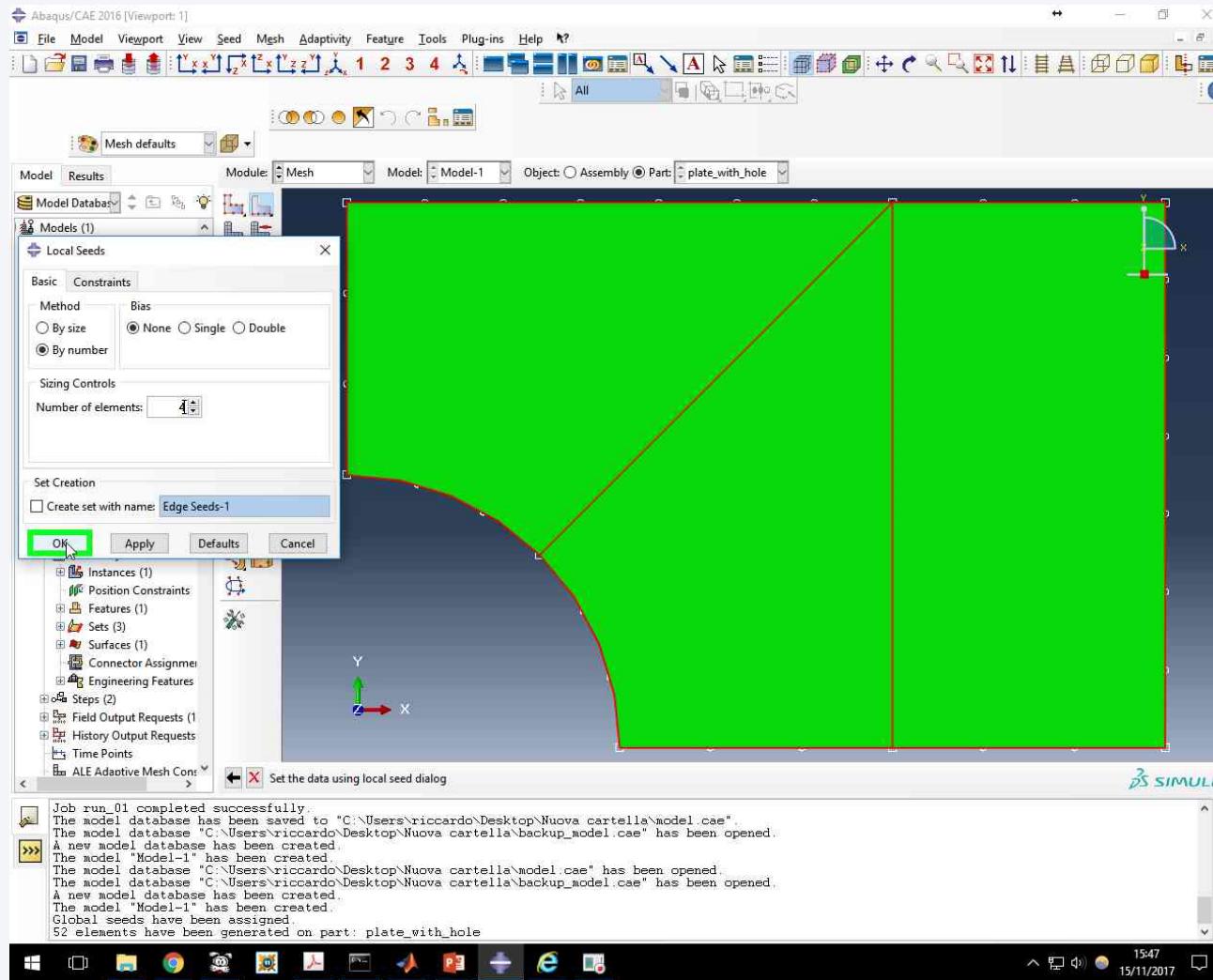
Select again the same options.

Remeshing using partitions



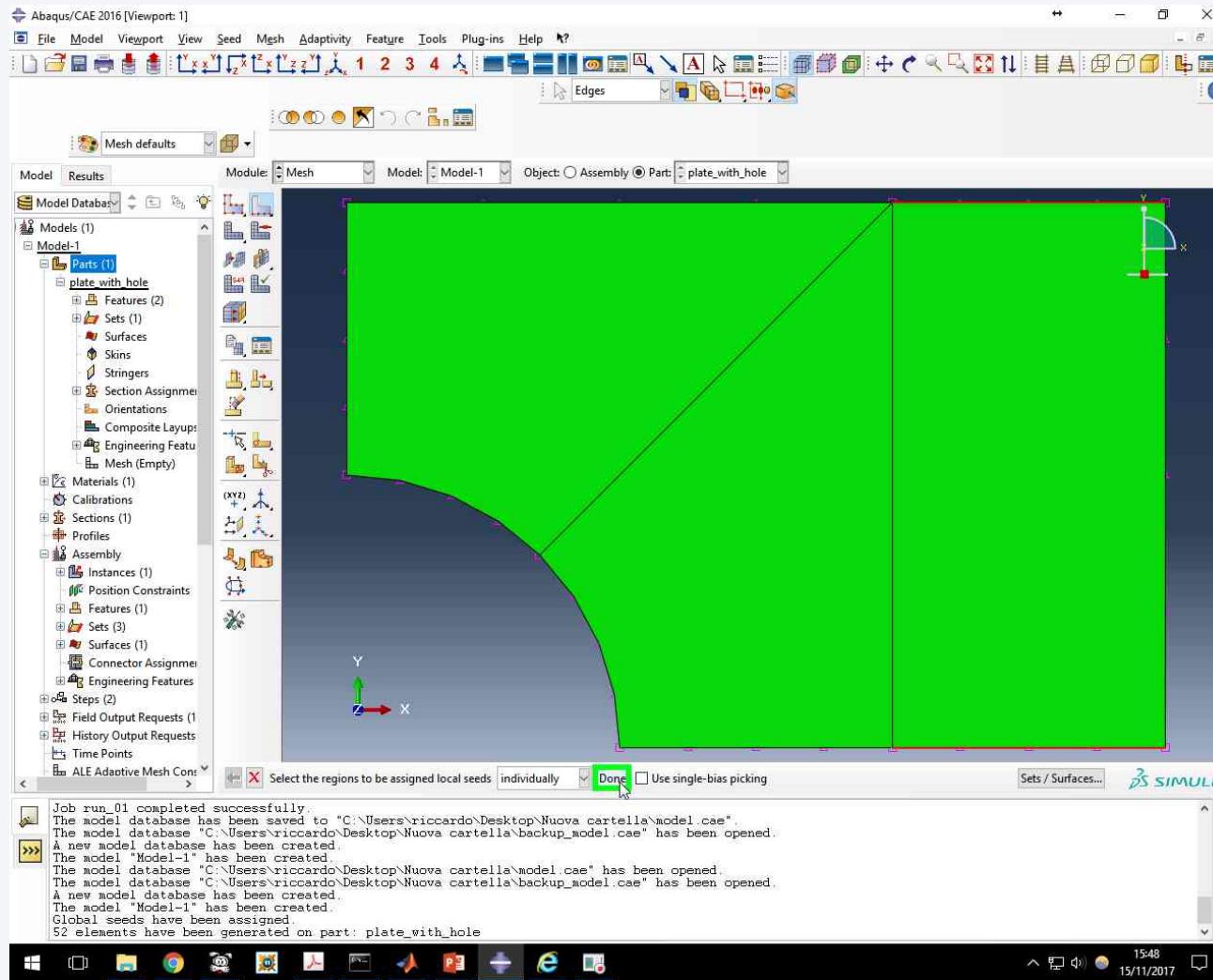
Define now the number of nodes along each of the lines composing the boundaries of the three domains.

Remeshing using partitions



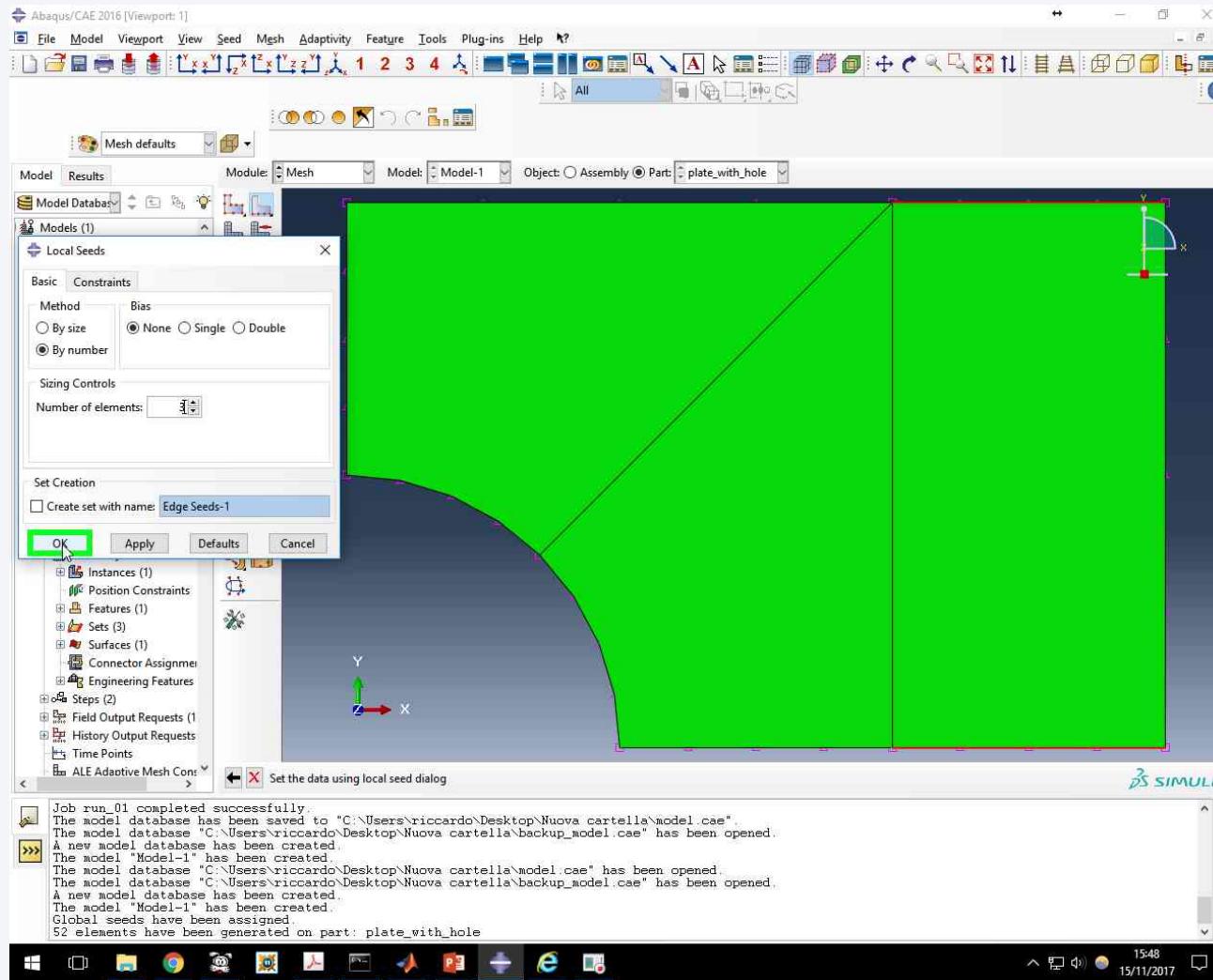
Select 4 elements for edges here highlighted.

Remeshing using partitions



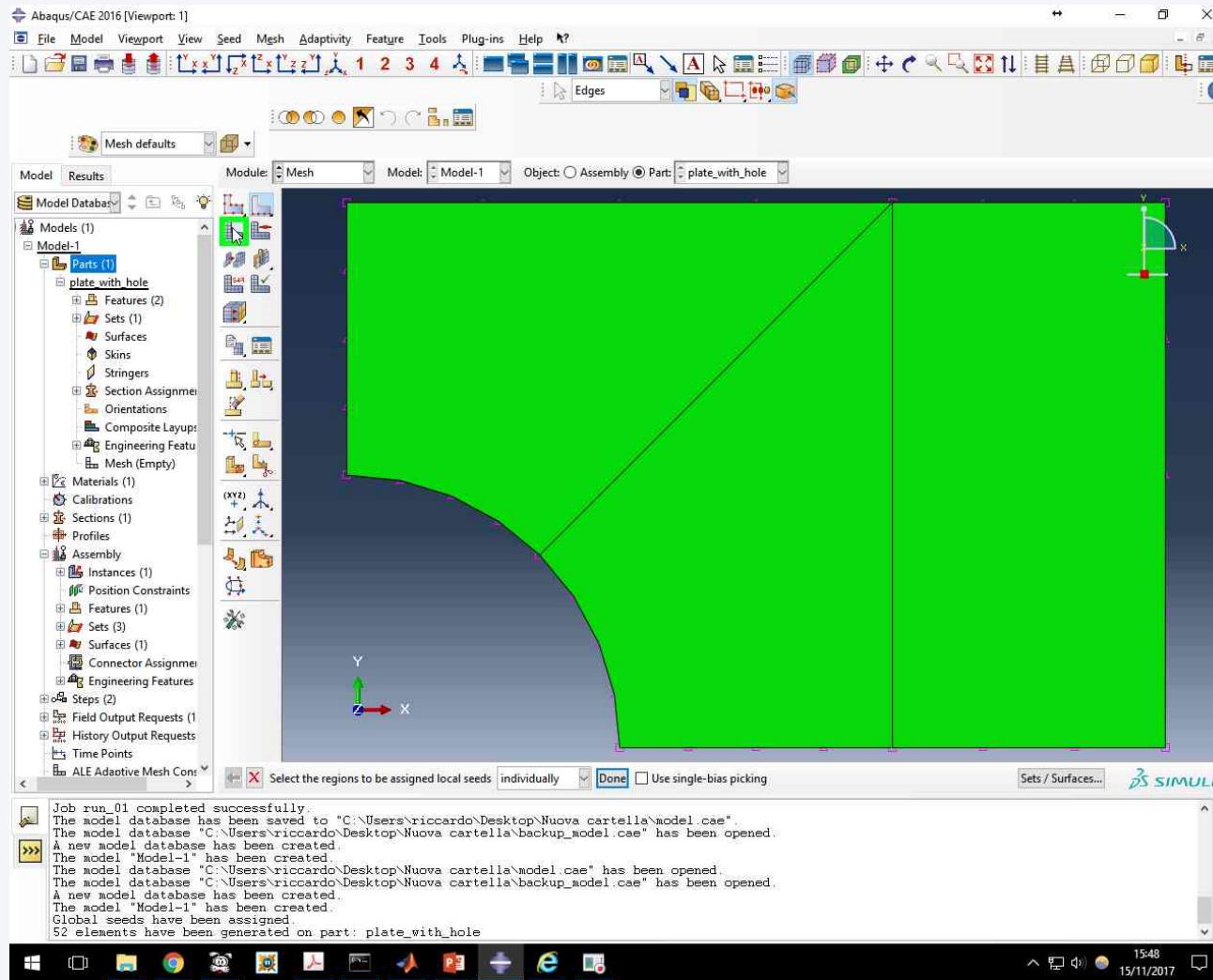
Select 3 elements for the remaining two edges.

Remeshing using partitions



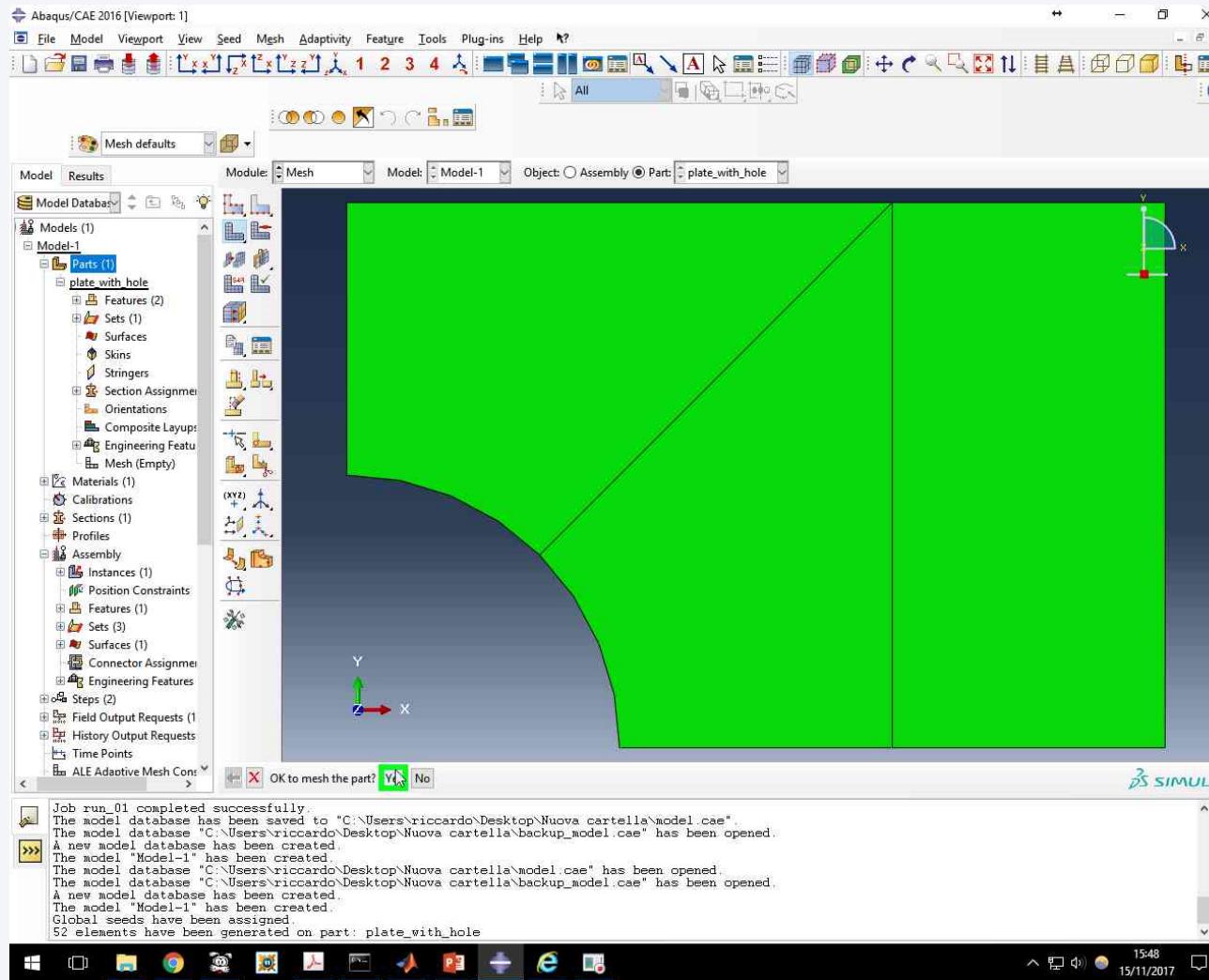
Select 3 elements for the remaining two edges.

Remeshing using partitions



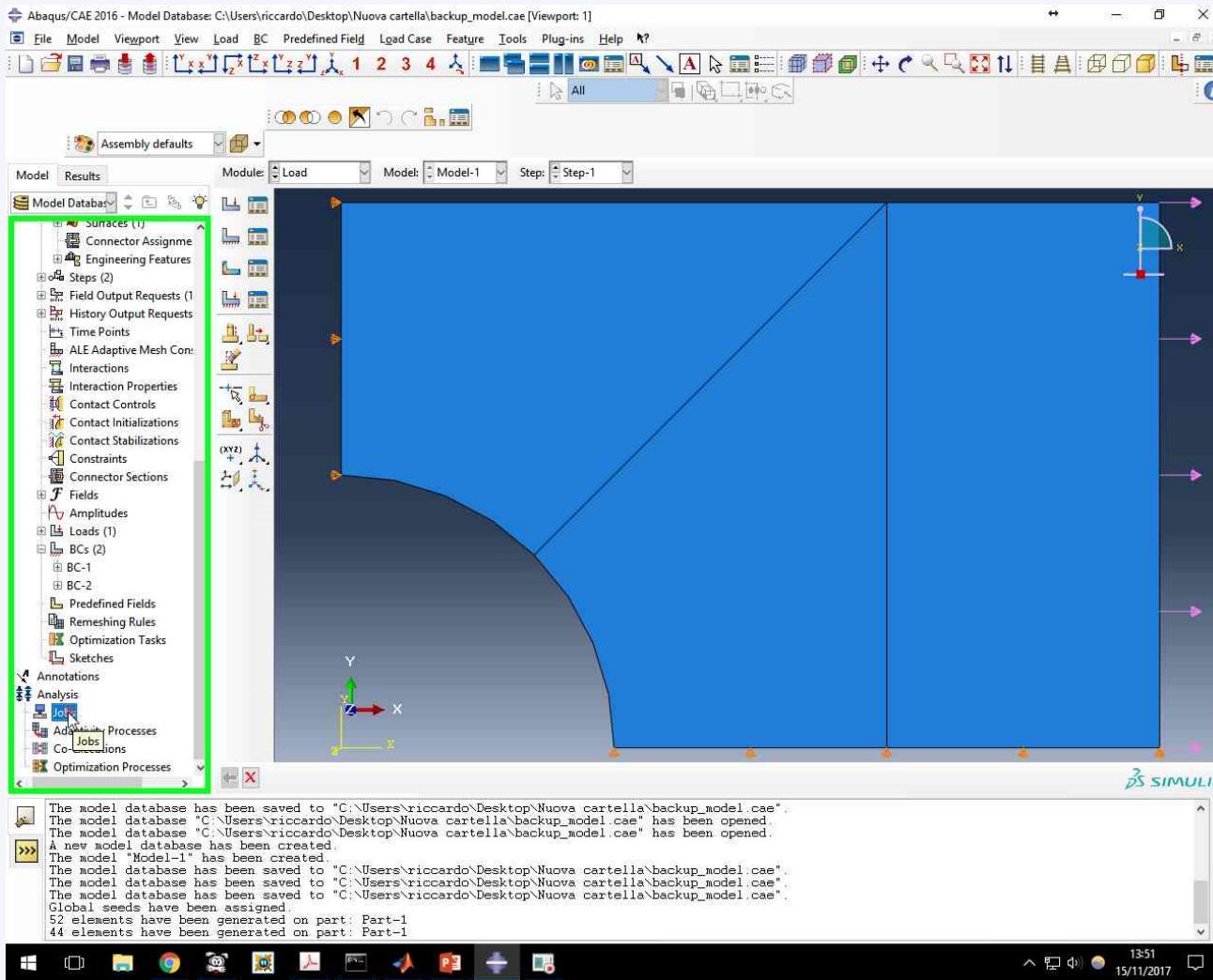
Mesh again the part.

Remeshing using partitions



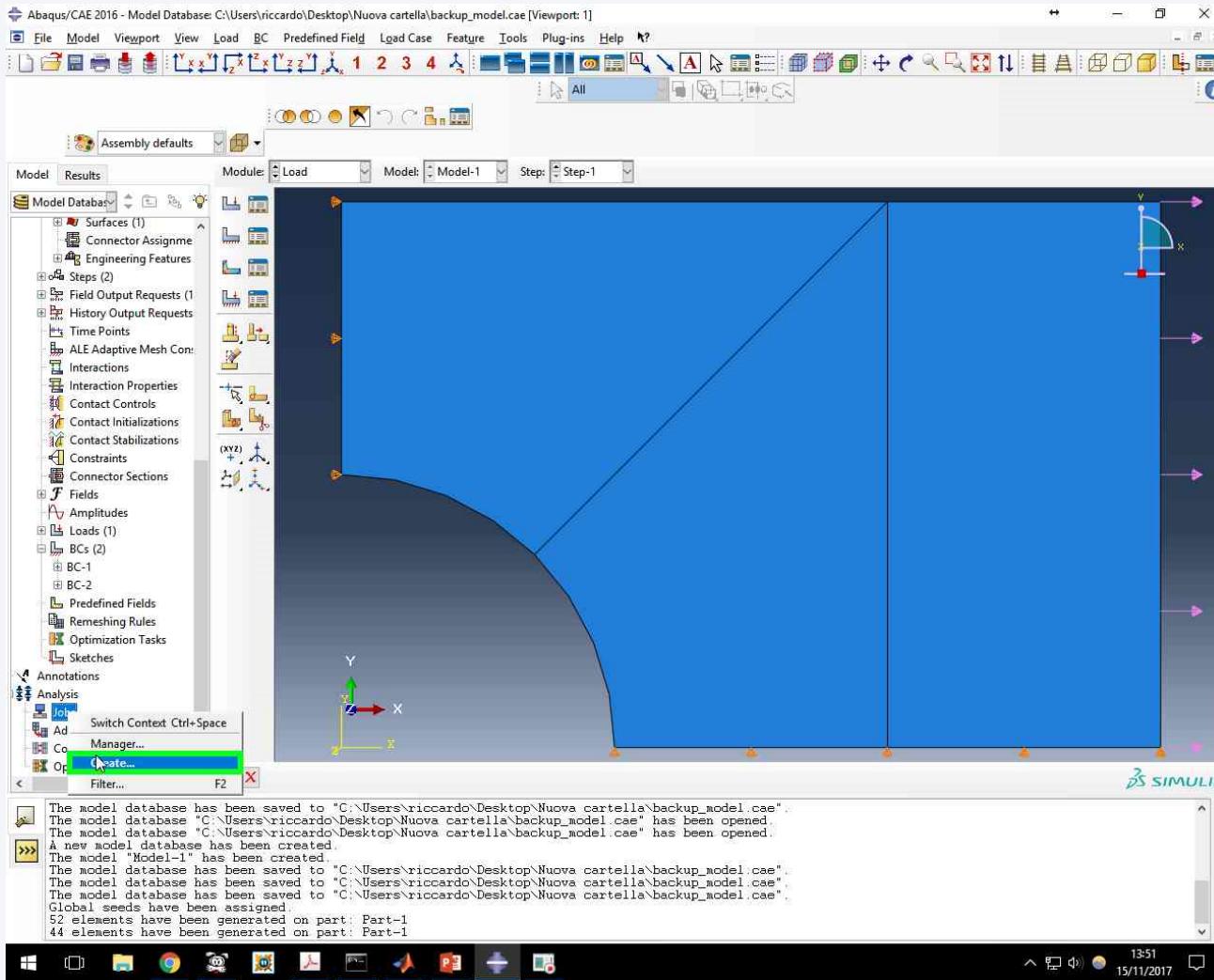
Confirm.

Create job



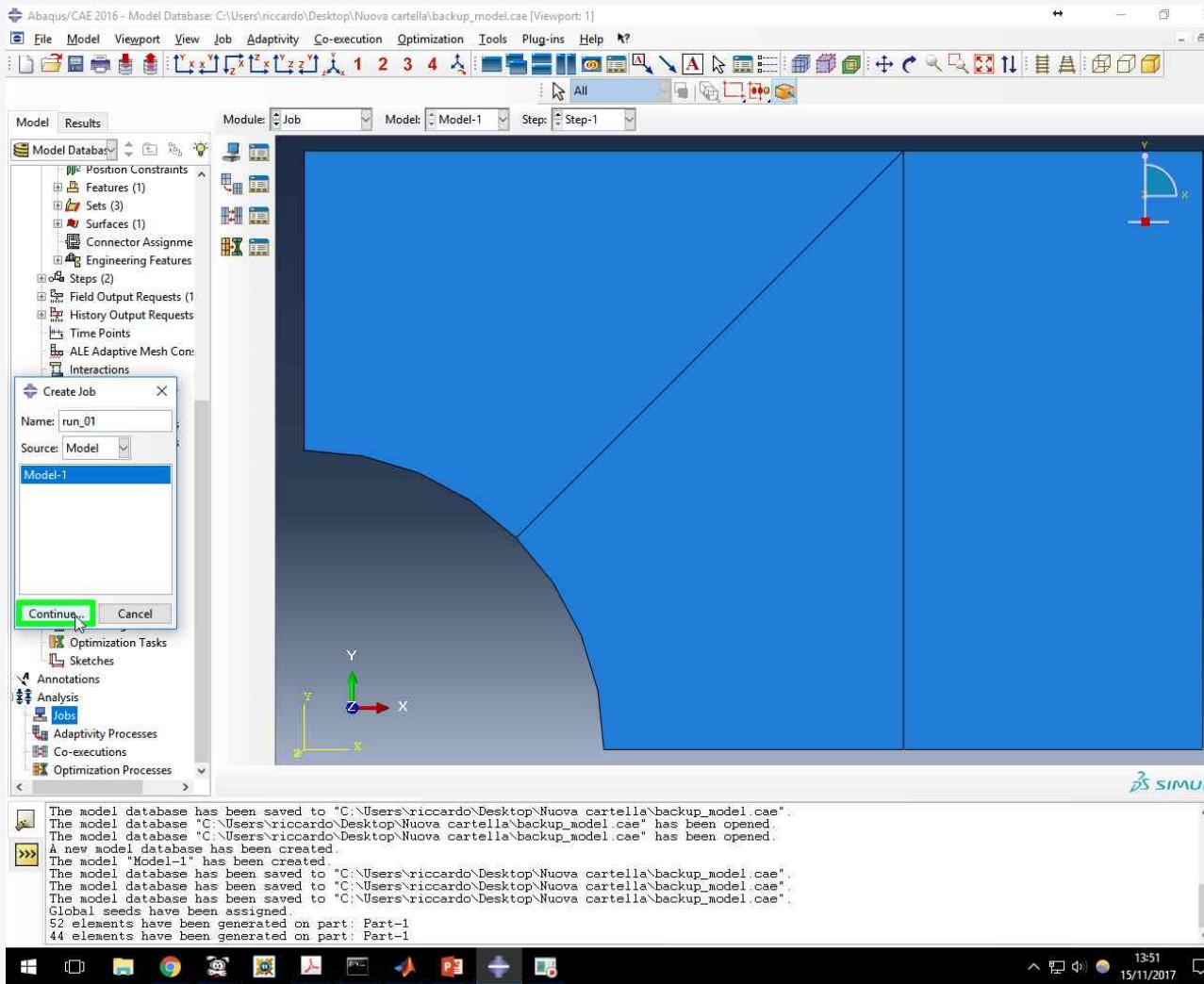
Creating the job is the last step before launching the analysis. This operation consists in assigning a name to the analysis, writing the input file and submitting the analysis.

Create job



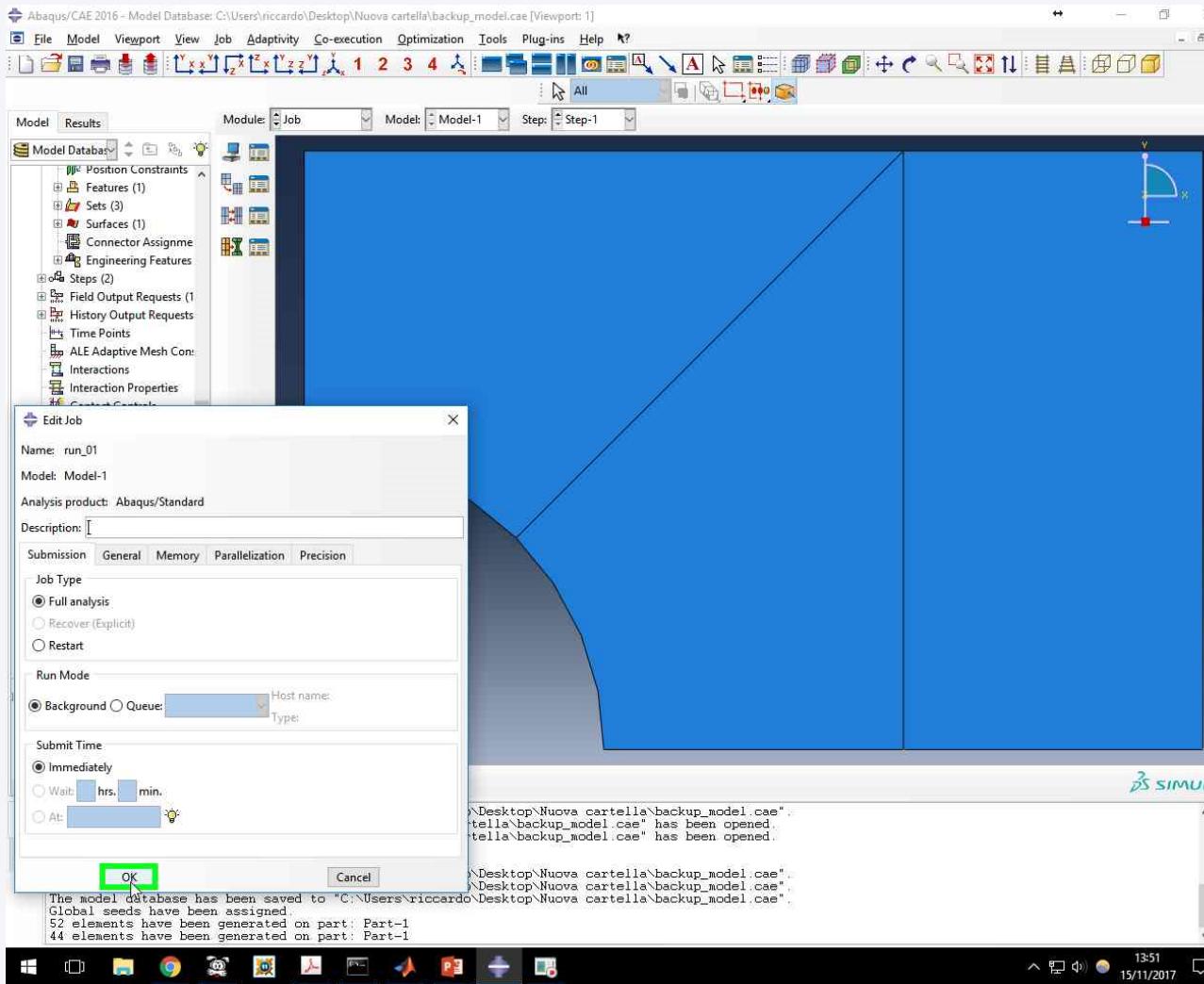
Right-click on job and select "Create".

Create job



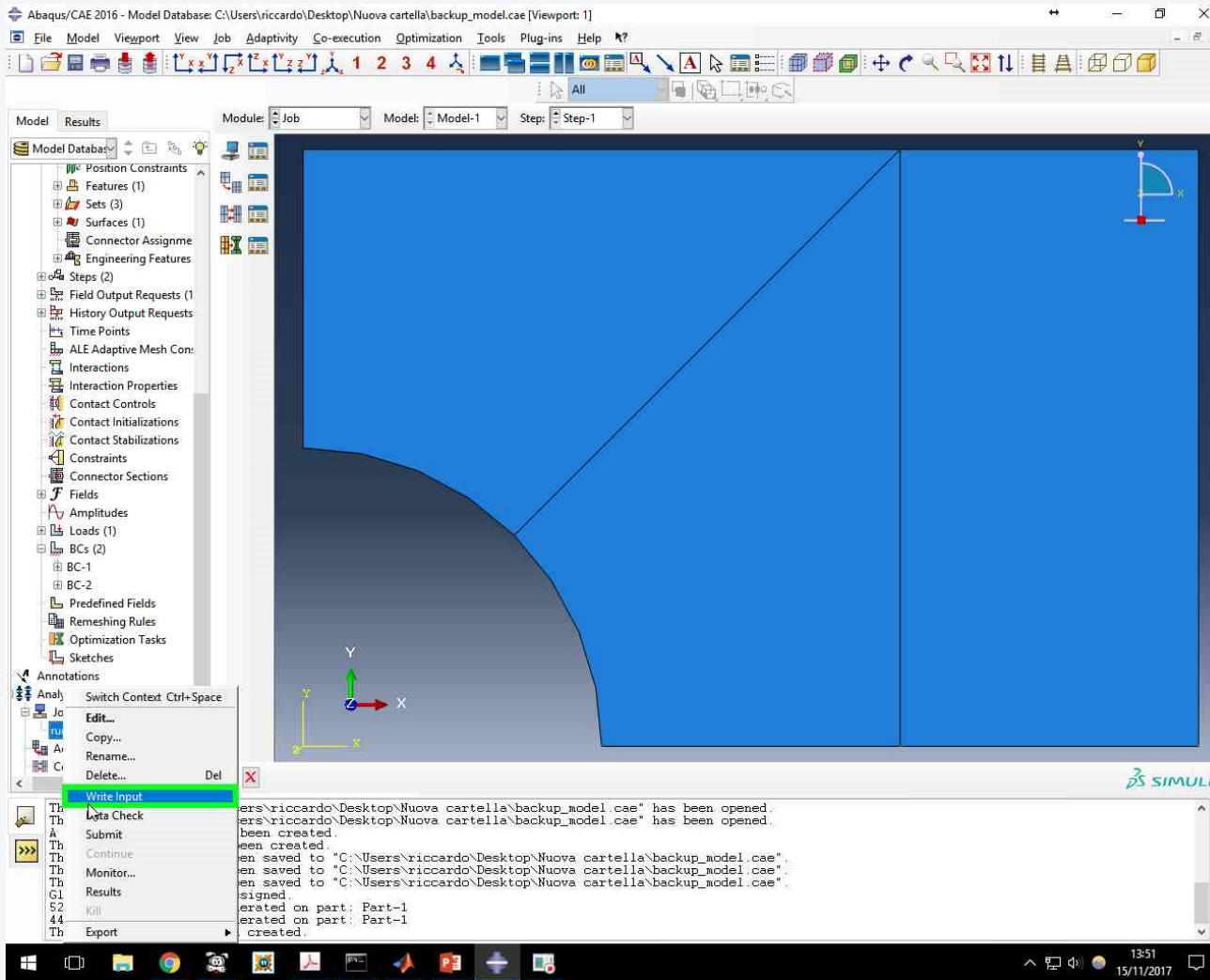
Confirm.

Create job



Confirm.

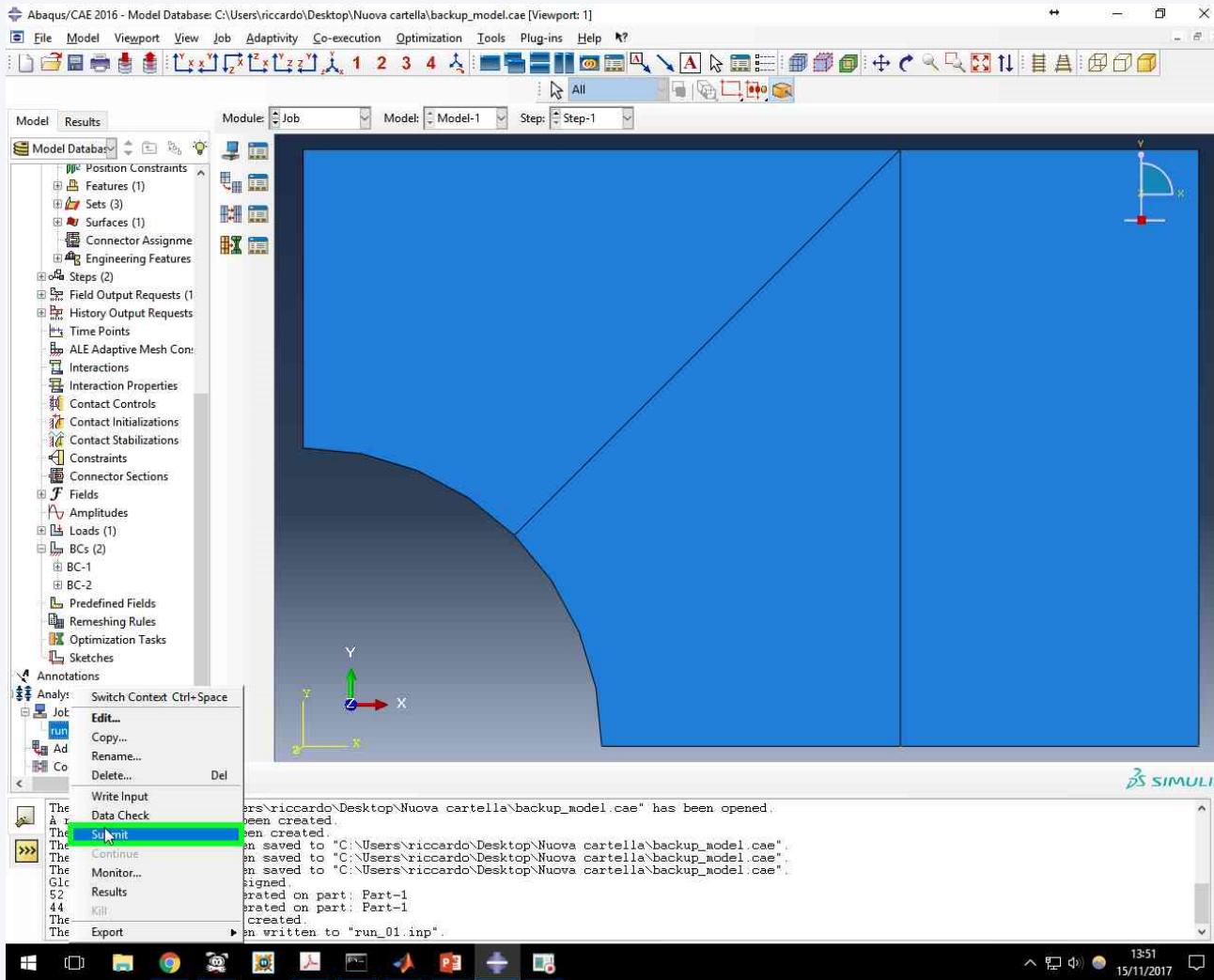
Create job



Although not strictly necessary, write the input file. The file is written with .inp extension into the folder where Abaqus cae was launched. The name of the file is the name selected for the job.

The .inp file can be edited with any text editor.

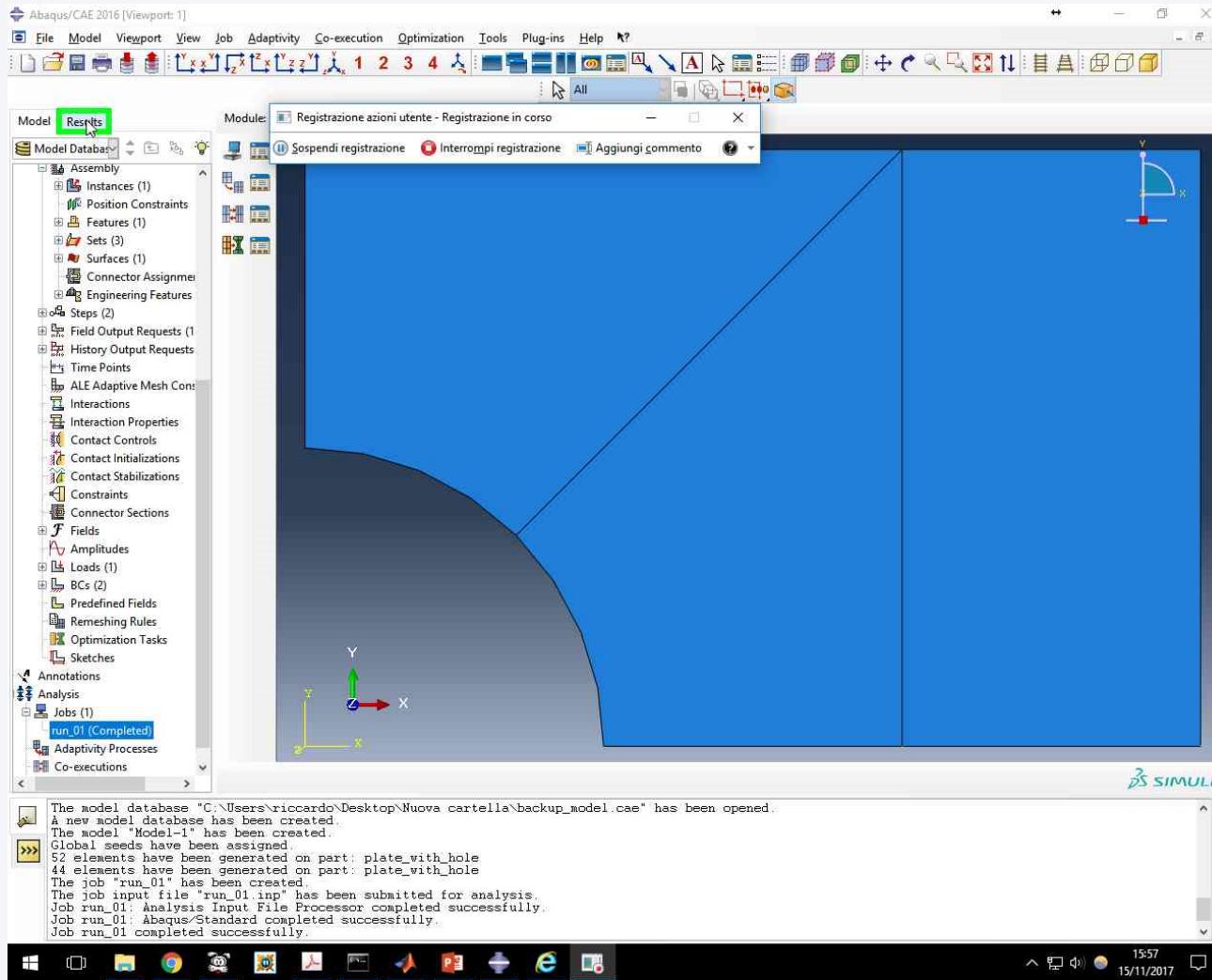
Create job



Click on "Submit" for running the analysis. This operation is reported here for completeness but is not recommended. In general, a better practice consists in:

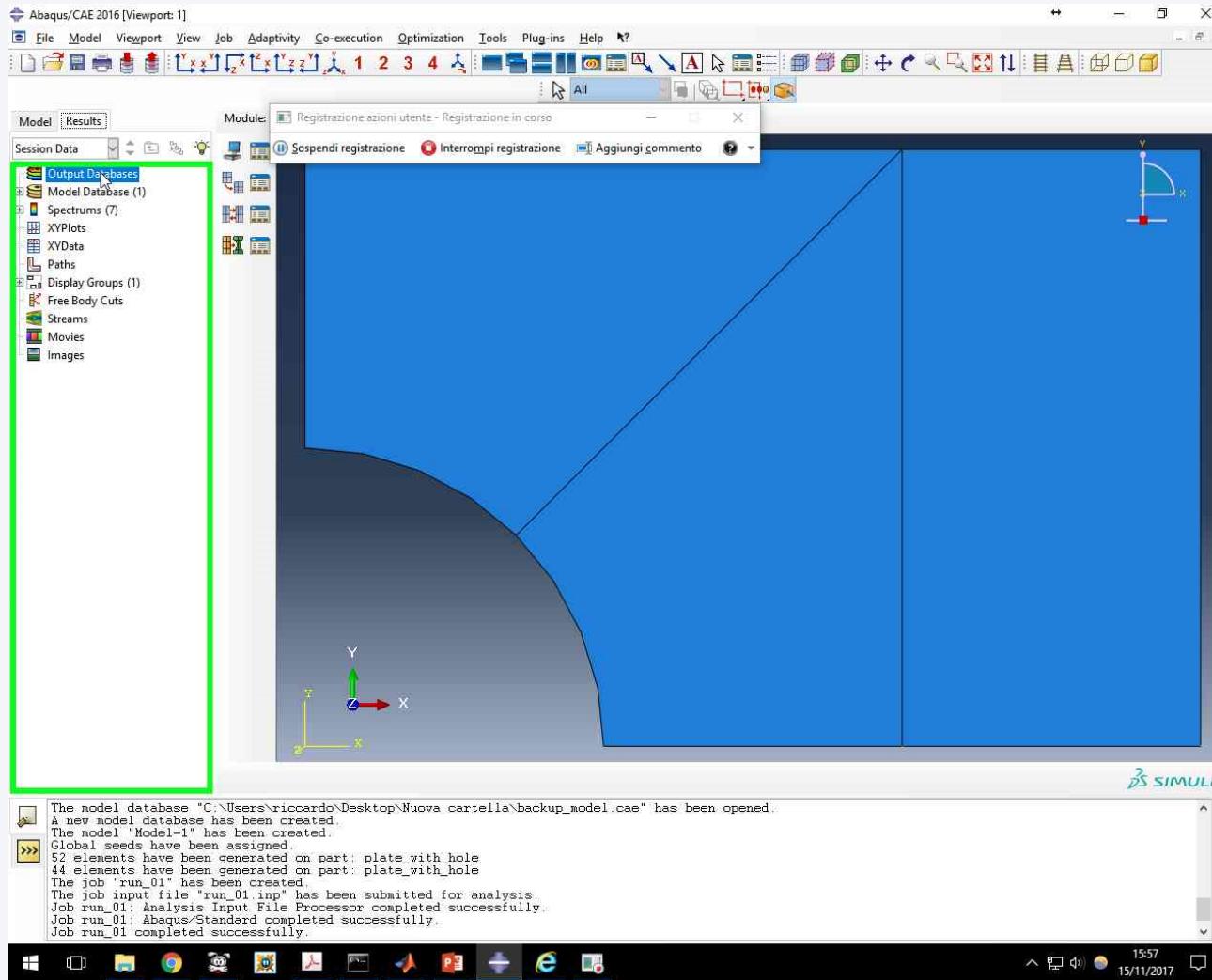
1. editing the .inp file to check the model
2. Running the analysis via command prompt

Accessing results



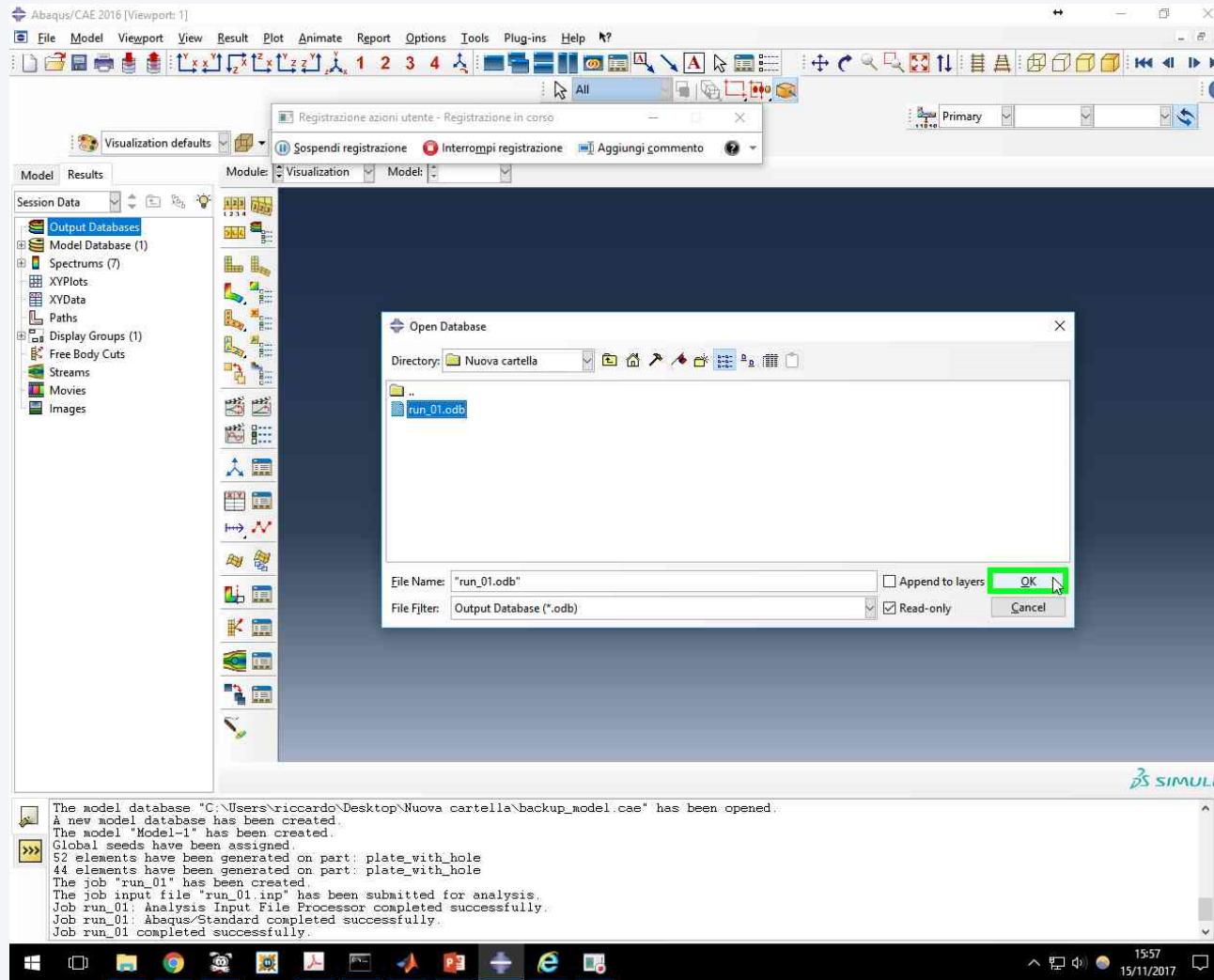
Once the submission status is "Completed" it is possible to open the result file. This file has extension .odb, and is a binary file.

Accessing results



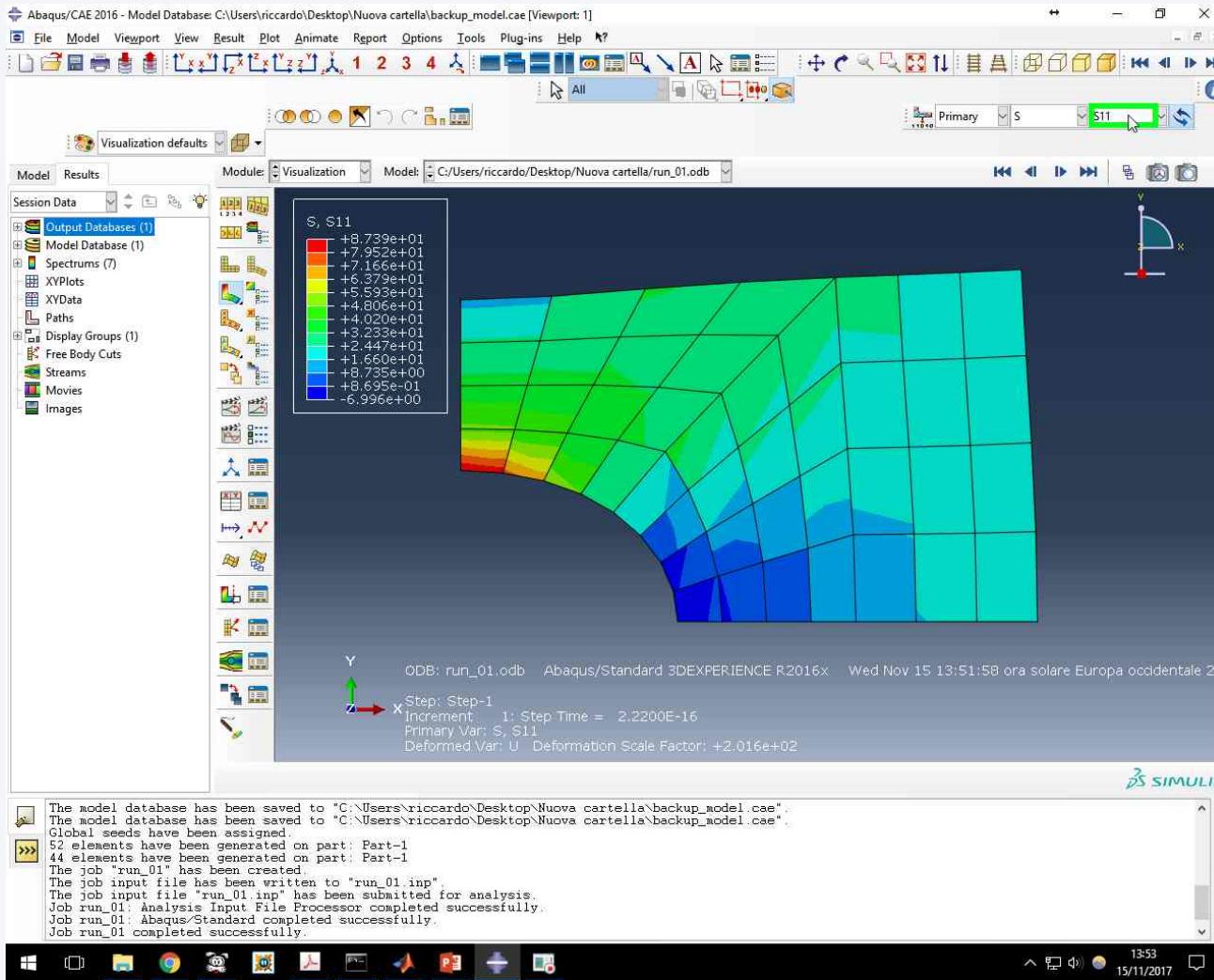
Once the submission status is "Completed" it is possible to open the result file. This file has extension .odb, and is a binary file.

Accessing results



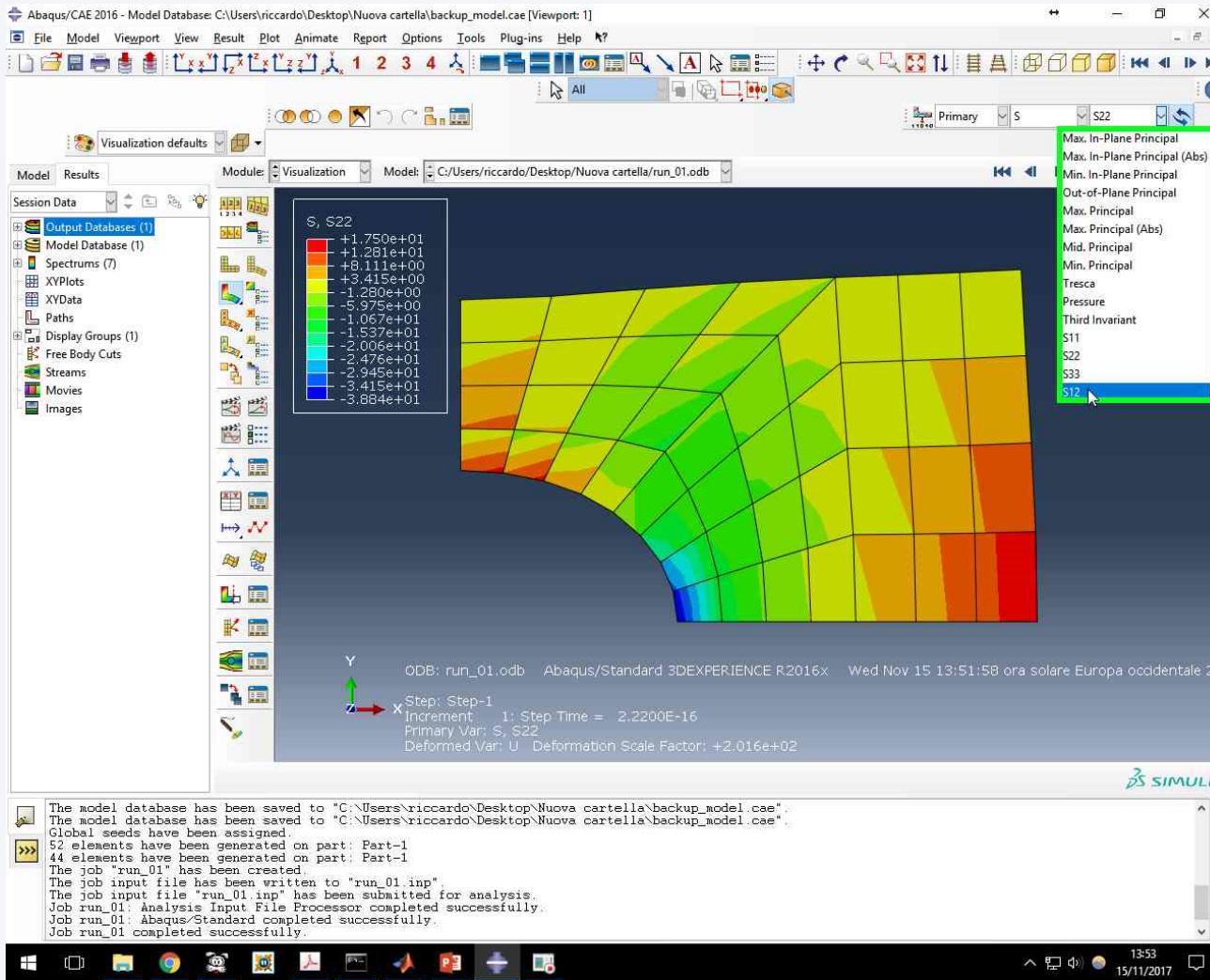
Open the result file.

Viewing results



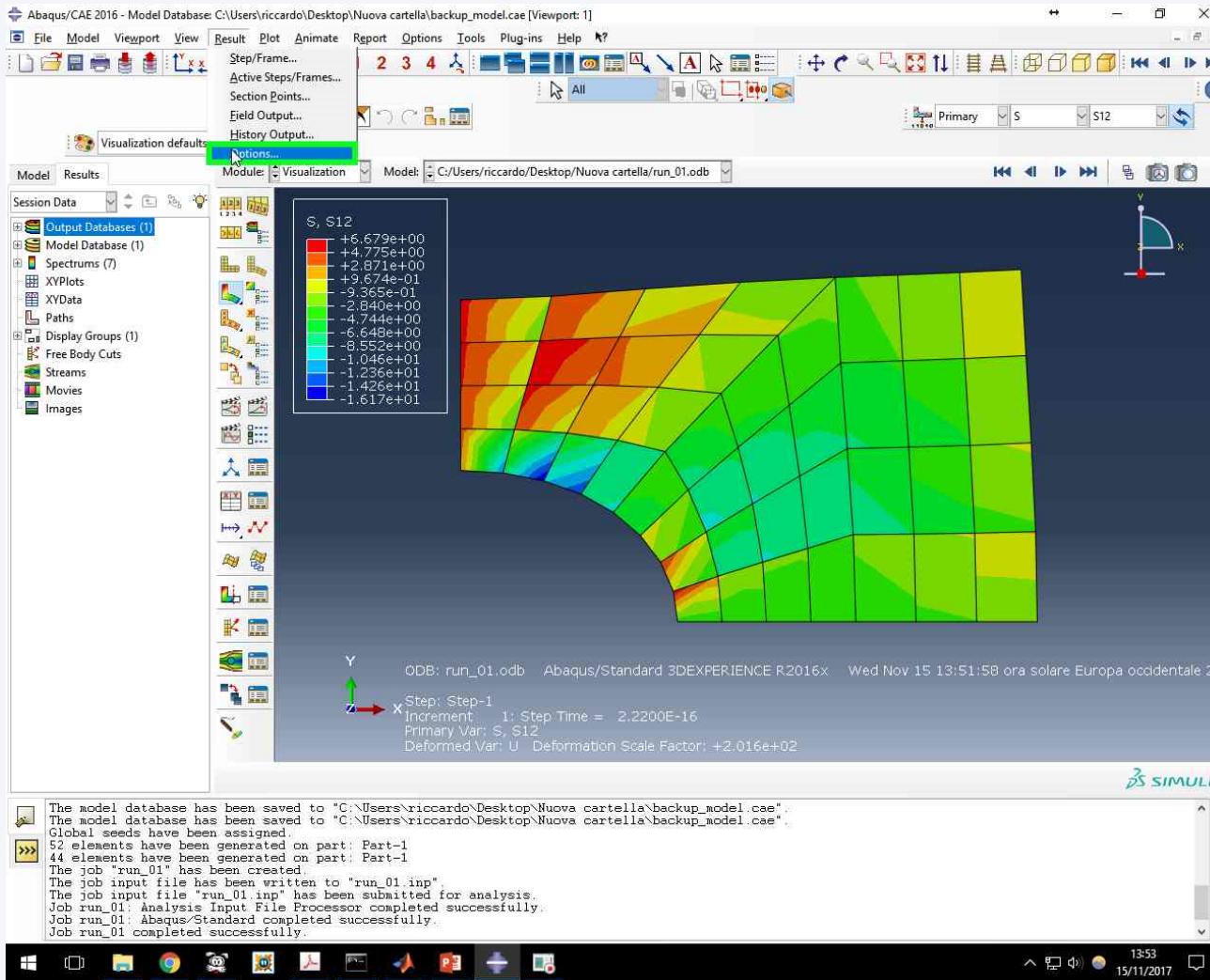
The results are reported in the form of contour plots. On the top right of the viewport it is possible to select the quantity to be used for the contour. In this example reported is the stress component s11 (=sxx).

Viewing results



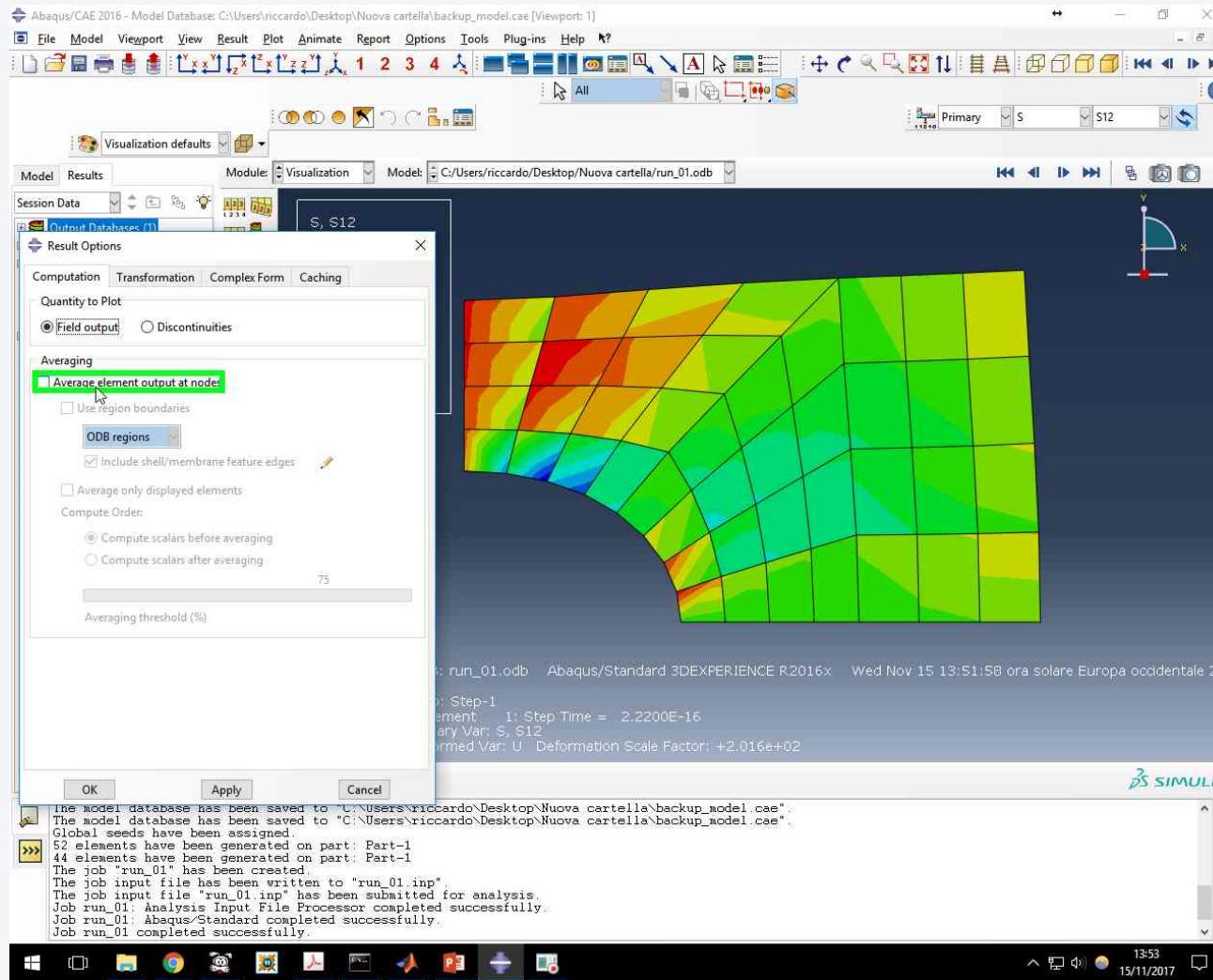
To visualize, for instance, the in-plane shear stresses, select s12.

Viewing results



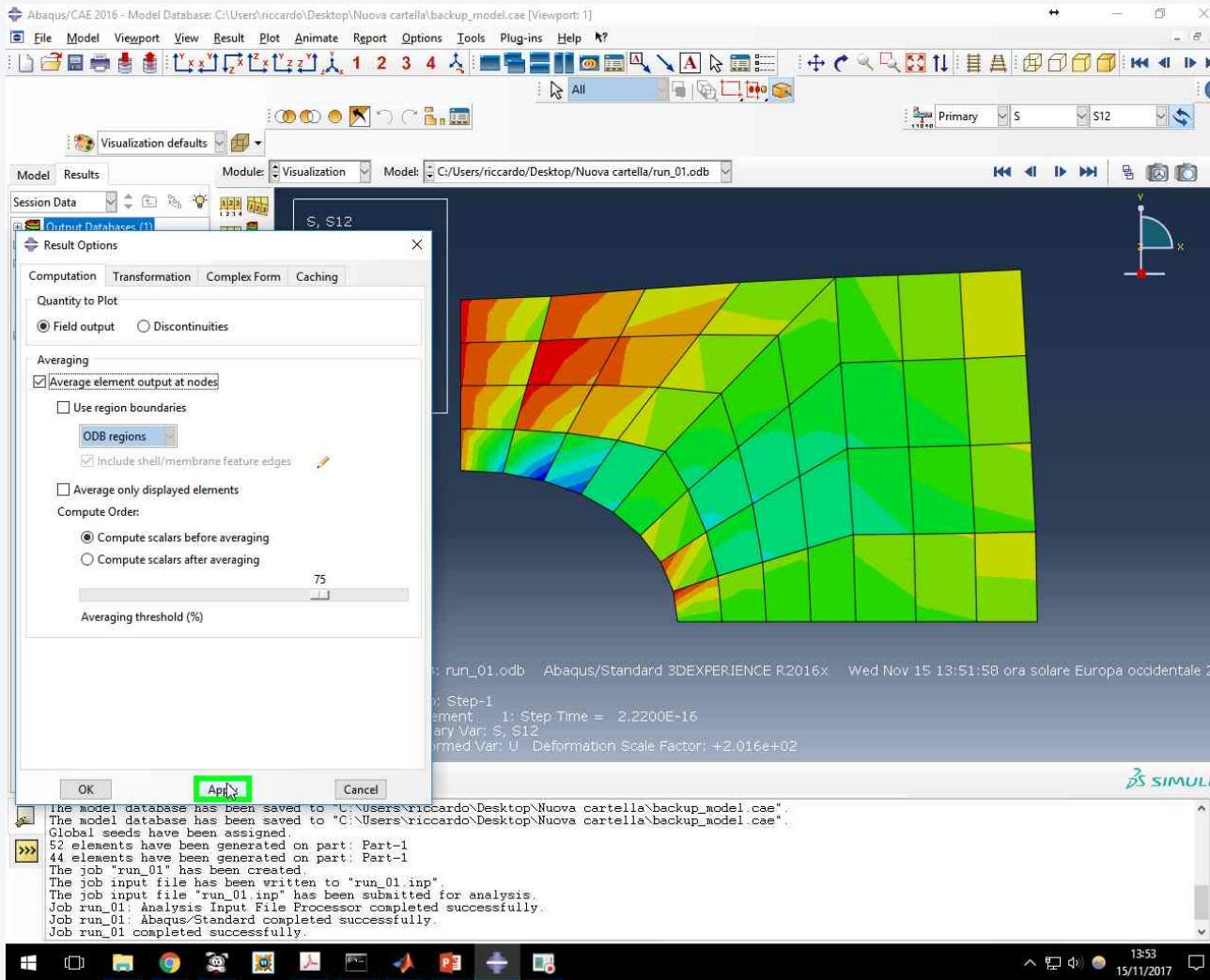
The contour can be realized with or without averaging the nodal values. In the current plot, the stresses are reported without averaging at common nodes of the elements.

Viewing results



The averaging option can be activated as shown.

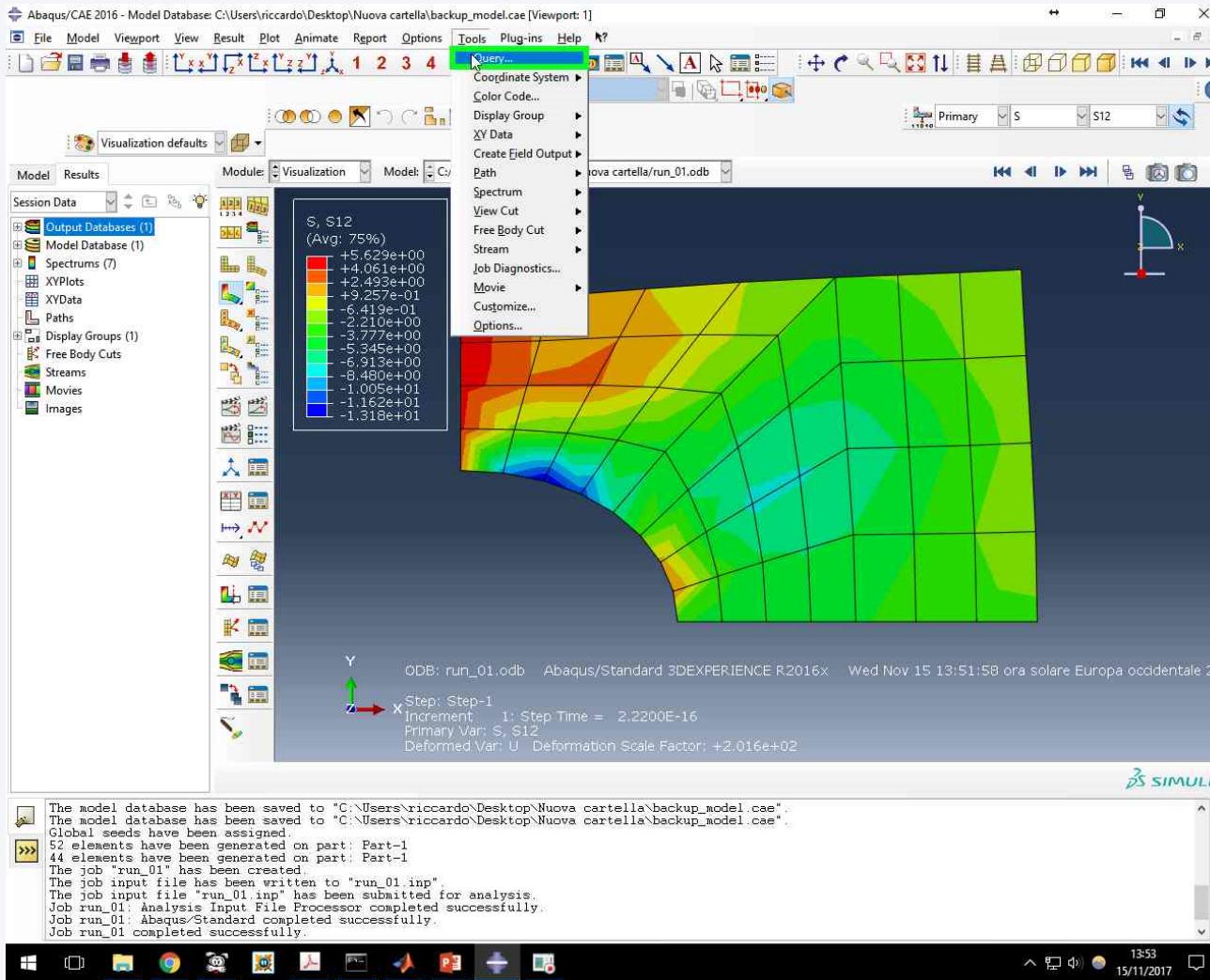
Viewing results



Confirm.

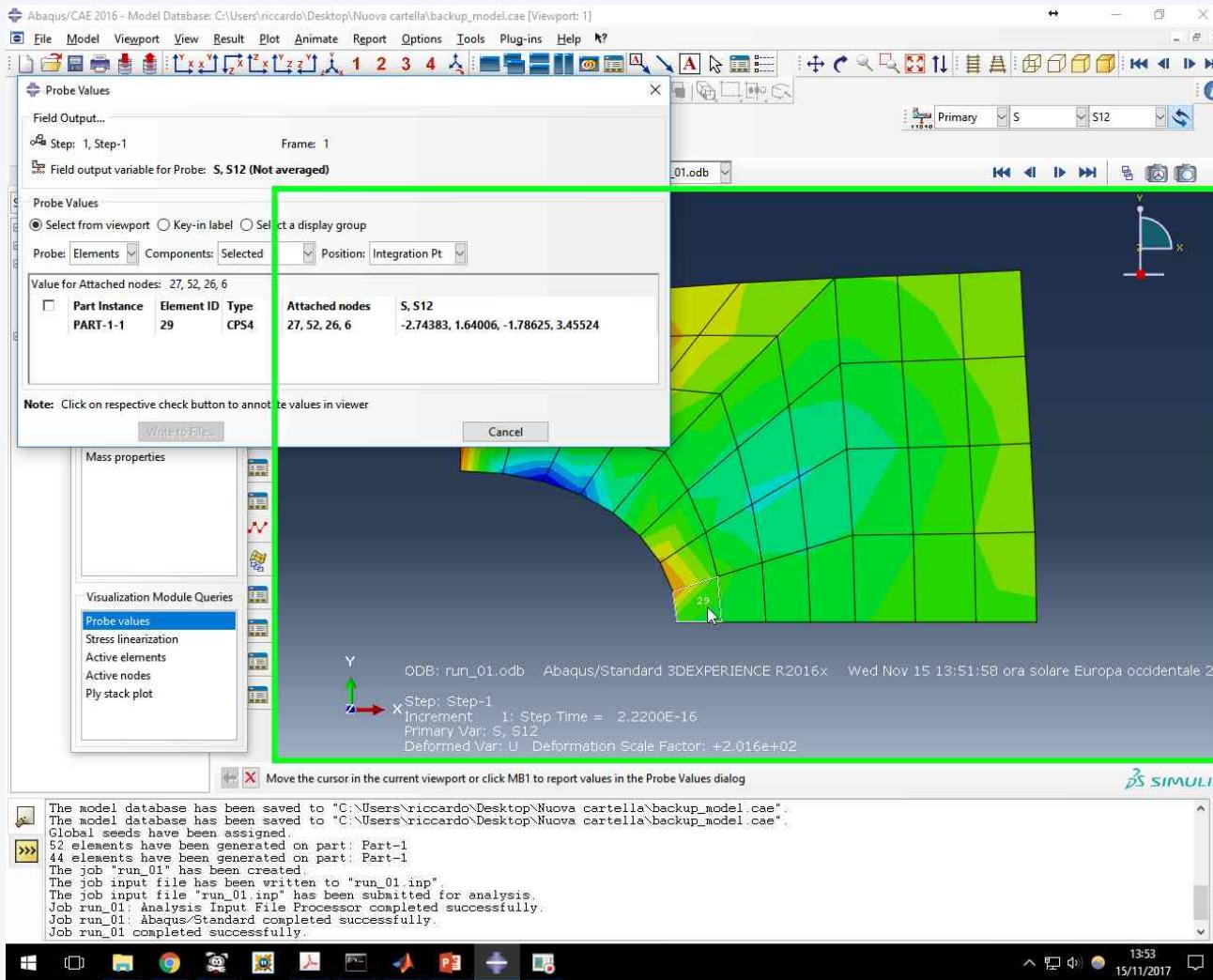
Due to the averaging process the plot is now more smooth. While the averaging process is helpful in providing nicer and more understandable plots, careful should be used to avoid not desired smoothing effects of stress discontinuities, as it happens in the presence of sudden variations of thickness, material properties, ...

Viewing results



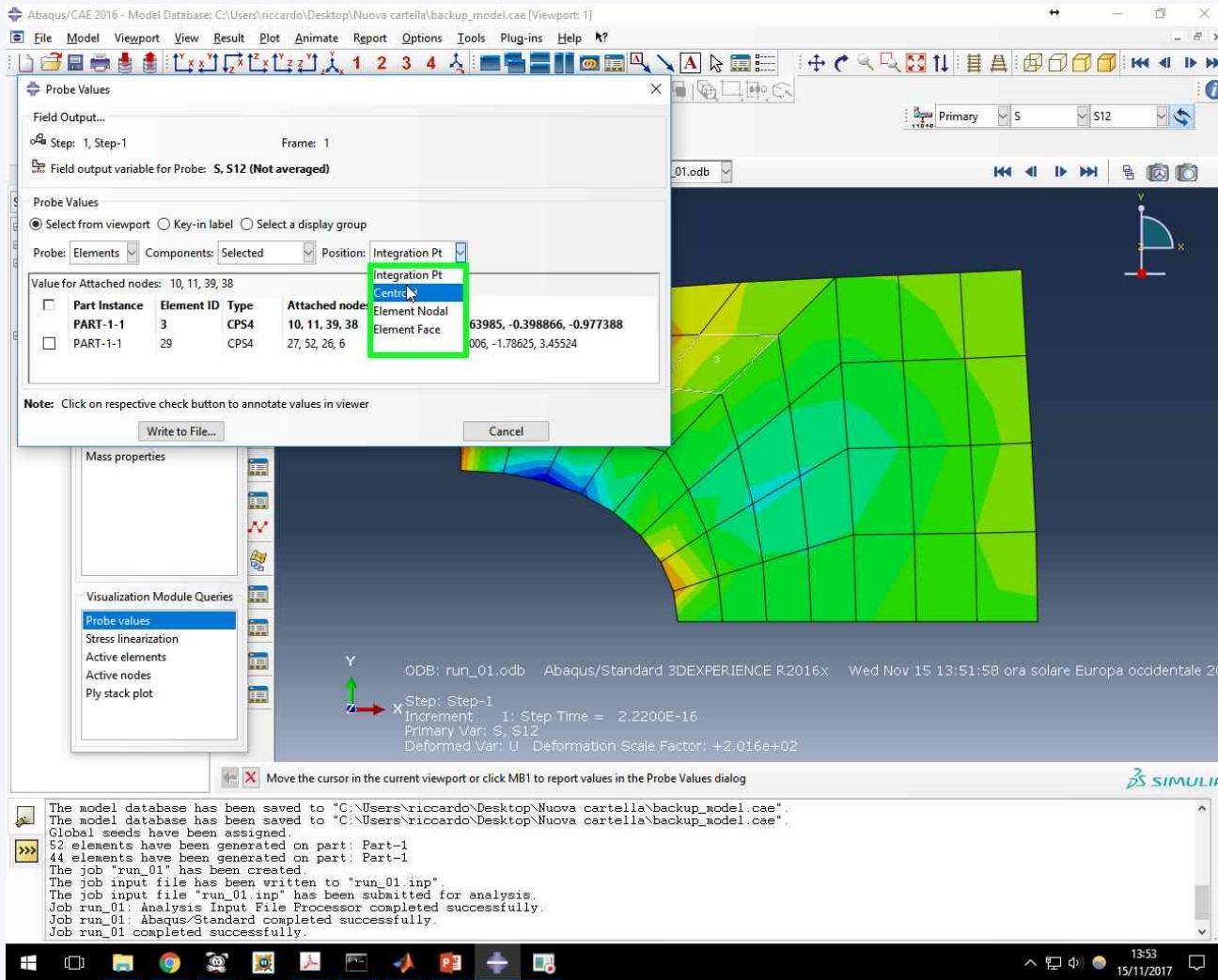
The local values can be required using the Tools → Query functionality.

Viewing results



By clicking on the elements, it is possible to visualize the stresses at the integration points...

Viewing results



... as well as their extrapolation to other positions. The centroid of the elements as well as the nodes.

A brief introduction to the use of Abaqus: the input file

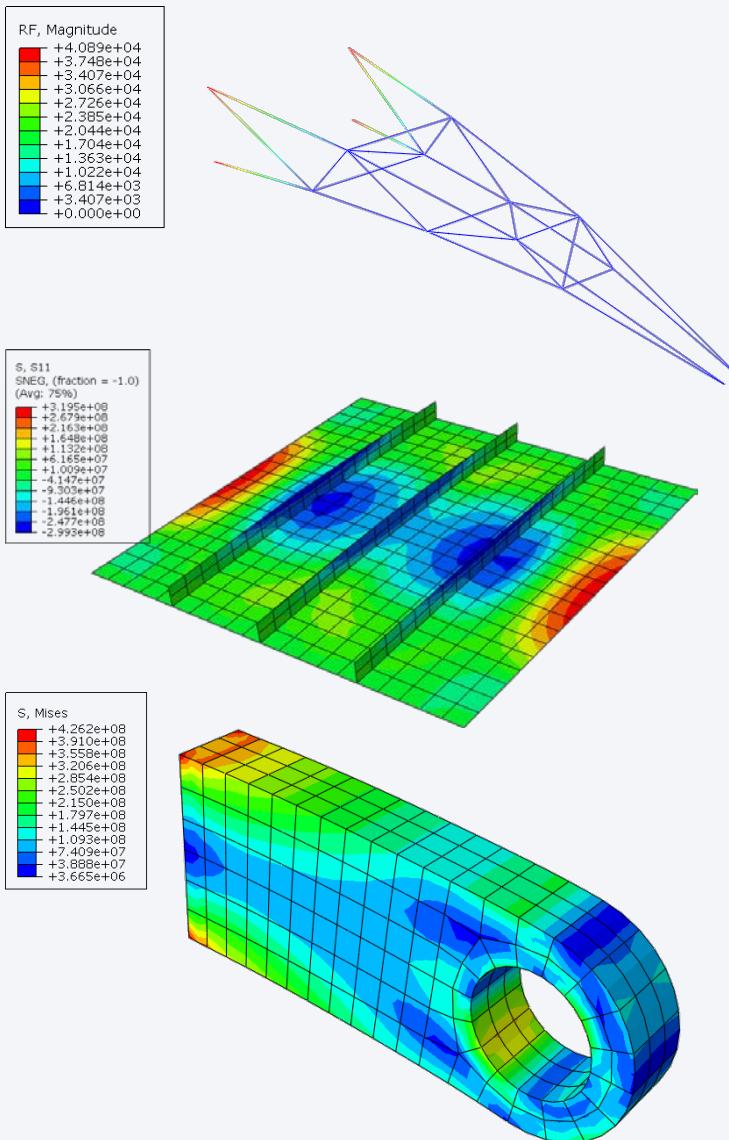
Course of Spacecraft Structures
AY 2017/2018

Riccardo Vescovini
Politecnico di Milano, Department of Aerospace Science and Technology

The ingredients of the finite element model

- Discretized geometry
 - finite elements and nodes
 - different elements available: beam, thin/thick shell, continuum shell, brick,...
- the collection of elements and nodes defines the mesh
- Element section properties
 - for many elements, the geometry is not completely defined by the coordinates of the nodes (e.g. the dimension of a beam section)
- Material data
 - isotropic, anisotropic,...

The ingredients of the finite element model



Example: model of a cargo crane using truss elements

Example: model of a stiffened plate using shell elements

Example: model of a lug using continuum elements

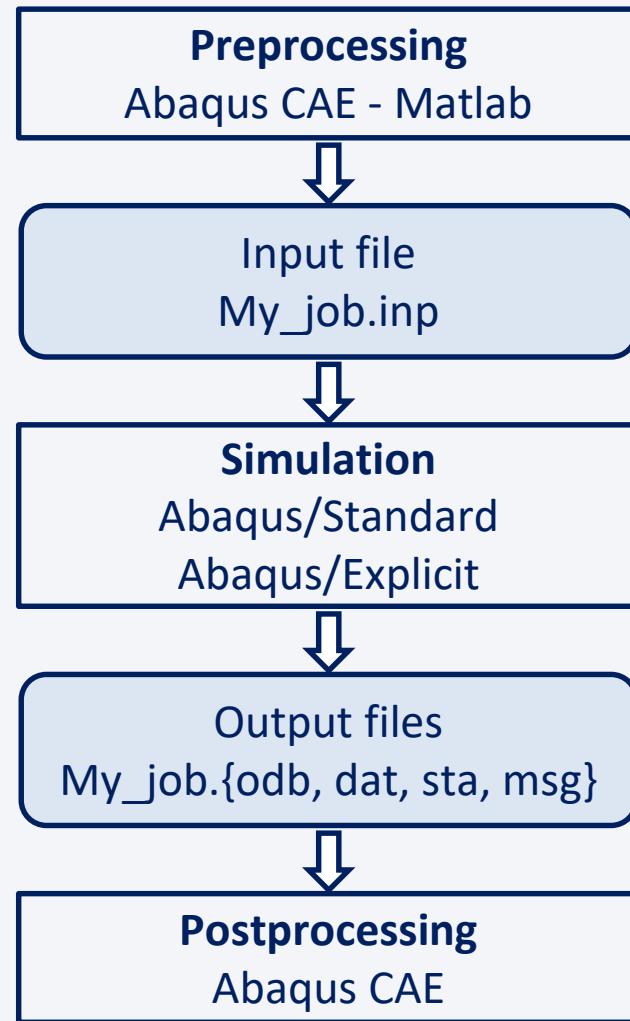
The ingredients of the finite element model

- Loads and boundary conditions
 - point loads, pressure loads, body forces, thermal loads
 - concentrated nodes applied in correspondence of a node
 - distributed loads should be reported to the nodes
 - BCs should prevent unrestrained rigid body motion (check error/warning message)
- Analysis type
 - different type of simulations available: static, eigenvalue, dynamic,...
 - is a linear approach appropriate?
 - are nonlinearities (geometric and/or material) present?
 - are dynamic effects important?

The ingredients of a finite element model

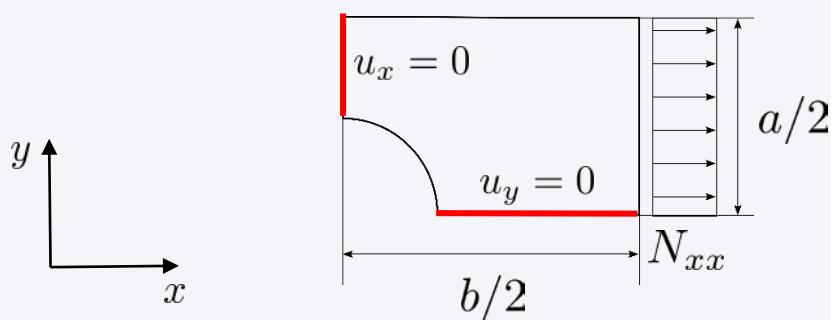
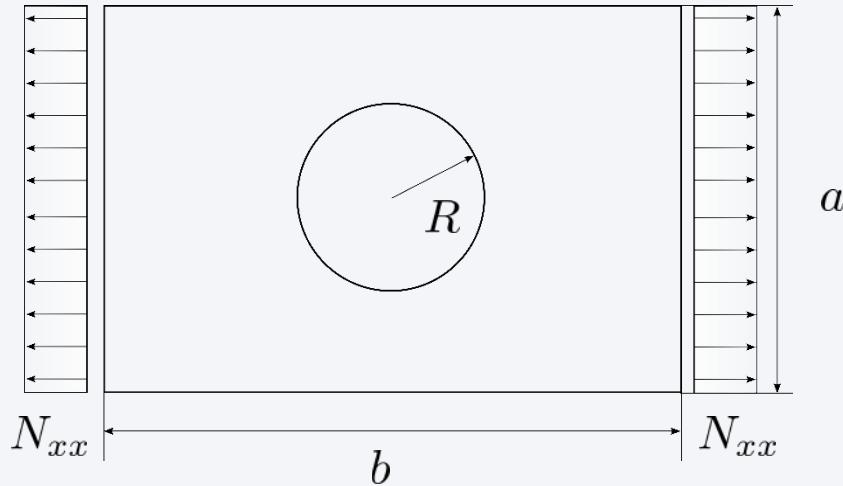
- Output
 - displacements, stresses, reaction forces,...
 - in a limited set of nodes/element or in the overall model
 - with different frequencies (for multi-step analysis)
 - in binary and/or ASCII files

Steps of the procedure



Example

- Open-hole test: membrane with center hole, loaded in traction (1/4 of the structure due to the double symmetry of the problem)



Input data

$$a = 100 \text{ mm}$$

$$b = 150 \text{ mm}$$

$$R = 25 \text{ mm}$$

$$t = 1 \text{ mm}$$

$$E = 72 \text{ GPa}$$

$$\nu = 0.3$$

$$N_{xx} = 20 \text{ N/mm}$$

The input file (generated by the CAE)

- Consider now the input file generated with the graphical interface after selecting “Write input” in the job creation phase. The content of the .inp file represents the one and only set of data which is used by the Abaqus for solving the problem
- With this regard, the input file can be created with any kind of pre-processor (not necessarily the Abaqus one), including the generation of the .inp file via script (for instance, in Matlab)

The input file (generated by the CAE)

- The content of the input file generated during the pre-processing phase can be examined by opening the .inp file with a text editor
- In Abaqus the symbol * denotes a keyword (Abaqus command); ** is a comment

```
*Heading  
** Job name: run_01 Model name: Model-1  
** Generated by: Abaqus/CAE 2016  
*Preprint, echo=NO, model=NO, history=NO,  
contact=NO  
**  
** PARTS  
**  
*Part, name=plate_with_hole  
*Node  
    1,    17.6776676,    17.6776714  
    2,    49.9999924,      50.  
    3,          0.,      50.  
    4,          0.,      25.  
    5,    49.9999924,      0.  
    6,          25.,      0.  
    7,          75.,      0.
```

Keyword *Part (alle the keyword below are referred to this part). When the file is written by hand, this keyword is not necessary

Keyword *node (all the entries here below are quantities referred to the node definition)

The syntax, in this case, is intuitive: Id of the node, x-position, y-position. (the z-position is not reported for a planar analysis, but in general wold be the fourth entry)

The input file (generated by the CAE)

- The content of the input file generated during the pre-processing phase can be examined by opening the .inp file with a text editor
- In Abaqus the symbol * denotes a keyword (Abaqus command); ** is a comment

```
*Heading
** Job name: run_01 Model name: Model-1
** Generated by: Abaqus/CAE 2016
*Preprint, echo=NO, model=NO, history=NO,
contact=NO
**
** PARTS
**
*Part, name=plate_with_hole
*Node
    1,    17.6776676,    17.6776714
    2,    49.9999924,      50.
    3,          0.,      50.
    4,          0.,      25.
    5,    49.9999924,      0.
    6,          25.,      0.
    .
    .
    .
```

note that the ID is an integer. A real number would cause an error!

The input file (generated by the CAE)

```
*Element, type=CPS4
1, 1, 9, 37, 20
2, 9, 10, 38, 37
3, 10, 11, 39, 38
4, 11, 2, 12, 39
5, 20, 37, 40, 19
6, 37, 38, 41, 40
7, 38, 39, 42, 41
8, 39, 12, 13, 42
9, 19, 40, 43, 18
.
.
.
```

Keyword *element; defines the type of element with the parameter «type» (in this case CPS4), and the nodes composing the elements

Element 8, composed by nodes 39, 12, 13 and 42

The input file (generated by the CAE)

```
*Nset, nset=Set-1, generate  
 1, 60, 1  
*Elset, elset=Set-1, generate  
 1, 44, 1  
** Section: Section-1  
*Solid Section, elset=Set-1,  
material=Material-1  
1.,  
*End Part
```

keyword *Nset: create a set of nodes whose name is Set-1

keyword *Elset: create a set of elements whose name is Set-1

keyword *Solid Section: specifies the section properties. In this case this is done by specifying the name of the material in the parameter material=, and the thickness of the membrane in the first line

*end of the part

The input file (generated by the CAE)

```
*Assembly, name=Assembly          → Define the assembly (=collection of
**                                         parts)
*Instance, name=plate_with_hole-1,   → One and only part composing the assembly
part=plate_with_hole
*End Instance

**
*Nset, nset=Set-1, instance=plate_with_hole-1 → Specify the set of nodes Set-1
  3, 4, 15, 16, 17
*Nset, nset=Set-2, instance=plate_with_hole-1 → Specify the set of nodes Set-2
  5, 6, 7, 27, 28, 29, 30, 31
*Elset, elset=_Surf-1_S3, internal,
instance=plate_with_hole-1, generate      → Specify the information (set
  41, 44, 1                                of elements and surface) for
                                              applying the distributed load.
*Surface, type=ELEMENT, name=Surf-1
  _Surf-1_S3, S3
*End Assembly                                → Terminate assembly definition
```

The input file (generated by the CAE)

```
*Assembly, name=Assembly
**
*Instance, name=plate_with_hole-1,
part=plate_with_hole
*End Instance

**
*Nset, nset=Set-1, instance=plate_with_hole-1
  3, 4, 15, 16, 17
*Nset, nset=Set-2, instance=plate_with_hole-1
  5, 6, 7, 27, 28, 29, 30, 31
*Elset, elset=_Surf-1_S3, internal,
instance=plate_with_hole-1, generate
  41, 44, 1
*Surface, type=ELEMENT, name=Surf-1
_Surf-1_S3, S3
*End Assembly
```

Note that in this case, and for any model which is not composed by several parts, no practical advantages exist in creating a model as an assembly of part instances.

When preparing the input file by script, these commands could be removed. They are generated by the Abaqus CAE, as this is the way Abaqus organizes the model.

Interlude: assemblies and parts

Assembly

An assembly is a collection of positioned part instances. An analysis is conducted by defining boundary conditions, constraints, interactions, and a loading history for the assembly.

Part

A part is a finite element idealization of an object. Parts are the building blocks of an assembly and can be either rigid or deformable. Parts are reusable; they can be instanced multiple times in the assembly. Parts are not analyzed directly; a part is like a blueprint for its instances.

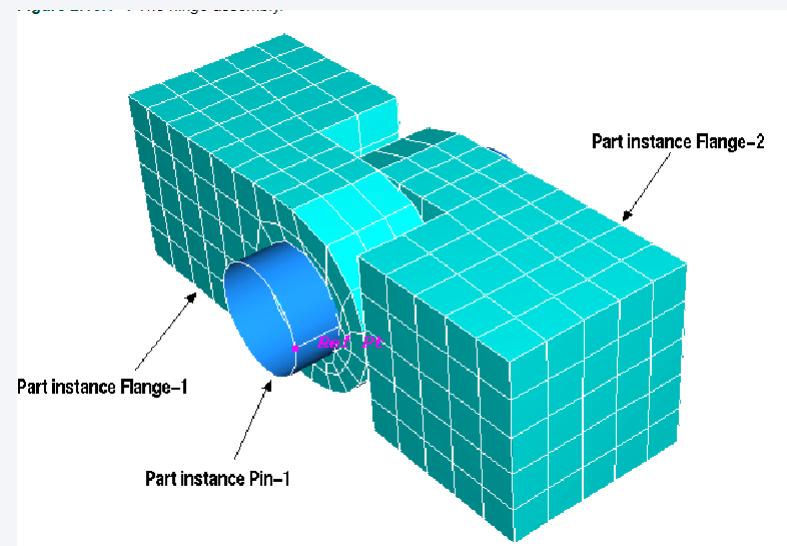
Part instance

A part instance is a usage of a part within the assembly. All characteristics (such as mesh and section definitions) defined for a part become characteristics for each instance of that part—they are inherited by the part instances. Each part instance is positioned independently within the assembly.

Interlude: assemblies and parts

- In this case the advantages of building the assembly as a collection of parts is clear. The Abaqus code would be:

```
*ASSEMBLY, NAME=Hinge
  *INSTANCE, NAME=Flange-1, PART=Flange
    <positioning data>
  *END INSTANCE
  *INSTANCE, NAME=Flange-2, PART=Flange
    <positioning data>
  *END INSTANCE
  *INSTANCE, NAME=Pin-1, PART=Pin
    <positioning data>
  *END INSTANCE
  *ELSET, ELSET=Top
  ...
  *NSET, NSET=Output
  ...
*END ASSEMBLY
```



The mesh of the flange is used twice

The input file (generated by the CAE)

```
**  
** MATERIALS  
**  
*Material, name=Material-1 → Define a linear elastic material  
*Elastic  
72000., 0.3  
** -----  
-----  
**  
** STEP: Step-1  
**  
*Step, name=Step-1, nlgeom=NO, perturbation → Specify the solution procedure  
*Static  
**  
** BOUNDARY CONDITIONS  
**  
** Name: BC-1 Type: Displacement/Rotation  
*Boundary → Set the boundary conditions:  
Set-1, 1, 1  
** Name: BC-2 Type: Displacement/Rotation  
*Boundary  
Set-2, 2, 2  
the node Set-1 is constrained from 1 (x-direction) to 1 (x-direction)
```

The input file (generated by the CAE)

```
**  
** LOADS  
**  
** Name: Load-1    Type: Surface traction  
*Dsload, follower=NO, constant resultant=YES → Specify the distributed load (see manual  
Surf-1, TRVEC, 20., 1., 0., 0.  
**  
** OUTPUT REQUESTS  
**  
**  
** FIELD OUTPUT: F-Output-1  
**  
*Output, field, variable=PRESELECT → Output request (post-processing).  
**  
** HISTORY OUTPUT: H-Output-1  
**  
*Output, history, variable=PRESELECT  
*End Step  
The option preselect is intended to  
require a set of default output  
(displacement, stresses,...)
```

The input file: a simple template (1/2)

- In many cases, unless the model is relatively complex, the input file created by Abaqus can be simplified by removing parts, instances and templates

```
*node, nset = my_nodes
1, 17.6776676, 17.6776714

*element, type = CPS4, elset = my_els
1, 1, 9, 37, 20

*material, name = my_material ←
*elastic
72000., 0.3

*solid section, elset = Set-1, material = my_material ←
1.,

*nset, nset = edge_1 ←
3, 4, 15, 16, 17

*step
*static

*boundary
edge_1, 1, 1 ←

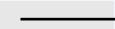
*cload
```

■ : arbitrary names

The input file: a simple template (2/2)

- The simpler way for requesting the output is:

```
*output, field, variable = preselect  
*end step
```



The parameter `field` specifies that the output has to be written on the binary `.odb` file

- In many cases, it is necessary to specify the kind of output, and the where the output has to be written. Thus, the following commands can be used

```
*output, field  
*element output, elset = eglobal  
s, e  
*node output, nset = nglobal  
u,  
*el print, elset = eglobal  
s, e  
*node print, elset = nglobal  
u
```



`*element output`: element quantities to be written to the `.odb` file



`*el print`: element quantities to be written to the ASCII `.dat` file
`s,e`: stresses and strains

The output files

- .odb: binary file containing the results (to be opened with Abaqus CAE or Abaqus Viewer)
- .dat: information about the model definition; tabular output of results
- .msg: diagnostic messages about the progress of the solution
- .sta: status of the solution process (can be monitored when the solution is running)

Remarks: the input file

- Defining the type of element in the *element keyword is mandatory. Some examples of commonly used elements:
 - T3D2: truss in 3D with 2 nodes
 - B33: beam in 3D and cubic shape functions (Euler-Bernoulli)
 - B31: beam in 3D and linear shape functions (Timoshenko)
- S4: shell with four nodes
- S4R: shell with four nodes and reduced integration
- C3D8: solid element with 8 nodes
- C3D20: solid element with 20 nodes

Remarks: the input file

- Each element of the model has to be associated to a section definition. The definition of the section is different depending on the kind element adopted:
 - Truss → *solid section
 - Beam → *beam section
 - Shell → *shell section
 - Continuum element → *solid section

Remarks: running the analysis

- After generating the .inp file with a text editor (Notepad, Crimson Editor, ...), save it into a folder
- Open the dos command prompt and move to the folder of .inp file
- A preliminary check is suggested to verify the model
 - `abaqus j=job_name datacheck interactive`
 - warning/errors messages are written in the .dat file
- Running the analysis
 - `abaqus j=job_name interactive continue`
 - `abaqus j=job_name interactive` (if datacheck is not performed)
- Analysis of the results
 - ASCII file in the .dat file
 - graphical post-processing of the .odb using Abaqus viewer: `abaqus viewer`

Remarks: generating the input file from Matlab

- In Matlab, the fprintf function can be used to write the Abaqus input file. Recall the syntax:

```
fprintf(fileID, formatSpec, A1, ..., An)
```

- fileID: file identifier
- formatSpec: format of the output field



- `A1,...,An`: numeric or character arrays

Remarks: generating the input file from Matlab

- Assume the matrix of node IDs and coordinates has been computed

```
nodes =  
1.0000      0      0      0  
2.0000    10.0000      0      0  
3.0000    20.0000      0      0  
4.0000      0    2.5000      0  
5.0000    10.0000    2.5000      0  
6.0000    20.0000    2.5000      0
```

- The Abaqus input file can be written using the following Matlab code:

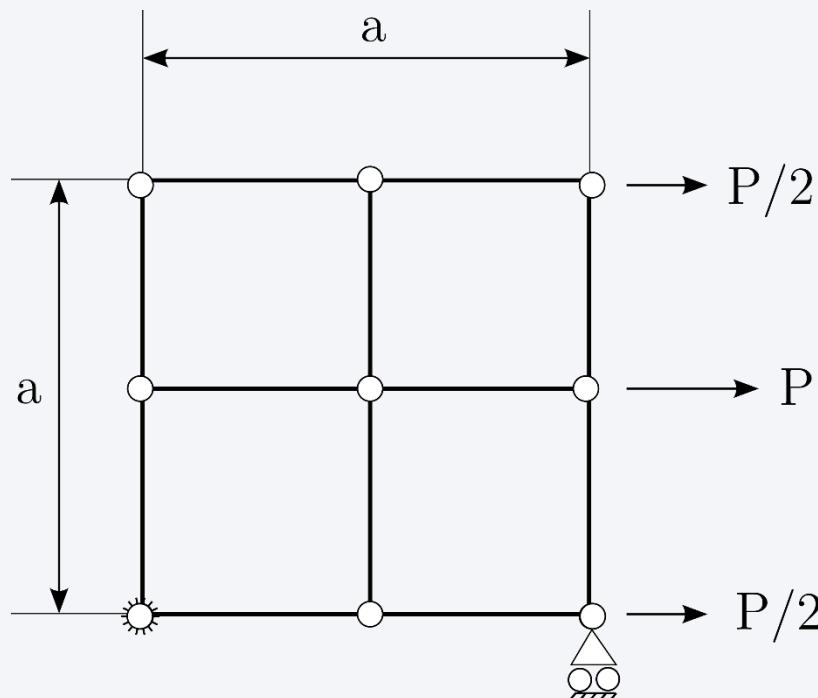
```
fileID = fopen('input_abaqus.inp','w');  
fprintf( fileID, '%20s\n', '*node, nset = nglobal');  
fprintf( fileID, '%2d, %6.2f, %6.2f, %6.2f\n', nodes' );  
fclose( fileID );
```

- The content of the input_abaqus.inp file is:

```
*node, nset = nglobal  
1, 0.00, 0.00, 0.00  
2, 10.00, 0.00, 0.00  
3, 20.00, 0.00, 0.00  
4, 0.00, 2.50, 0.00  
5, 10.00, 2.50, 0.00  
6, 20.00, 2.50, 0.00
```

Exercise 1

- Realize the finite element model here below, by manually preparing the input file using the template reported before
- Write the nodal displacements and the stresses to the .odb and .dat files



Input data

$$a = 100 \text{ mm}$$

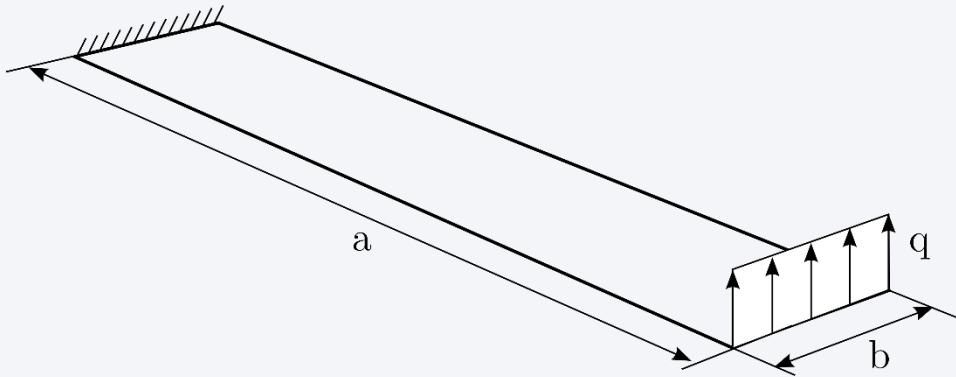
$$t = 1 \text{ mm}$$

$$E = 72 \text{ GPa}$$

$$\nu = 0.3$$

$$P = 10 \text{ N}$$

Exercise 2



Input data

$$a = 1000 \text{ mm}$$

$$b = 200 \text{ mm}$$

$$t = 1.5 \text{ mm}$$

$$q = 0.02 \text{ N/mm}$$

$$E = 72 \text{ GPa}$$

$$\nu = 0.3$$

- Use shell elements S4R (and *shell section to define the section property)
- Load discretization: the distributed load should be reported to the nodes
- Requests
 - Determine the displacement at the tip and compare it with the analytical solution (write this result to the .dat file)
 - Determine the maximum stress at the section at $x = 350 \text{ mm}$ and compare it with the analytical solution (write this result to the .dat file)
 - Study the convergence of the solution by considering different mesh sizes

Using the Abaqus Keyword Reference guide

The screenshot shows the main documentation page for Abaqus 2016. At the top left is the DS SIMULIA logo and "Abaqus 2016". A vertical teal bar on the right is labeled "DOCUMENTATION". At the top right are links for "PDF" (with an info icon), "Search All Guides", "Clear Search", "Advanced Search", and "Search Tips". Below these are two columns of links:

Category	Links
Modeling and Visualization	Abaqus/CAE User's Guide
Analysis	Abaqus Analysis User's Guide
Examples	Abaqus Example Problems Guide Abaqus Benchmarks Guide
Tutorials	Getting Started with Abaqus/CAE
Information	Using Abaqus Online Documentation
Installation and Licensing	Abaqus Installation and Licensing Guide
Reference	Abaqus Keywords Reference Guide (highlighted with a red box)
Programming	Abaqus Scripting User's Guide Abaqus Scripting Reference Guide Abaqus GUI Toolkit User's Guide Abaqus GUI Toolkit Reference Guide
Abaqus 2016 Update Information	Abaqus Release Notes

Example: *node keyword

*NODE Specify nodal coordinates.

This option is used to define a node directly by specifying its coordinates. Nodal coordinates given in this option are in a local system if the [*SYSTEM](#) option is in effect when this option is used.

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

Type: Model data

Level: Part, Part instance, Assembly

Abaqus/CAE: Mesh module

Link(s) for additional information

Reference:

- [“Node definition,” Section 2.1.1 of the Abaqus Analysis User’s Guide](#)

Optional parameters:

→ Optional: to be defined if needed, but not mandatory

INPUT

Set this parameter equal to the name of the alternate input file containing the data lines for this option. See [“Input syntax rules,” Section 1.2.1 of the Abaqus Analysis User’s Guide](#), for the syntax of such file names. If this parameter is omitted, it is assumed that the data follow the keyword line.

NSET

Set this parameter equal to the name of the node set to which these nodes will be assigned. Node sets created or modified with this option will always be sorted.

SYSTEM

Set SYSTEM=R (default) to give coordinates in a rectangular Cartesian coordinate system. Set SYSTEM=C to give coordinates in a cylindrical system. Set SYSTEM=S to give coordinates in a spherical system. See [Figure 14.10–1](#).

The SYSTEM parameter is entirely local to this option. As the data lines are read, the coordinates given are transformed to rectangular Cartesian coordinates immediately. If the [*SYSTEM](#) option is also in effect, these are local rectangular Cartesian coordinates, which are then immediately transformed to global Cartesian coordinates.

Example: *node keyword

Data lines to define the node: →

First line:

- 1. Node number.
- 2. First coordinate of the node.
- 3. Second coordinate of the node.
- 4. Third coordinate of the node.
- 5. First direction cosine of the normal at the node (optional).
- 6. Second direction cosine of the normal at the node (optional). For nodes entered in a cylindrical or spherical system, this entry is an angle given in degrees.
- 7. Third direction cosine of the normal at the node (optional). For nodes entered in a spherical system, this entry is an angle given in degrees.

Data to be defined in the rows after the keyword definition

The normal will be used only for element types with rotational degrees of freedom. See [Part VI, "Elements," of the Abaqus Analysis User's Guide](#).

Repeat this data line as often as necessary.

Example: *cload keyword

*CLOAD

Specify concentrated forces and moments.

This option is used to apply concentrated forces and moments at any node in the model. The [*CLOAD](#) option can also be used to specify a fluid reference pressure for incompressible flow in an Abaqus/CFD analysis and to specify concentrated buoyancy, drag, and inertia loads in an Abaqus/Aqua analysis.

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

Type: History data

Level: Step

Abaqus/CAE: Load module

References:

- [“Concentrated loads,” Section 34.4.2 of the Abaqus Analysis User’s Guide](#)
- [“Abaqus/Aqua analysis,” Section 6.11.1 of the Abaqus Analysis User’s Guide](#)
- [“Analysis of models that exhibit cyclic symmetry,” Section 10.4.3 of the Abaqus Analysis User’s Guide](#)
- [“Defining ALE adaptive mesh domains in Abaqus/Explicit,” Section 12.2.2 of the Abaqus Analysis User’s Guide](#)

Applying concentrated loads

Data lines to define concentrated loads for specific degrees of freedom:

First line:

1. Node number or node set label.
2. Degree of freedom.
3. Reference magnitude for load.

Repeat this data line as often as necessary to define concentrated loads.

Example: *shell section keyword

*SHELL SECTION

Specify a shell cross-section.

This option is used to specify a shell cross-section.

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

Type: Model data

Level: Part, Part instance

Abaqus/CAE: Property module

References:

- [“Shell elements: overview,” Section 29.6.1 of the Abaqus Analysis User’s Guide](#)
- [“Using a shell section integrated during the analysis to define the section behavior,” Section 29.6.5 of the Abaqus Analysis User’s Guide](#)

Required parameter:

→ Mandatory !

ELSET

Set this parameter equal to the name of the element set containing the shell elements for which the section behavior is being defined.

Required, mutually exclusive parameters:

→ Choose between these two options

COMPOSITE

Include this parameter if the shell is made up of several layers of material.

MATERIAL

Set this parameter equal to the name of the material of which the shell is made.

Example: *shell section keyword

- In this case the data to insert depend on the choice of the parameters

Data line to define a homogeneous shell (the MATERIAL parameter is included):

First (and only) line:

1. Shell thickness. This value is ignored if the NODAL THICKNESS or SHELL THICKNESS parameters are included.
2. Number of integration points to be used through the shell section. The default is five points if Simpson's rule is used and three points if Gauss quadrature is used. The number of integration points must be an odd number for Simpson's rule and is equal to the number of temperature degrees of freedom at a node of the element if this section is associated with heat transfer or coupled temperature-displacement elements. The maximum number of points for Simpson's rule is 99, and in the case of heat transfer or coupled temperature-displacement elements it is 19. This number must be at least 2 and less than or equal to 15 for Gauss quadrature. For Simpson's rule it must be at least 3, except in a pure heat transfer analysis, where the number of integration points can be 1 for a constant temperature through the shell thickness.

Data lines to define a composite shell (the COMPOSITE parameter is included):

First line:

1. Positive scalar value defining layer thickness or the name of a distribution ("Distribution definition," Section 2.8.1 of the Abaqus Analysis User's Guide) that defines spatially varying layer thicknesses. A distribution for composite layer thickness can be used only for conventional shell elements (not continuum shell elements). The layer thickness is modified if the NODAL THICKNESS or SHELL THICKNESS parameter is included.
2. Number of integration points to be used through the layer. The default is three points if Simpson's rule is used and two points if Gauss quadrature is used. The number of integration points must be an odd number for Simpson's rule, and it determines the number of temperature degrees of freedom at a node of the element if this section is associated with heat transfer or coupled temperature-displacement elements. The maximum number of points for Simpson's rule is 99, and in the case of heat transfer or coupled temperature-displacement elements it is 19. This number must be less than or equal to 15 for Gauss quadrature.
3. Name of the material forming this layer.
4. Name of the orientation to be used with this layer, an orientation angle, ϕ , or the name of a distribution ("Distribution definition," Section 2.8.1 of the Abaqus Analysis User's Guide) that defines spatially varying orientation angles. If the name of an orientation is used, the orientation cannot be defined with distributions. Orientation angles (in degrees) are measured positive counterclockwise relative to the orientation definition given with the ORIENTATION parameter. If the ORIENTATION parameter is not included, ϕ is measured relative to the default shell local directions (see "Orientations," Section 2.2.5 of the Abaqus Analysis User's Guide).
5. Name of the ply. Required only for composite layups defined in Abaqus/CAE.

Final remarks: creating the model

- No built-in system of units is provided. A consistent system has to be defined
 - m, N, kg, s, Pa, kg/m³
 - mm, N, t, s, MPa, t/mm³
- Preliminary analysis
 - perform preliminary calculations to compare with FEM results is always a good practice
 - nice contours do not mean good results
- Beginning with simple models
 - initial models do not need to be highly accurate
 - it is easier to identify errors on simple models
 - even in the context of nonlinear problems, an initial linear analysis can provide useful information

Final remarks: creating the model

- Element size
 - the use of very fine mesh has to be avoided, if not necessary
 - good practice to begin with a coarse mesh, and refine step by step
 - regular shapes guarantee, in commercial finite element codes, a better accuracy. Ideally, quadrangular element should be square
 - abrupt mesh transitions or warped elements should be avoided

Final remarks: verification phase

- Pay attention to the visualization options available on the post-processor (e.g. animations to view the results of static analyses are meaningless!)
- Check the satisfaction of the boundary conditions
- The finite element model is stiffer than the real structure
- To avoid interpretation errors, check how the results are plotted. Are the stresses, for instance, averaged at the nodes?
- Unexpected stress gradients can be due to modeling errors

Final remarks: possible sources of errors

- Singularities in the stiffness matrix (can the structure undergo a rigid body motion?)
- Nodes not connected to the elements
- Coincident nodes
- “Double” elements (responsible for unexpected local stiffening)