



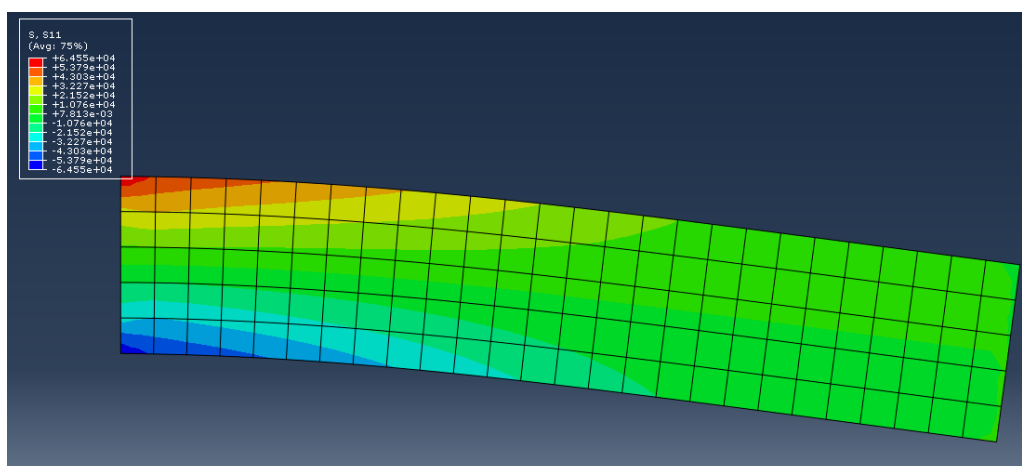
Universidad Politécnica de Madrid

Máster en Ing. de Caminos, Canales y Puertos

Computational Methods in Civil Engineering. Practical cases.

Chapter 1. Introduction to the Finite Element Method

Grupo de Mecánica Computacional



Introduction

The goals of this practical class are:

1. To introduce the applied concepts associated with numerical analysis by the finite element method.
2. To introduce the use of the finite element program Abaqus[®] (version *Interactive*) by simulating a linear problem.

The practice consists of two sections:

Section 1. Description of the finite element analysis procedure of a linear mechanical problem with the Abaqus[®] program.

Section 2. A exercise proposed to be solved by the student.

1 Introduction to Numerical Analysis with Abaqus

1.1 Motivation

Introduction to the Abaqus program. Abaqus is a program (or suite of programs) to perform finite element analysis with two ways of working:

1. **Interactive:** Using the graphical user interface Abaqus/CAE (look at Fig. ??).
 - It is not necessary to learn the syntax of the program's input file.
 - It is more laborious to change a parameter and perform a new analysis.

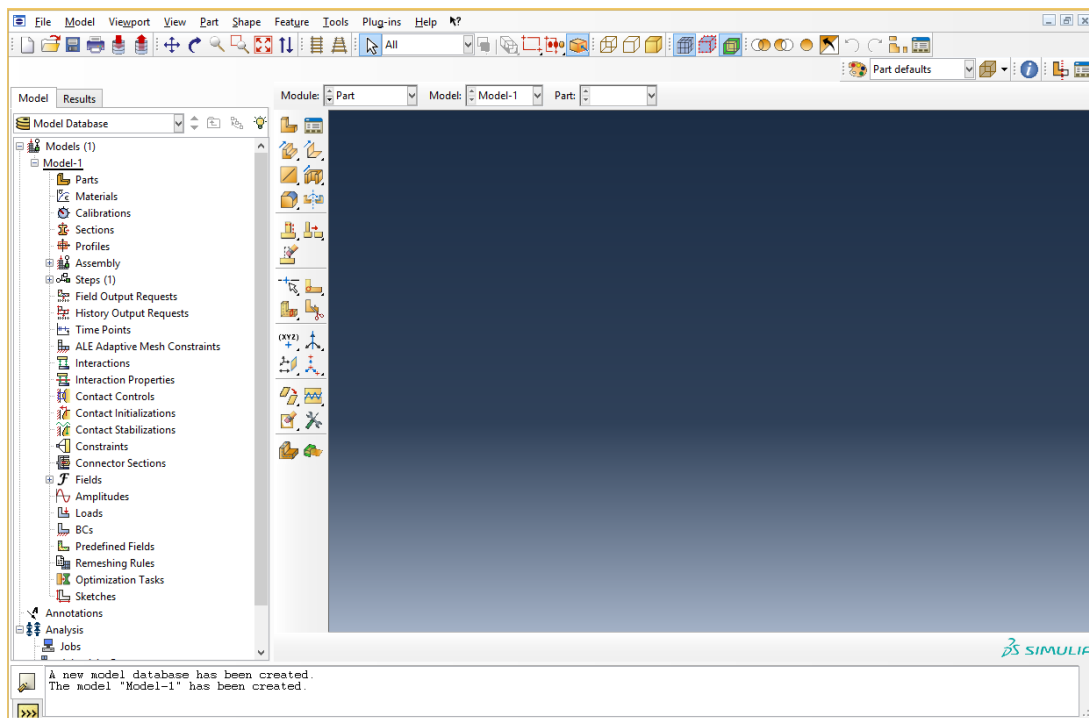


Figure 1: Abaqus/CAE graphic interface

2. **Keywords:** Writing a plain text input file `.inp` with the commands needed to perform the analysis (look at Fig. ??).
 - You must know the syntax of the program's input file.
 - It is very easy to modify something and perform a new analysis.
 - Although we work in *interactive* mode, at the end Abaqus always generates a plain text file.

We are going to work in this course in *interactive* mode. To start The Abaqus/CAE graphic preprocessing program we must write **abaqus cae** at the command console or click the direct link in the computer. Once started, among the options the program gives us we choose **Create Model Database with Standard/Explicit Model** as indicated in Fig. ??).

We have the Abaqus/CAE working environment where we can identify the following sections (look at Fig. ??):

```

*HEADING
Two-dimensional overhead hoist frame
SI units (kg, m, s, N)
1-axis horizontal, 2-axis vertical
*PREPRINT, ECHO=YES, MODEL=YES, HISTORY=YES
**
** Model definition
**
*NODE, NSET=NALL
101, 0., 0., 0.
102, 1., 0., 0.
103, 2., 0., 0.
104, 0.5, 0.866, 0.
105, 1.5, 0.866, 0.
*ELEMENT, TYPE=T2D2, ELSET=FRAME
11, 101, 102
12, 102, 103
13, 101, 104
14, 102, 104
15, 102, 105
16, 103, 105
17, 104, 105
*SOLID SECTION, ELSET=FRAME, MATERIAL=STEEL
** diameter = 5mm --> area = 1.963E-5 m^2
1.963E-5,

**
** History data
**
*STEP, PERTURBATION
10kN central load
*STATIC
*BOUNDARY
101, ENCASTRE
103, 2
*CLOAD
102, 2, -10.E3
*NODE PRINT
U,
RF,
*EL PRINT
S,
*****
** OUTPUT FOR ABAQUS QA PURPOSES
*****
*EL FILE
S,
*NODE FILE
U, RF
*END STEP

```

Figure 2: Example .inp file

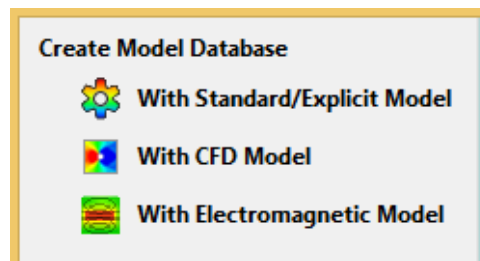


Figure 3: Start of the program Abaqus

- *Viewport*: Main part of the Abaqus/CAE screen where we visualize the pre and post processing of our analysis.
- *Model Tree View*: All the steps we follow in our model (everything we define on it) are represented in this area in the form of tree nodes. Each node is subdivided into several subnodes with their corresponding functionalities. If the *Model Tree* is not visible, make it visible by pressing **View/Show Model Tree** or pressing **CTRL+T**.
- *Toolbar Section*: Each node of the *Model Tree* corresponds to a toolbar where the user can access the associated commands. We can access these commands also through the drop-down menus of the top bar (although in this tutorial we will use the graphic icons of the toolbar).
- *Prompt region*: When we select a certain command, information about the next action to be performed appears in the *Prompt region*.

Stages of analysis To carry out a finite element analysis, we must perform a series of steps. Abaqus has grouped these steps into **Modules** sequentially. The functionalities

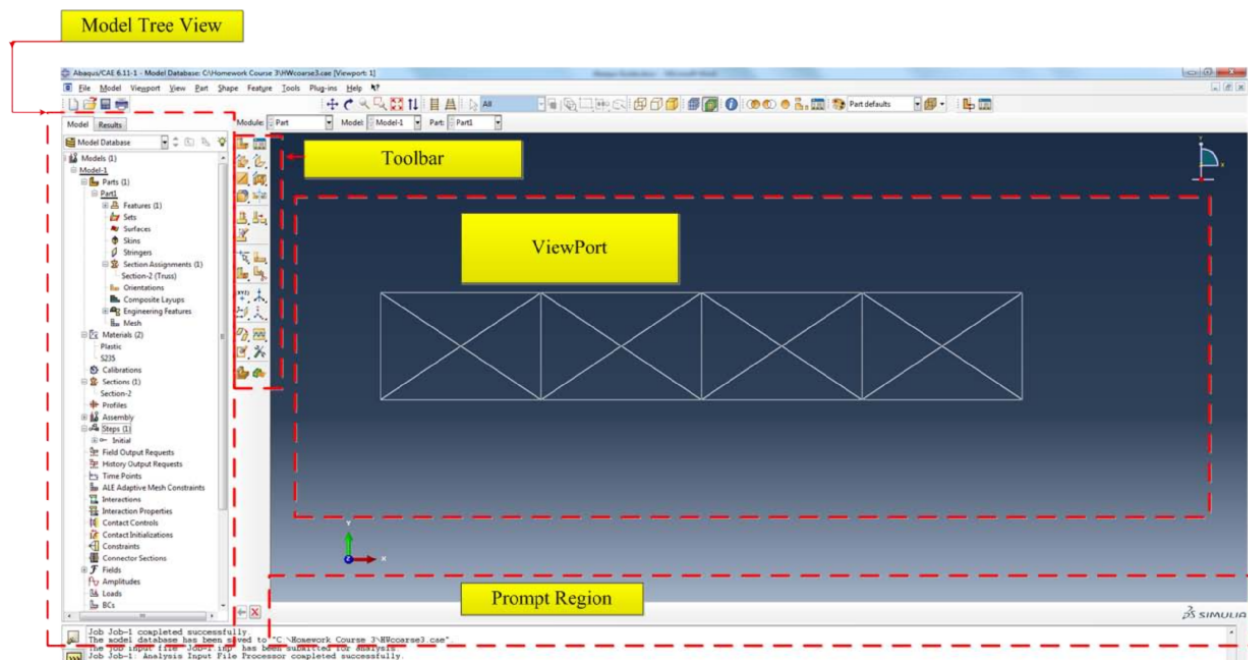


Figure 4: Abaqus/CAE working space description

associated with each module ¹ is summarized now:

Module Part. Create the geometry of the domain's elements. (*Parts*)

1. Sketch the geometry of the domain's elements.
2. Create *Parts* with the geometry of the domain's elements.

Module Property. Define materials and sections.

1. Define the properties of the materials.
2. Define the sections (which we associate with the materials).
3. Assign the sections to the corresponding *Parts*.

Module Assembly. Assemble the model by creating copies (instances) of the *Parts*.

Module Step. Set the analysis procedure.

1. Define calculation steps
2. Define the type of analysis in each calculation step (thermal, mechanical, stationary, transient, ...)
3. Define the variables that the program must save to display in the post process.
4. Define the parameters of the numerical algorithms used in each calculation step.

Module Interaction*. Create constraints between elements of our geometry.

Module Load. Apply the boundary conditions at each step of time.

¹Modules with an asterisk * will not to be used in this course

Module Mesh. Create the mesh.

Module Optimization*. Use finite element analysis to optimize a property of our model.

Module Job. Create the job and launch the numerical analysis.

Module Visualization. Perform the post-process.

Description of the problem to be solved. One of the first things we must do before building the numerical model is to decide which system of units we are going to use. Abaqus does not have a predefined system, we simply have to work on a consistent unit system. In Fig. ?? four systems of consistent units are shown:

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	Pa (N/m^2)	MPa (N/mm^2)	lbf/ft^2	psi (lbf/in^2)
Energy	J	mJ (10^{-3} J)	ft lbf	in lbf
Density	kg/m^3	tonne/mm^3	slug/ft^3	$\text{lbf s}^2/\text{in}^4$

Figure 5: Consistent unit systems

The problem we want to solve is summarized in Fig. ?. It is a beam clamped in its left end that has a tension vector $\mathbf{t}^* = [0, -1000, 0]^T$ Pa applied to the right half of its upper face. The measures of the beam are 5 meters in length and a square section of 1 meter in side. The material is concrete with elastic constants $E = 27$ GPa and $\nu = 0.3$. As you can see, our problem can be reduced to a 2D problem in plane stress.

Starting of the analysis with Abaqus. Before starting to work let us do three things:

1. Defining a working directory in **File/Set Work Directory** where all the files we generate will be saved.
2. See help from Abaqus:
 - If you go to **Help/On Context** we will get help on the icon (command) that we click afterwards.
 - If you go to **Help/On Module** we will get help on the module you are in at the moment.
 - If you go to **Help/Search and Browse Guide** we will go to the main page of Abaqus help.
3. We assign a name to our model. Place the cursor over the model at the top of the *Model Tree*, right click the mouse, select **Rename** and assign it the name *Mensula*. Be sure you obtain something like Fig. ?.

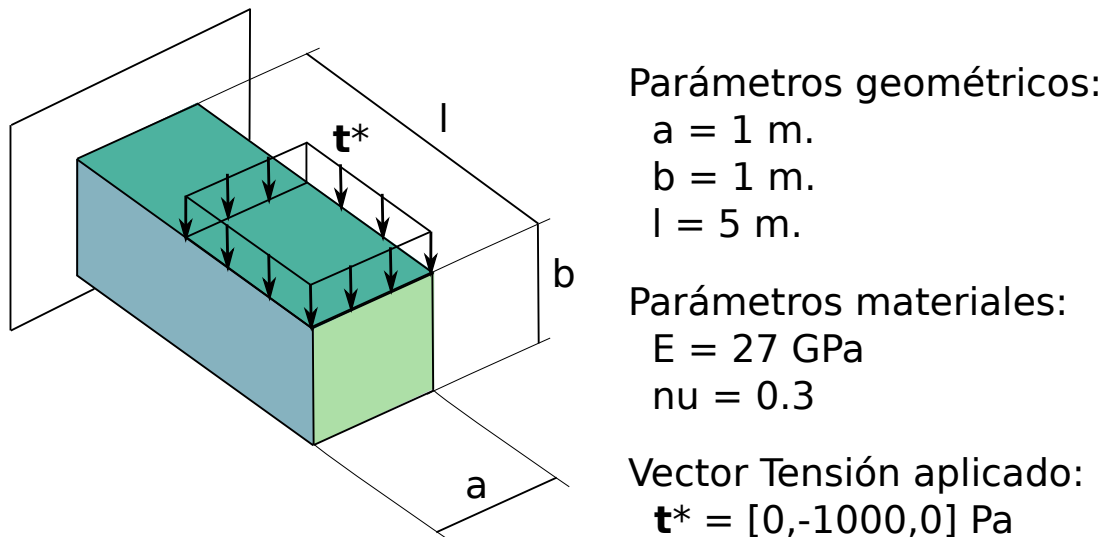


Figure 6: Description of model

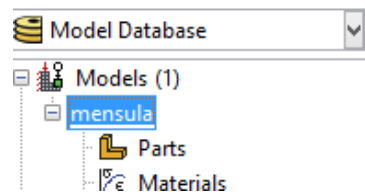


Figure 7: Change of the model's name

In the remaining of this chapter we will describe all the phases of the numerical analysis by using the example described in Fig. ??.

1.2 Module Part. Create the geometry of the domain's elements.

In Abaqus we must define the geometry of the elements that form our model. These elements are called *parts* and will allow us to assemble a model by creating one or more copies (instances) of each *part*, in case there are repeated elements.

In order to define the geometry of the elements of our model:

1. We activate module **Part** and we press command **Create Part** (look at Fig. ??).

In the lower left corner Abaqus shows a message (Prompt) that tells us what to do (in this case “*Fill out the Create Part dialog*”). Press the **Cancel** button to cancel the task we are doing, the **Previous** button to cancel the step we are doing inside a task and return to the previous step and the **Done** button to finish the task (see Fig. ??).

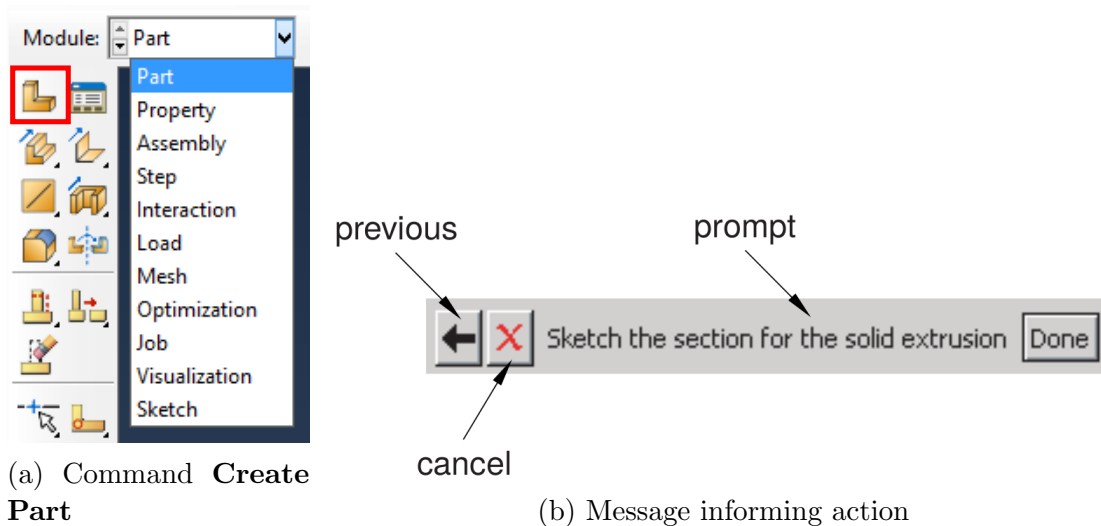
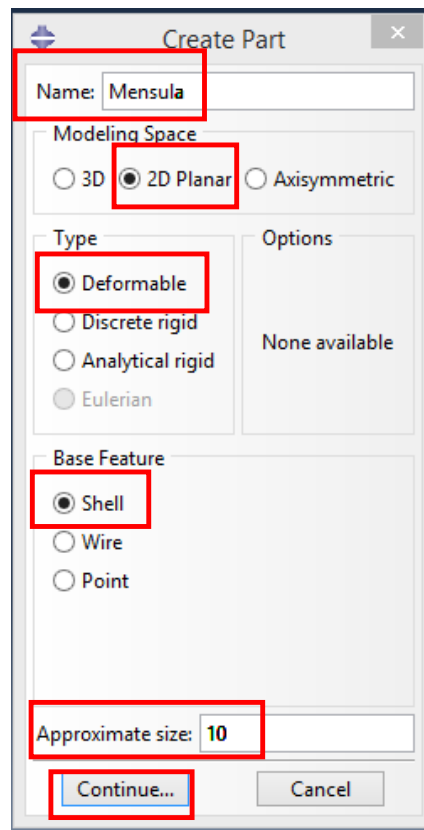


Figure 8: Onset of module *Part* and *Prompt* message of information

We see now the dialog **Create Part** where we are asked if our model is going to be a three-dimensional, 2D planar or axisymmetric domain (**modeling space**), if it is going to be a deformable or rigid object (**Type**) and what type of geometry is (**Base Feature**). For our problem we will choose the options indicated in Fig. ?. We put a value of 10 (meters in our assumed unit system) in the variable **Approximate size**, which is double the maximum dimension of our domain. With this dimension Abaqus will provide us with a working environment where we can create the geometry.

2. Next we get a screen of a working environment (*Sketcher*) with the CAD tools needed to create the geometry of our *part*. Regarding the sketcher we must know:
 - You can use the points of the working window (by clicking the mouse) to define geometric entities.
 - Dotted lines are the X and Y axes and intersect at the origin.
 - The orientation of the work plane is defined with the axes at the bottom left.
 - When you activate a drawing tool, the coordinates are drawn in the upper left.

Figure 9: Dialog **Create Part**

There are two ways to define this geometry: (a) define the geometric entities precisely and (b) define them quickly using the grid and then apply constraints to achieve the desired geometry. Let's work the second way:

- (a) Press the **Create lines: Rectangle** button and draw a rectangle without worrying about the dimensions (see Fig. ??).

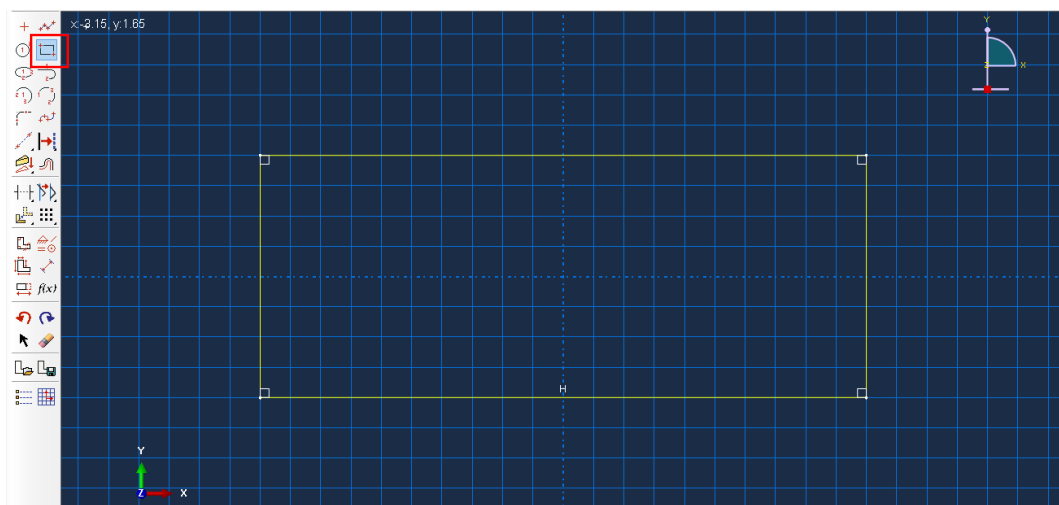


Figure 10: Initial Rectangle

- (b) Press the command **Add dimension**, select one of the horizontal sides and modify the dimension to 5 meters (see Fig. ??).

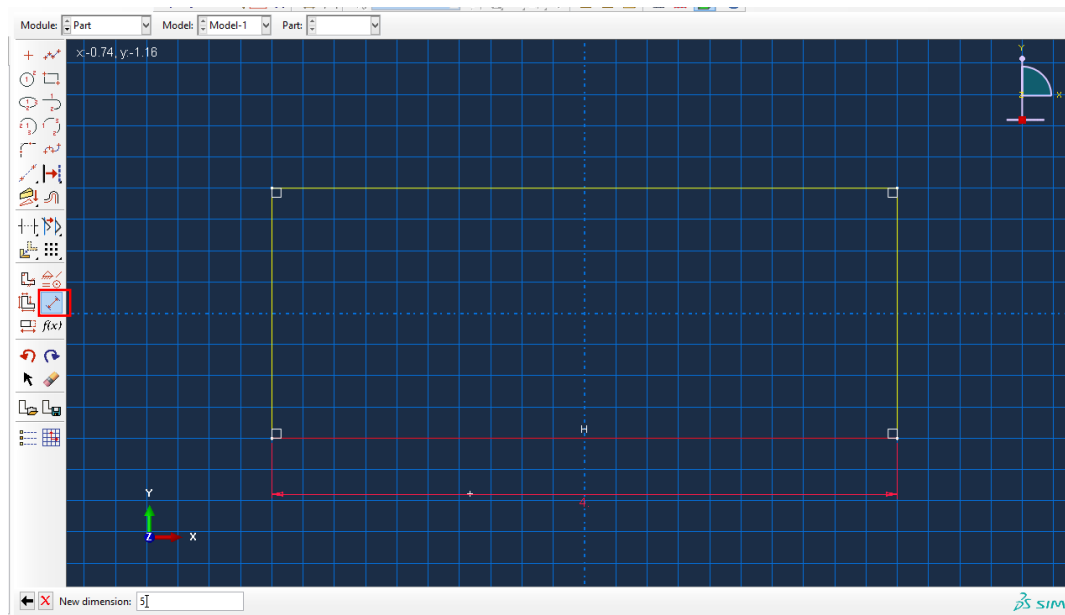


Figure 11: Modifying a Dimension

- (c) As in the previous action modifies the vertical dimension to 1 meter. Once done, press **Auto Fit view** to center the image (it should look like in Fig. ??).

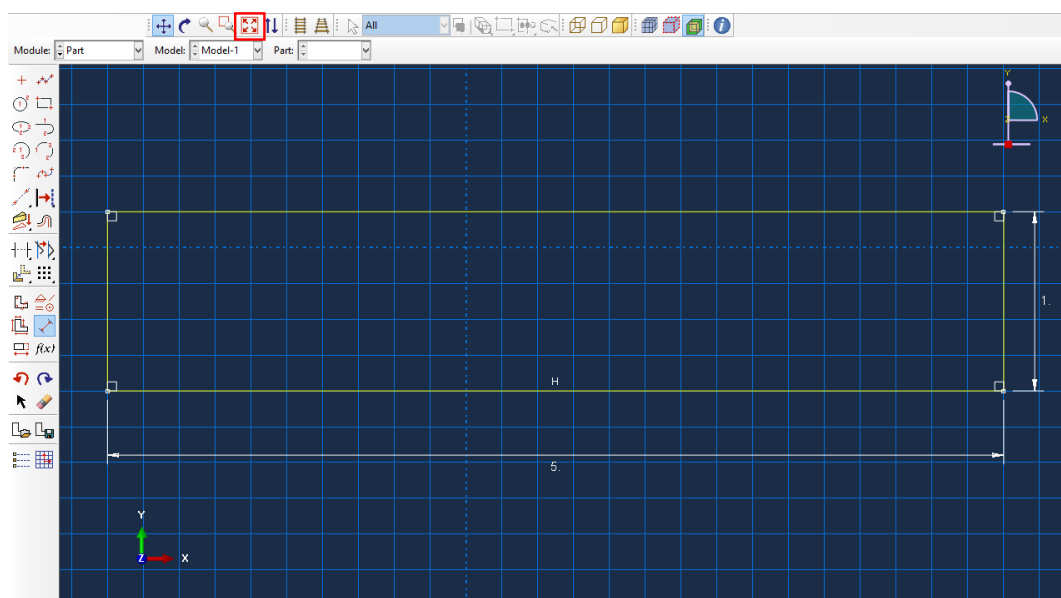
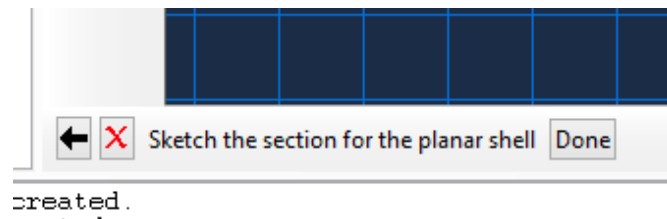
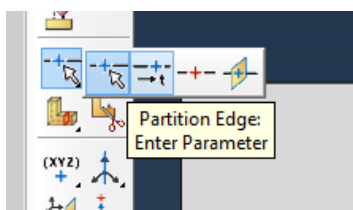


Figure 12: Final Rectangle

- (d) Finally, press cancel to quit the Sketcher tool and press **Done** to generate the *part* that we were looking for (see Fig. ??).

Figure 13: Final order to create object *Part*

3. If we look at the distributed loads of our model, we must divide the upper edge into two halves in order to assign the loads only in the right upper half as indicated in Fig. ???. To do so, follow these steps:
 - (a) Among the icons associated with modifying a *part*, press **Partition Edge: Enter Parameter** as shown in Fig. ??. Then select the upper edge of the beam (Fig. ??).

(a) Activation of **Partition Edge**

(b) Selection of upper edge

Figure 14: Split of upper edge

- (b) Then it asks for a parameter between 0 and 1 to indicate the cut-off point according to the direction of the vector that is seen on the screen. Since we want to divide by half, indicate 0.5 as indicated in Fig. ??.

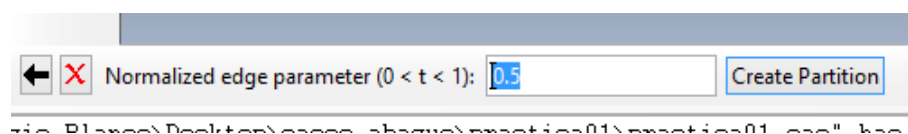


Figure 15: Introduction of split parameter

Once the *Part* is created, save the model with the **Save model database** command.

1.3 Module Property. Define materials and sections.

In this module we must define the materials, the sections and assign this information to the *part* we created in the previous section.

1. **Define the material.** We choose the **Property** module and click on the **Create Material** icon (see Fig. ??). In the window that appears to define the material we give a name, include a brief description and assign the constitutive mechanical behavior **Elastic** as summarized in Fig. ?. Finally we assign the mechanical properties of a standard concrete ($E = 27 \text{ GPa}$, $\nu = 0.3$) as shown in Fig. ?.

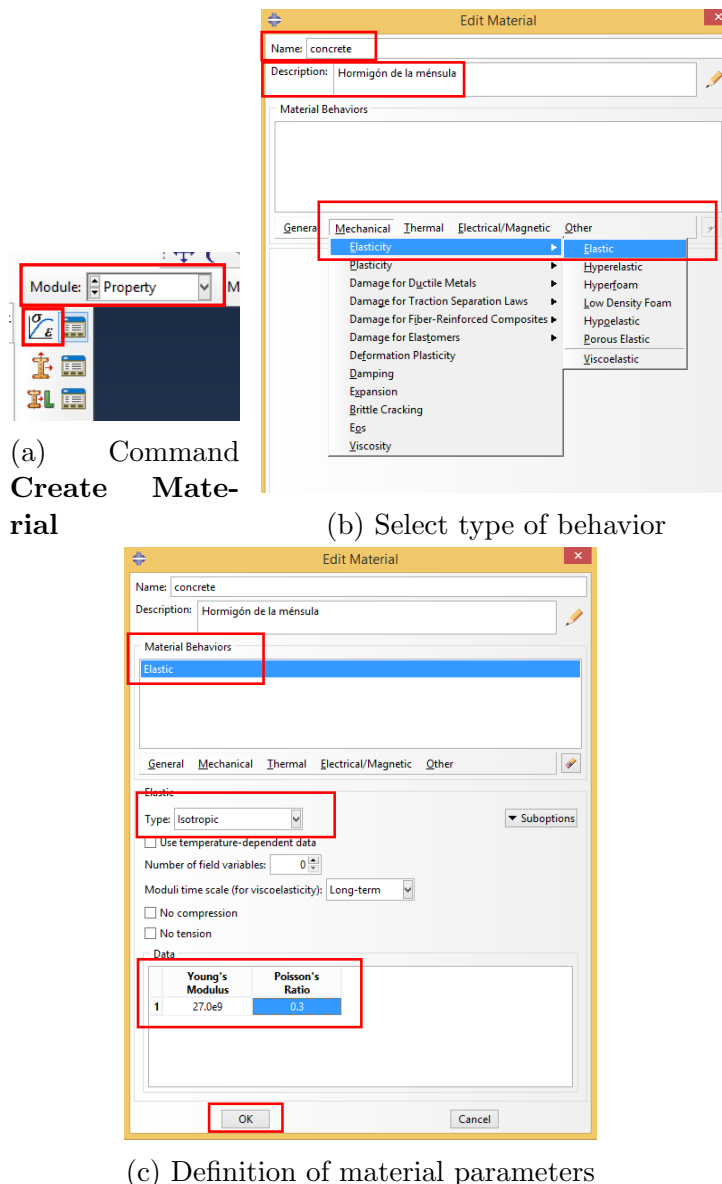
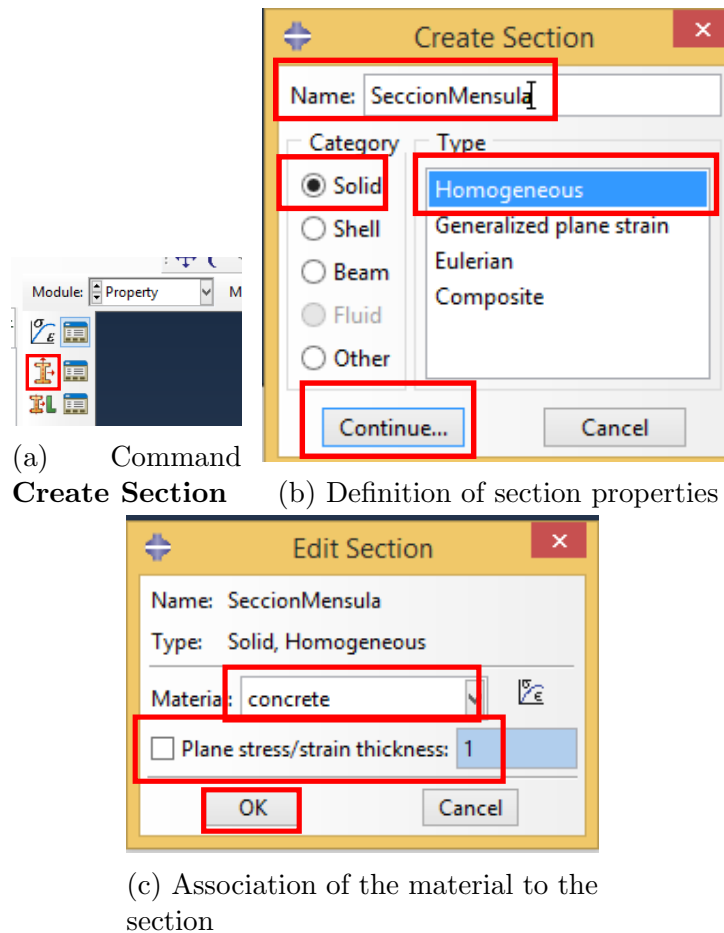
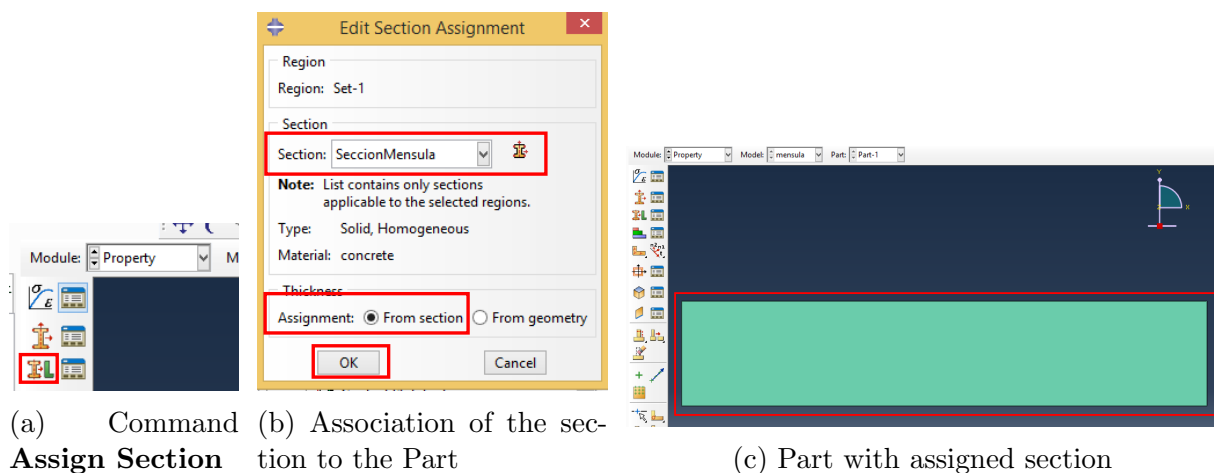


Figure 16: Definition of *concrete* material

2. **Define the section.** Still within the **Property** module click on the **Create Section** icon (see Fig. ??). In the next window (Fig. ??) let's assign the name *section-Mensula*, the category of finite element **Solid** and the type **Homogeneous**. Finally we associate this section with the concrete material as indicated in Fig. ?? (note that the default thickness of the piece is 1 meter, which matches our dimension).

Figure 17: Definition of section *SeccionMensula*

3. **Assign the section to a part.** Finally, in this module we must assign the previous section to the created *part*. To do so, click on the icon **Assign Section** (see Fig. ??) and select the region to which we want to assign the section (the beam). We choose we want to assign the section *SeccionMesula* (see Fig. ??) and our *part* should have changed color as indicated in Fig. ??

Figure 18: Assigning the section to a *Part*

1.4 Module Assembly. Assemble the model

Every Abaqus model is constructed as a set of copies (*instances*) of the created *parts*. Our problem is very simple and we just have to create a set (*Assembly*) with a single copy of our *part*. However, imagine that we would like to model a car, and that we have created *parts* of its different components. With Abaqus's logic we would have created a *part* defining a wheel, and at the time of assembling the final model, we would make four copies of that *part* (one for each wheel of the car).

In order to assembly our model follow the next steps:

1. Activate the module **Assembly** and press the **Create instance** icon as shown in Fig. ??.
2. In the dialog box that appears (see Fig. ??), select the *part* we are going to make a copy of (there is only one in our problem) and indicates that the copy will be of type **Dependent** (we will define the mesh in the *part* and it will be replicated in the copy we are creating).

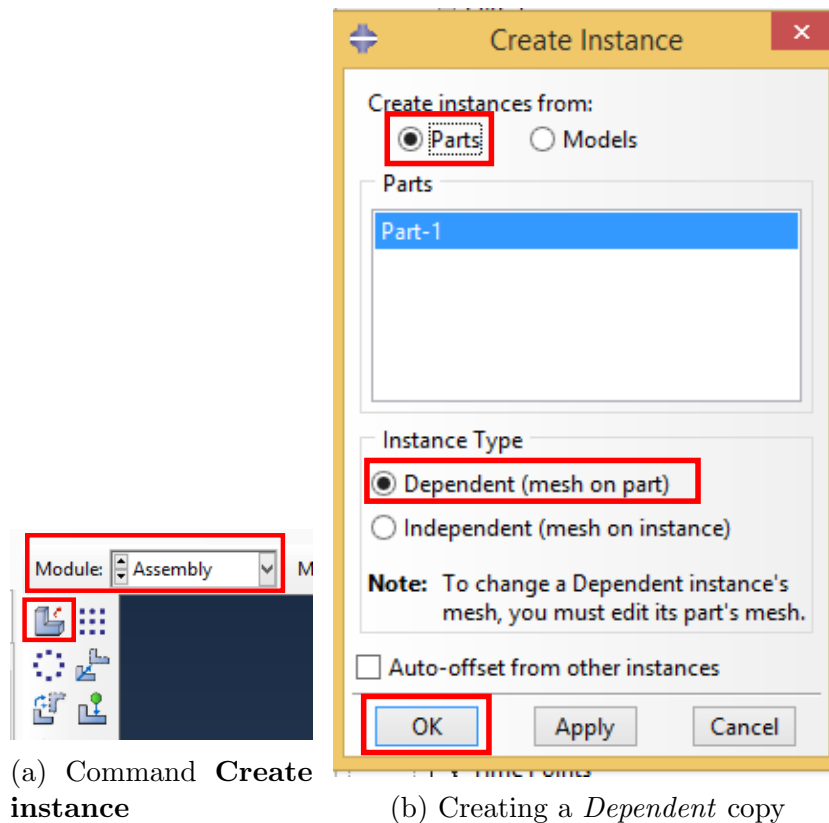
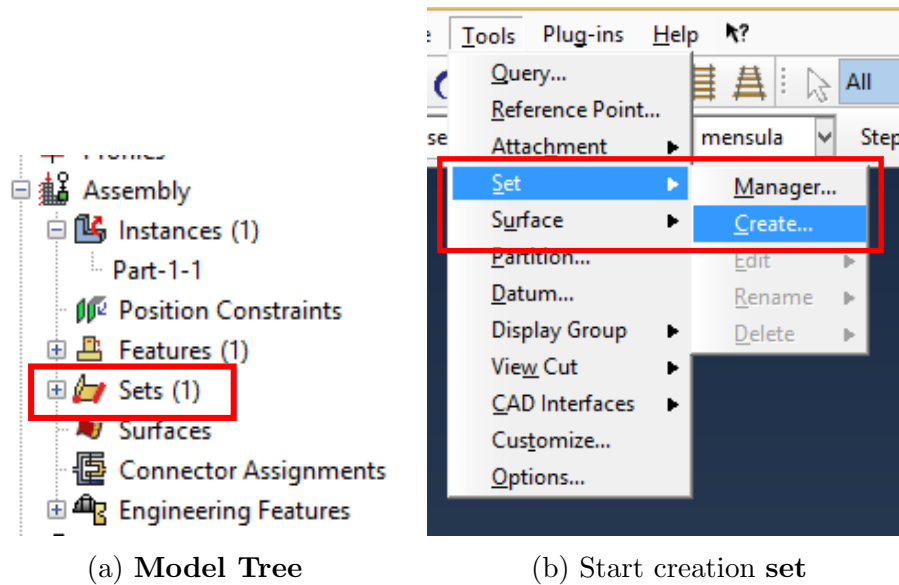


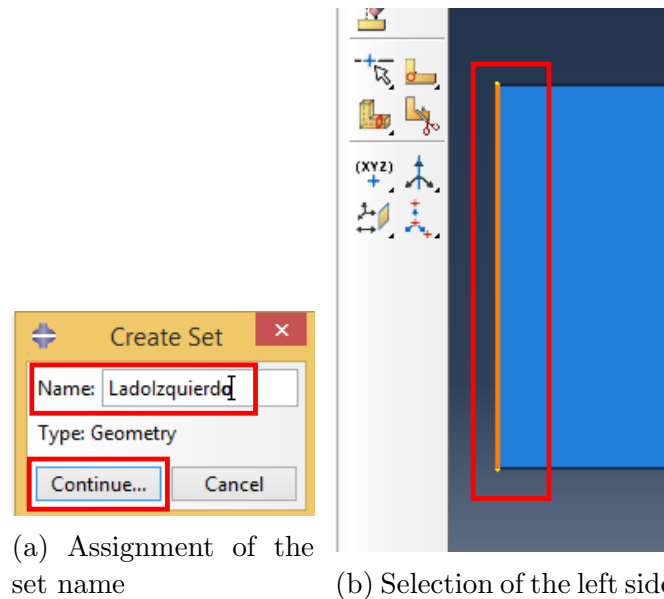
Figure 19: Action **Create Instance**

In order to facilitate later operations, Abaqus allows us to group nodal points of geometric entities into objects called **sets**. In this way, when we want to obtain the results in that geometric entity we can request it to Abaqus without needing to know the nodes of the mesh that form it (only the name of the **set**). In our case we want to know the distribution of the nodal forces in the clamped end of the beam. So we are going to create a set of the future mesh nodes in this geometry:

1. To create a set, double-click on the **Sets** element in the **Model Tree** (see Fig. ??) or press **Tools/Set/Create** (see Fig. ??).

Figure 20: Creation of a **set**

2. Set the name *LadoIzquierdo* in next window (see Fig. ??) and select the left side of the beam (see Fig. ??).

Figure 21: Creation of a **set**

3. Make sure that in the Model Tree a new set has been created (as shown in Fig. ??)

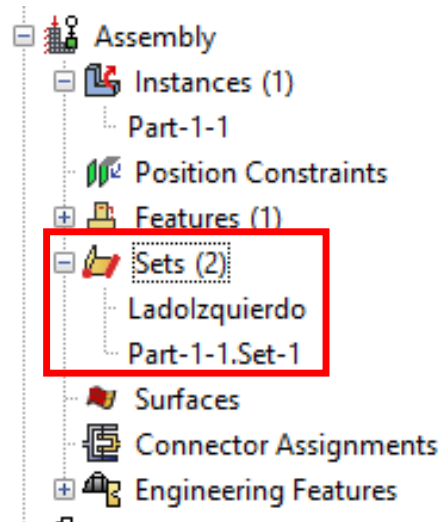


Figure 22: Model Tree with two sets

1.5 Module Step. Configure the analysis procedure.

Once the model is built, we must define in Abaqus the calculation steps (**steps**) that we will use in our analysis and what information we want to keep from each one. Abaqus defines a calculation step *Initial* that we will assume is the beginning of our analysis (everything that happens before our analysis) where, in our case, we will apply the boundary condition in displacements (the encastre) at the left side of the beam. After that, to apply the imposed tension vector, we will define a single calculation step and perform a static analysis. To define the calculation step (**Step**) do as follows:

1. Activate the **Step** module and click on the **Create Steps** icon as shown in Fig. ???. In the dialog box that appears (see Fig. ??) name the case as *Static Case* and indicates that the analysis is of **Static, General** type. Set everything else as default.

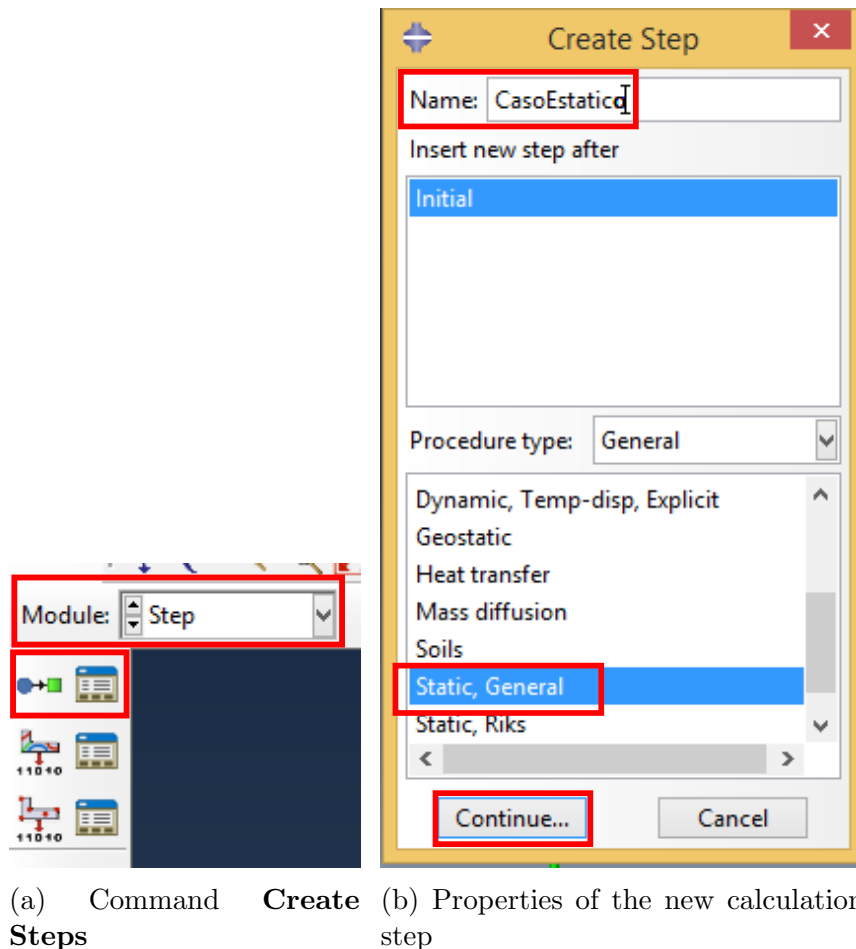
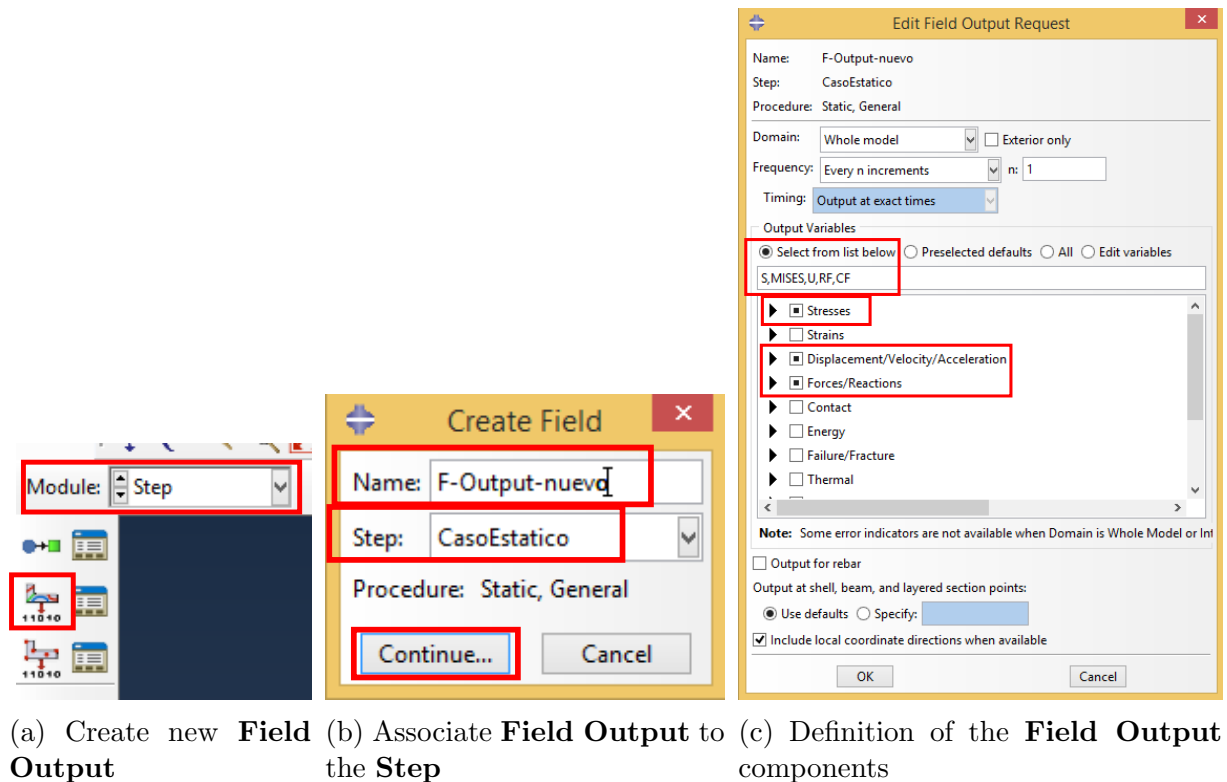
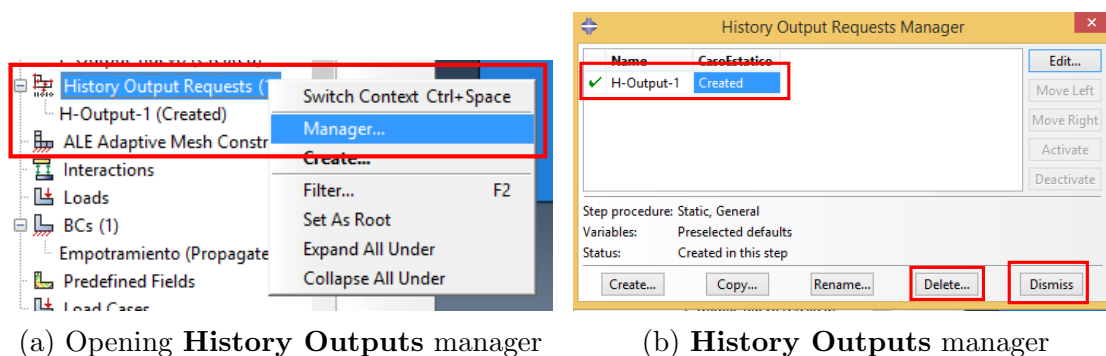


Figure 23: Create new calculation step

- Now let's decide what information we want to keep from the calculation step created to be able to analyze it afterwards it in the postprocess. Under the name **Field Output** Abaqus stores solution information in the space domain at a given time. Select the **Create Field Output** icon (see Fig. ??) to create a new result field. In the following dialog name it *F-Output-New* and assign it to the interval just created *CasoEstatico* (see Fig. ??). From the fields available for postprocessing, we will choose the stress tensor (S), the von Mises equivalent stress (MISES), the displacements (U), the reaction forces and moments (RF) and the forces and moments concentrated (CF) as indicated in Fig. ??.

Figure 24: **Field Output** definition

- Under the name **History Output** Abaqus stores the solution information at a spatial point during a time interval. Since we are in a quasi-static problem we are not interested in the temporal evolution of a variable and we are going to eliminate the one that Abaqus creates by default. To do so press the right mouse button on the **History Output Request** node of the **Model Tree** and select **Manager** as indicated in Fig. ?? . We will see a window with the historical variable manager (see Fig. ??), select the variable **H-Output-1** and press **Delete**.

Figure 25: Delete of **History Outputs**

1.6 Module Load. Apply the boundary conditions

We want to apply essential boundary conditions (we impose null displacement in the encastre) and natural boundary conditions (we impose a tension vector \mathbf{t}^* in the right half of the upper part of the beam). The encastre will be applied in the calculation step *Initial* and the tension vector in the calculation step *CasoEstatico* previously created.

In order to impose the essential boundary condition do as follows:

1. Activate the **Load** module and press **Create Boundary Condition** (see Fig.??).

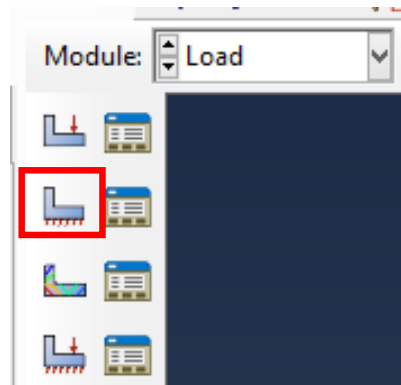
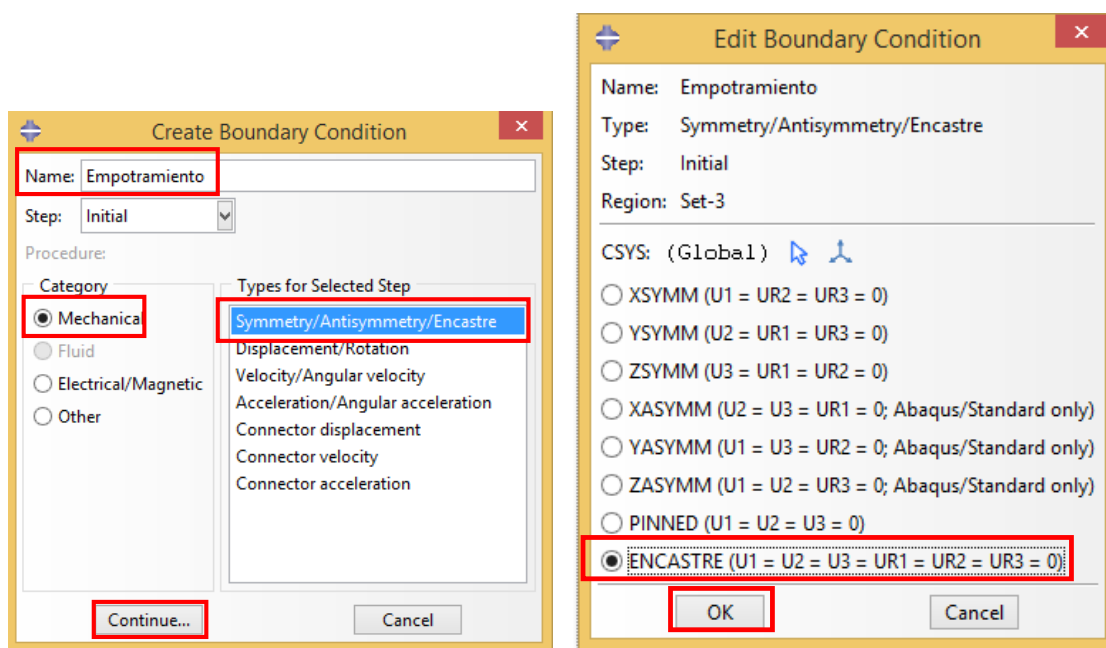


Figure 26: Command **Create Boundary Condition**

2. In the dialog box that appears (see Fig.??), put the name *Empotramiento* to the boundary condition, assign it to the calculation step **Initial**, set the category **Mechanical** and the type **Symmetry/Antisymmetry/Encastre**. Select the left side where we will impose the boundary condition. In the following dialog box set the boundary condition is of **Encastre** (clamped) type as indicated in Fig. ??.



(a) Creating of boundary condition

(b) Editing of boundary condition

Figure 27: Definition of boundary condition *encastre*

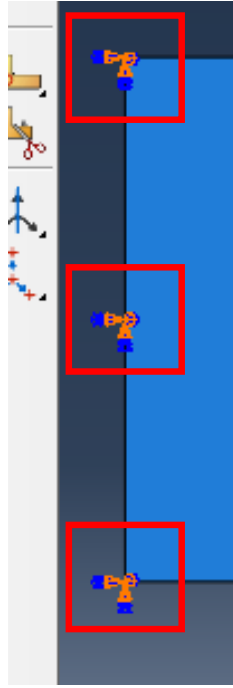


Figure 28: Detail of the encastre boundary condition

3. Finally, the model we have created has to appear on the working screen with the degrees of freedom on the left side of the beam set to zero (see Fig.??)

In order to impose the natural boundary condition do as follows:

1. Press the **Create Load** icon as shown in Fig. ???. In the dialog box that appears (see Fig. ??) name the boundary condition as *CargaRepartida*, apply it in the calculation step *CasoEstatico*, indicating that its category is **Mechanical** and the type **Surface traction**. Select the right half of the upper edge of the beam as described in Fig. ??.

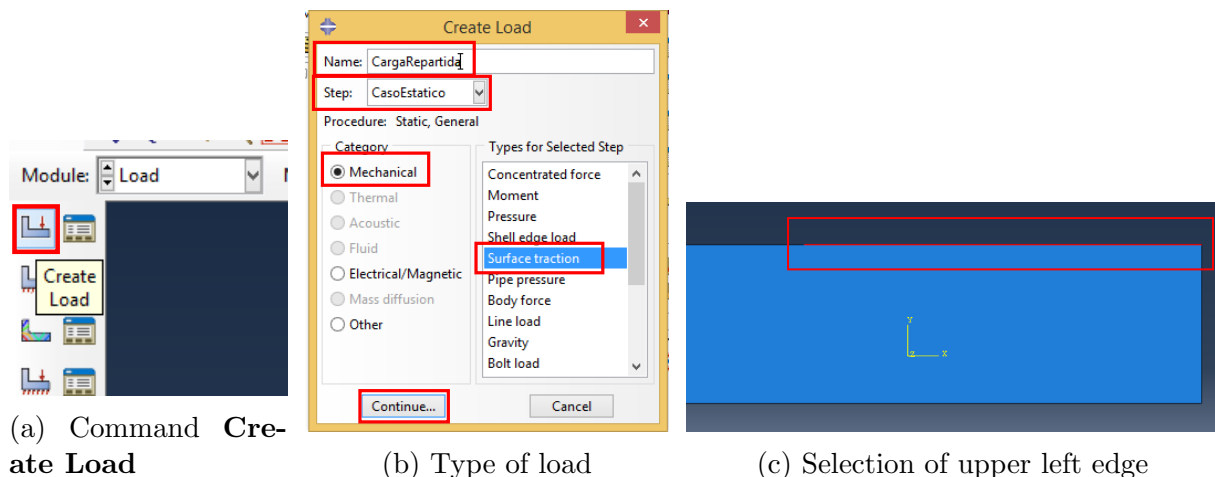
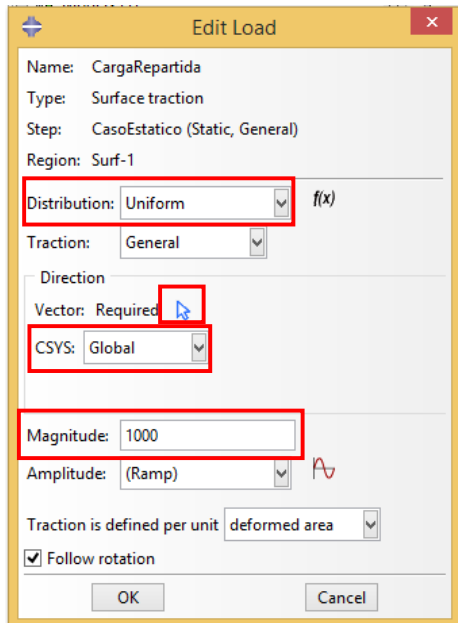


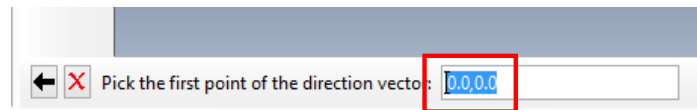
Figure 29: Definition of distributed load (I)

2. The following dialog box allows us to define the tension vector. Set the type as **General** and a module of value 1000 Pa (see Fig. ??). It is necessary to define the direction and sense of the tension vector. Click on the blue arrow next to the word

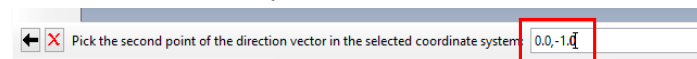
Vector to define a vector that gives us the direction and sense of the vector tension (it does not have to be unitary). It can be done by pressing two vertices of our geometry or by entering two pairs of coordinates. Let's do it the last way. Enter the coordinates of the starting point (Fig. ??, then press enter) and the end point (Fig. ??, then press enter).



a: Properties of tension vector



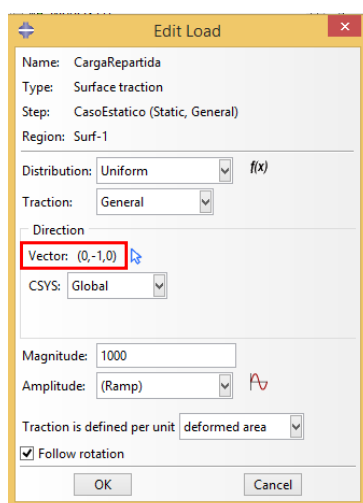
b: Start of auxiliary vector



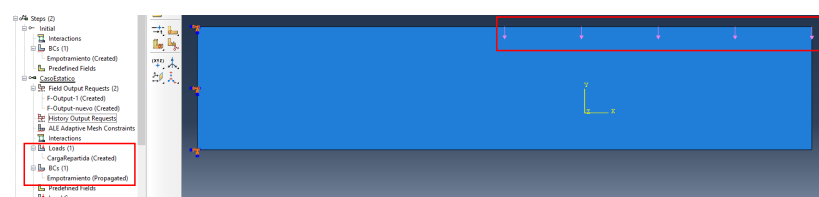
c: End of auxiliary vector

Figure 30: Definition of distributed load (II)

3. We now see the normalized vector in the previous dialog box (see Fig. ??) and, in accepting the definition, the load distribution scheme appears in the working window (see Fig. ??). Check in the **Model Tree** that the associated definitions have been added to the boundary conditions nodes (BCs and Loads).



(a) Properties of tension vector



(b) Detail of applied load

Figure 31: Definition of distributed load (III)

1.7 Module Mesh. Build the mesh.

In order to assign a mesh to a geometry we must inform Abaqus about:

1. The shape of the element (whether they are triangles or squares in two-dimensional problems and whether they are tetrahedral or hexahedral in three-dimensional problems).
2. Size of the element (either a global size for the geometry or by defining different sizes depending on regions).
3. The type of element (we will explain this concept in following classes).

Abaqus distinguishes which type of mesh can be made according to the geometry we have (although the user can split the *parts* manually to create more regular geometries). Each possible type of mesh in each region is identified with a color:

Green Meshes can be generated using structured methods

Yellow Meshes can be generated with the sweep method

Pink Meshes can be generated with the free method

Cinnamon Meshes can be generated with the button-up method

Orange You can not generate meshes and the geometry has to be divided

But in our case it happens that the geometry is blue. This is because we have made a dependent copy on the *part* and is the part the one that we have to mesh (then the mesh is replicated in the copy). Activate the **Mesh** module and change the *part* as the object to mesh as indicated in Fig. ?? and it will change to pink.

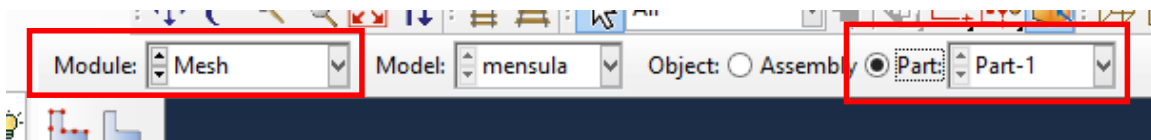


Figure 32: Start of Mesh module

To define the mesh, follow these steps:

1. First we will define the shape of the elements. Press **Mesh/Controls** in the upper menu bar and the dialog box in Fig. ?? appears. Let's set that we want to use **quadrilaterals**, the technique is **Free** and the algorithm is **Advancing front**.
2. Let us now define the size of the elements. Press the **Seed part** icon (see Fig. ??) to assign a global size to the elements of the geometry. Let us use a size of 0.2 m as shown in Fig. ?? and press the **Apply** button. We should see a division of the boundary of our geometry using the size we have indicated (see Fig. ??). Finally press OK.

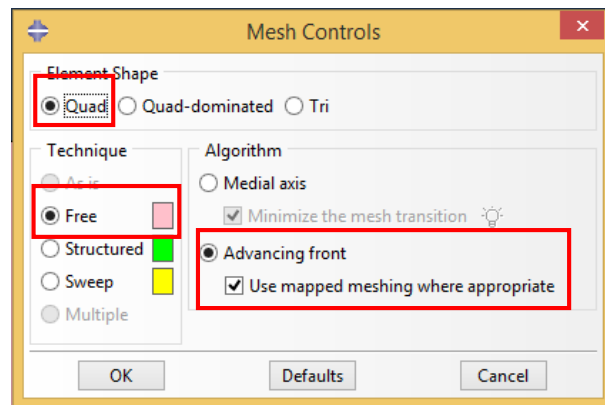


Figure 33: Definition of the element shape

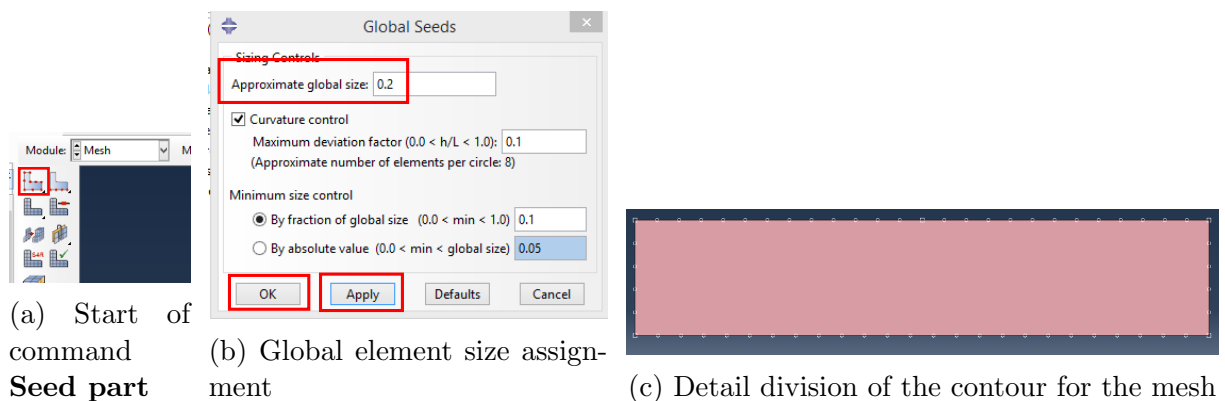


Figure 34: Mesh size assignment

- Now let's define the type of element by pressing the **Assign Element Type** icon (see Fig. ??). It will ask us to select the *Part* to which we will assign the type of element. When selecting it, it must change color as shown in Fig. ?. Finally a dialog box appears with the available element types. As shown in Fig. ?? we will select **Standard, Standard, Quadratic, Plane Stress** and **Reduced Integration** (you will learn in the remaining of the course what is the meaning of these names).

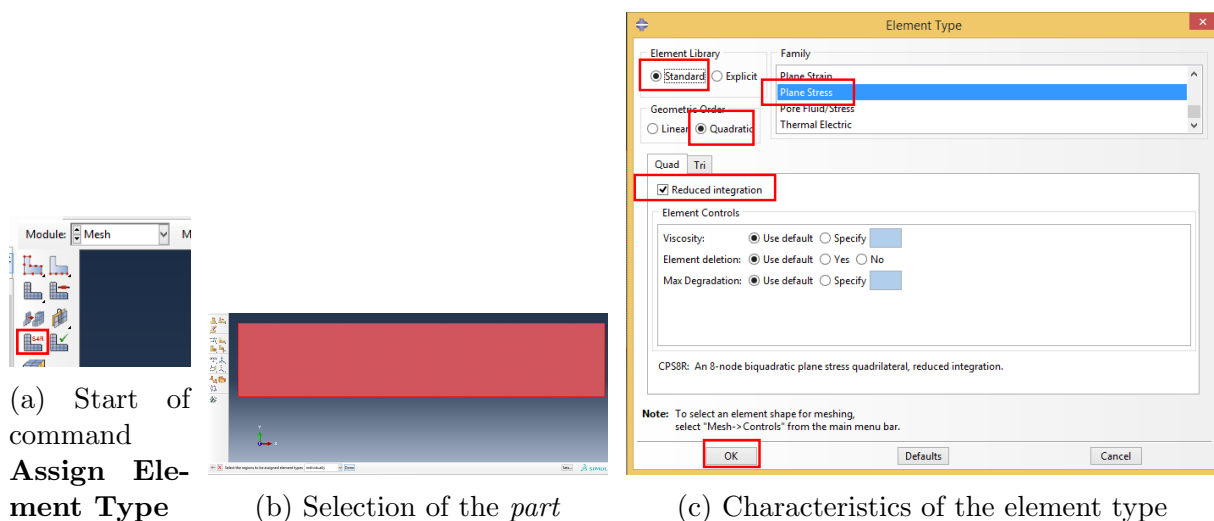


Figure 35: Element type definition

4. Once defined all the parameters of the mesh we only need to generate it. Click on the **Mesh Part** icon (see Fig. ??) and you should get the mesh shown in Fig. ??.

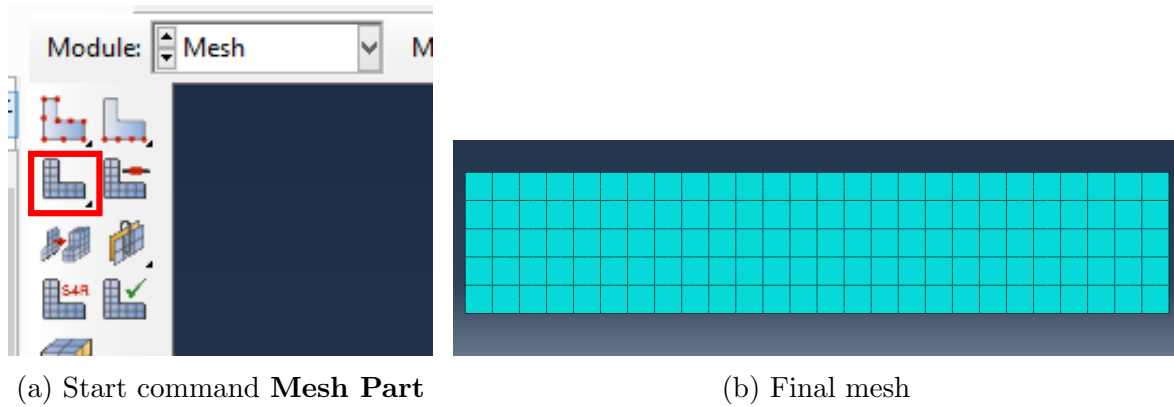


Figure 36: Generation of the mesh

1.8 Module Job. Create the job and launch the analysis.

At this point we must have a complete finite element model that can be executed. Before running the simulation in Abaqus, we need to create a **job** and launch it so that Abaqus understands that there is a finite element model ready to be simulated.

To create and launch a **job** follow these steps:

1. Activate the **job** module and press the **Create job** icon (see Fig. ??). Give the job's name *Caso01* and set that the data for the analysis must be taken from the Model we created and not from a data entry file *.inp* (see Fig. ??). You will see a dialog box to define the analysis (see Fig. ??). Include a description and set the remaining parameters by default.

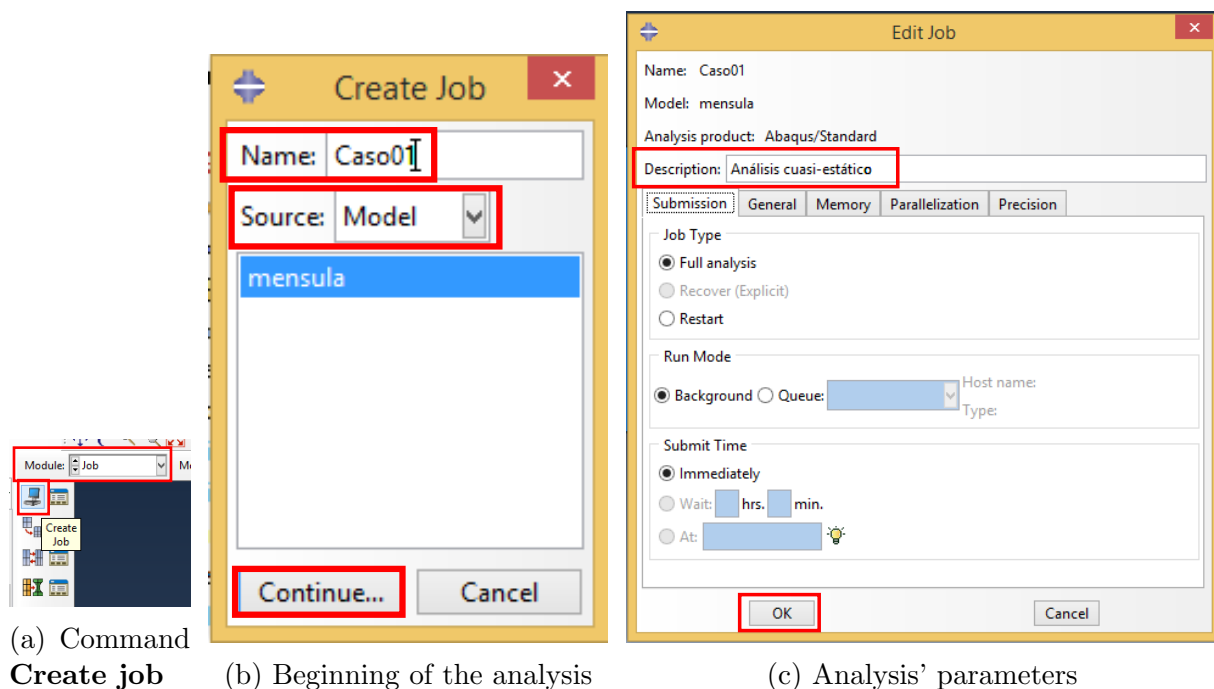


Figure 37: Creating a new analysis (**job**)

2. We see that a new job has been created in the **Model Tree** (under *Analysis*). To launch it, let's activate the job manager by right clicking on **Jobs** and selecting **Manager** (see Fig. ??). We see a dialog box (see Fig. ??) that allows us to manage all the analyzes we have (in this case we only have one, *Case01*). Press **Write Input** to write the file *case01.inp* to disk in case you would like to use it later.

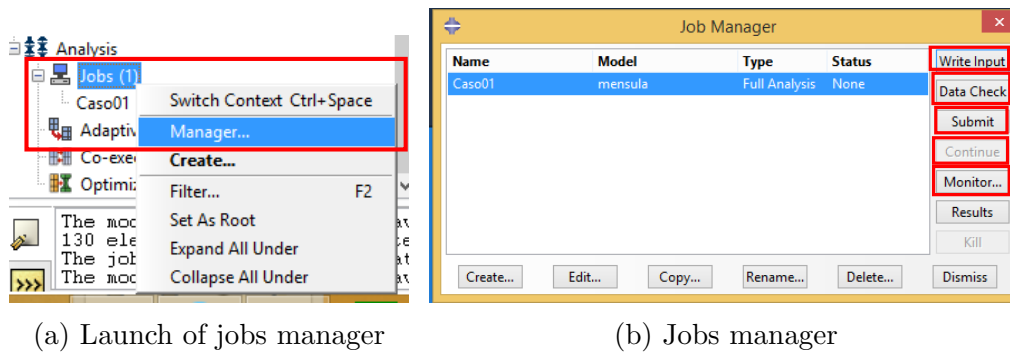


Figure 38: Launch of an analysis case (I)

3. Finally we can launch the analysis. We can launch it directly by pressing **Submit** or we can first check that everything is fine by pressing **Data Check** and then **Continue**. In either case, if all goes well, you should generate at the bottom of the screen the message of Fig. ?? and in the job manager the status of the job must have changed to **Completed** (See Fig. ??).

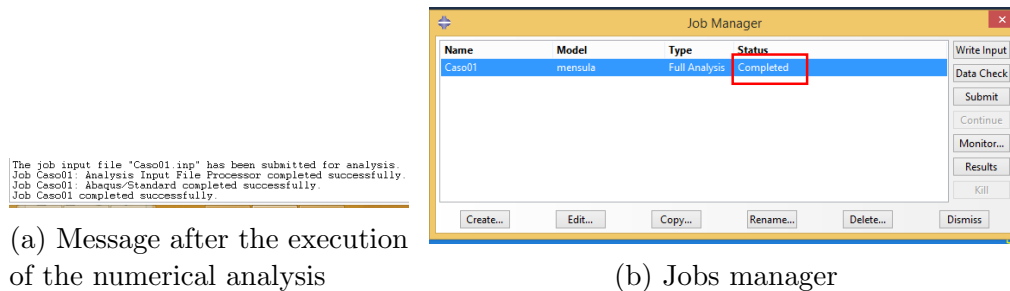


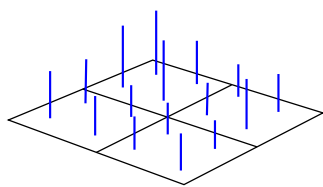
Figure 39: Launch of an analysis case (II)

1.9 Module Visualization. The post-process.

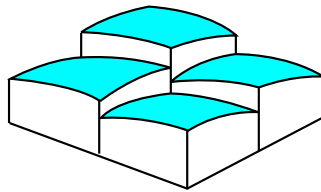
Before starting the visualization let's remember what types of variables we have and where they are calculated. First let's look at the types of variables and the information Abaqus gives about them:

- Scalar Variables (Temperature θ): 1 component
- Vector Variables (Displacement \mathbf{u}): 3 components + its module
- Tensorial variables of order 2 (stresses $\boldsymbol{\sigma}$): 6 components + Invariant values with respect to the coordinate system (von Mises, principal values, hydrostatic pressure)

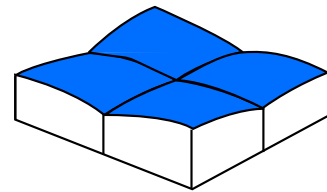
The variables displacement and temperature are calculated in the nodes of the mesh. The variables heat flux, strain and deformation are calculated at the integration points of the elements. These values are then extrapolated to the nodes of the elements and then smoothed in the nodes of the mesh (weighting the contribution of all elements that share the same node) as schematized in Fig. ??.



Valores de las tensiones en los puntos de integración



Valores de las tensiones extrapolados a los nodos del elemento



Valores de las tensiones suavizados en los nodos de la malla

Figure 40: Smoothing process of the variables defined in points of integration

To activate the post-processing of the analysis we just made click the right mouse button on the completed job *Caso01* and select **Results** as indicated in Fig. ??.

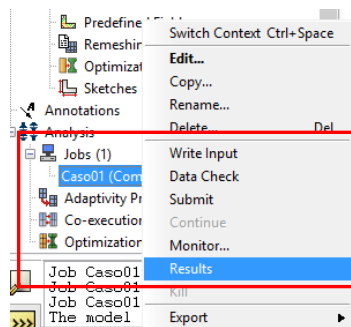


Figure 41: Start of results postprocessing

Below we will describe some of the most common actions that are performed in the postprocess of results:

1. Draw the deformed mesh.

With the **Visualization** module activated, click on the **Plot Deformed Shape** icon (see Fig. ??) to get the deformed shape. You will get the deformed shape shown in Fig. ?? where the displacements have been multiplied by a scale factor of $1633e4$.

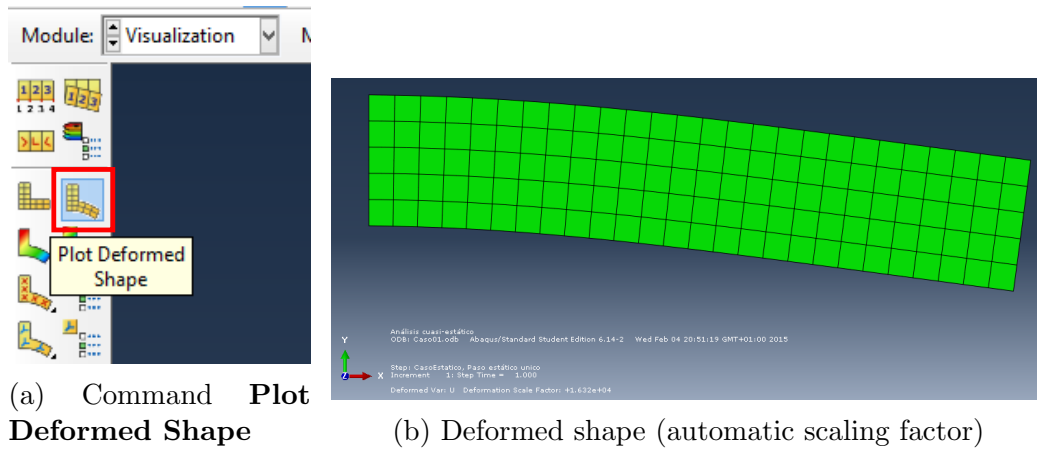


Figure 42: Representation of the deformed form (I)

To use a non-automatic scaling factor, press the **Common Options** icon (see Fig. ??). You will see the dialog box in Fig. ?? where you need to change the **Deformation Scale Factor** to **Uniform** and set a value of 10000. At the end you will get the deformed shape shown in Fig. ??.

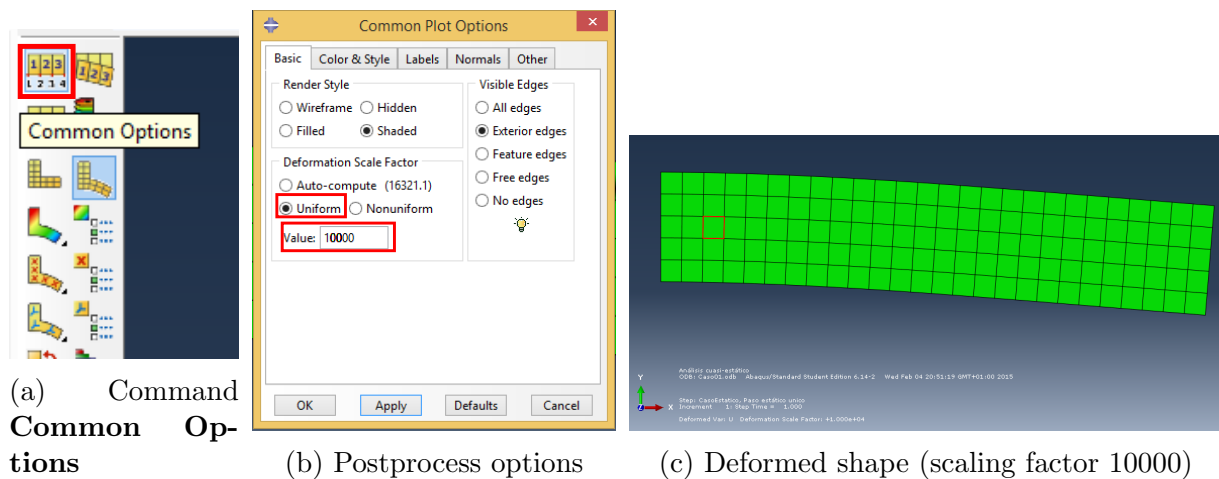


Figure 43: Representation of the deformed form (II)

2. Obtain the distribution of a scalar field (a scalar variable or a component of a vector or tensor variable) in the geometry

Click the **Plot Contours on Deformed Shape** icon or **Plot Contours on Undeformed Shape** (see Fig. ??). It will appear a field distribution of the solution as shown in Fig. ?? (in this case is the component σ_{xx} of the stress tensor). To change which variable you want to display, select the variable you want to draw from the drop-down menus in the upper toolbar (see Fig. ??).

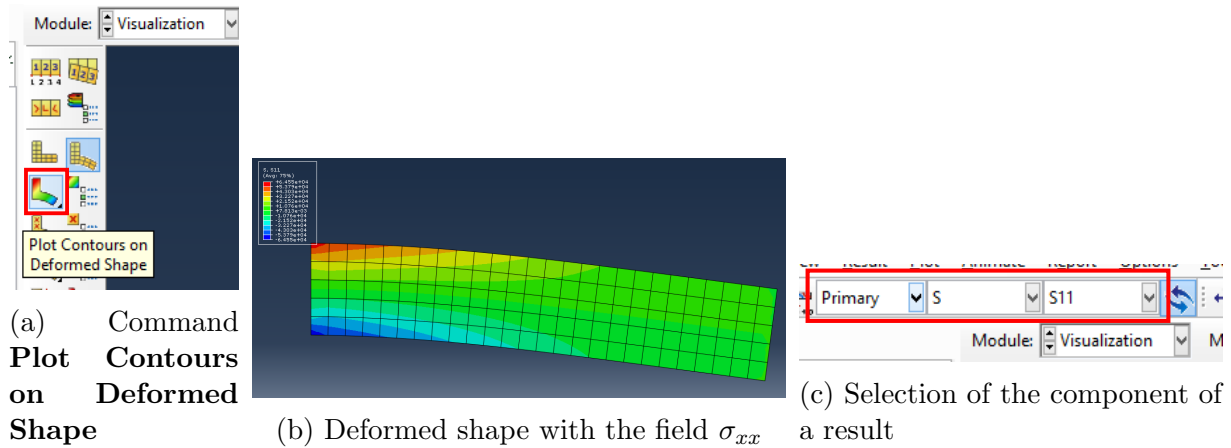


Figure 44: Distribution of a scalar field in geometry (I)

We can modify how these fields of the solution are presented. Press the **Contour Options** icon (see Fig. ??) and in the following dialog box select the contour type **Line** and the contour intervals **Discrete** equal to 20 such as shown in Fig. ?. You will have to obtain (for the variable you are painting) a distribution similar to that shown in Fig. ?. Finally you can save a graphic file with the data of the working screen by pressing **File/Print** to save a file .pdf, or press **CTRL + C** to save the image to your clipboard.

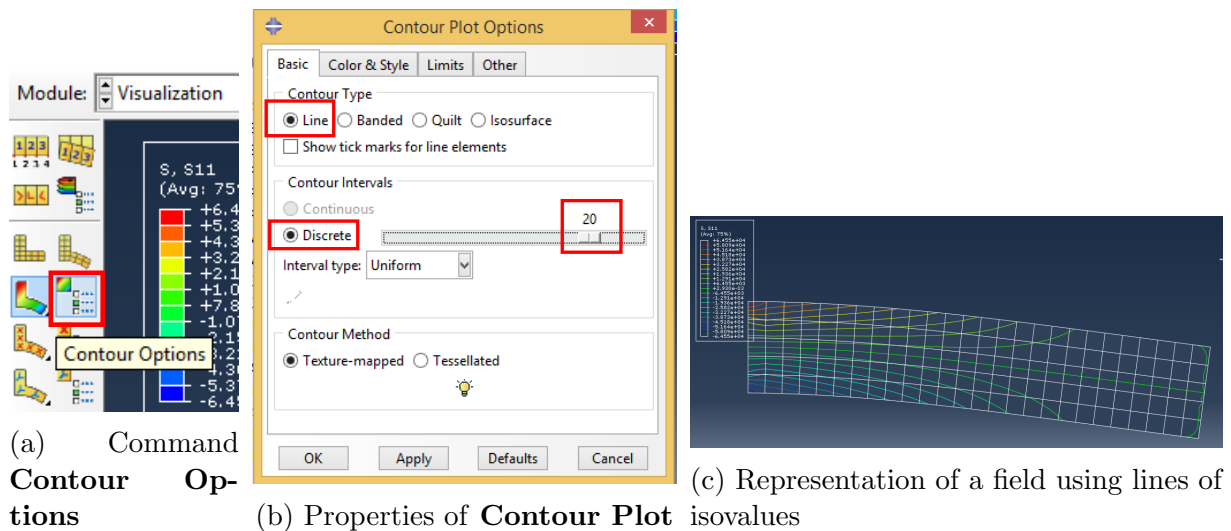


Figure 45: Distribution of a scalar field in geometry (II)

3. Draw vectors or principal components of tensors in the domain

Click the **Plot Symbols on Deformed Shape** icon or **Plot Symbols on Undeformed Shape** (see Fig. ??). You will see a distribution of vectors as shown in Fig. ?? (in this case it is the distribution of the principal components of the stress tensor at the integration points of the elements).

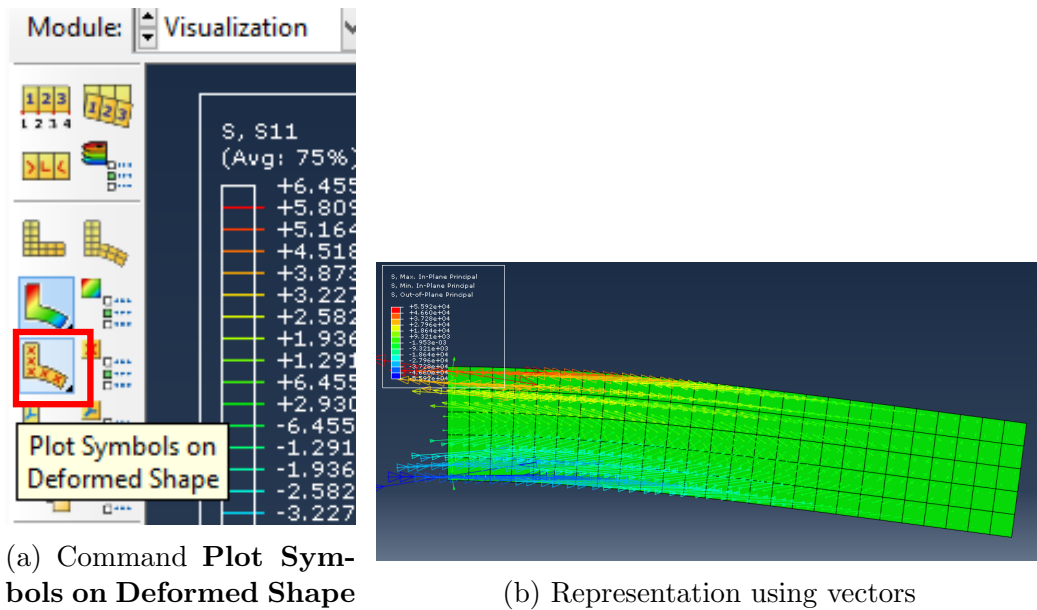


Figure 46: Distribution using vectors of a vectorial/tensorial magnitude (I)

We can modify the format of the vectors. In our case it would be useful to reduce the size a bit so that they can be better seen. To do so press the **Symbol Options** icon (see Fig. ??) and in the next dialog box (see Fig. ??) change the size to 2. You should get a distribution similar to that shown in Fig. ??.

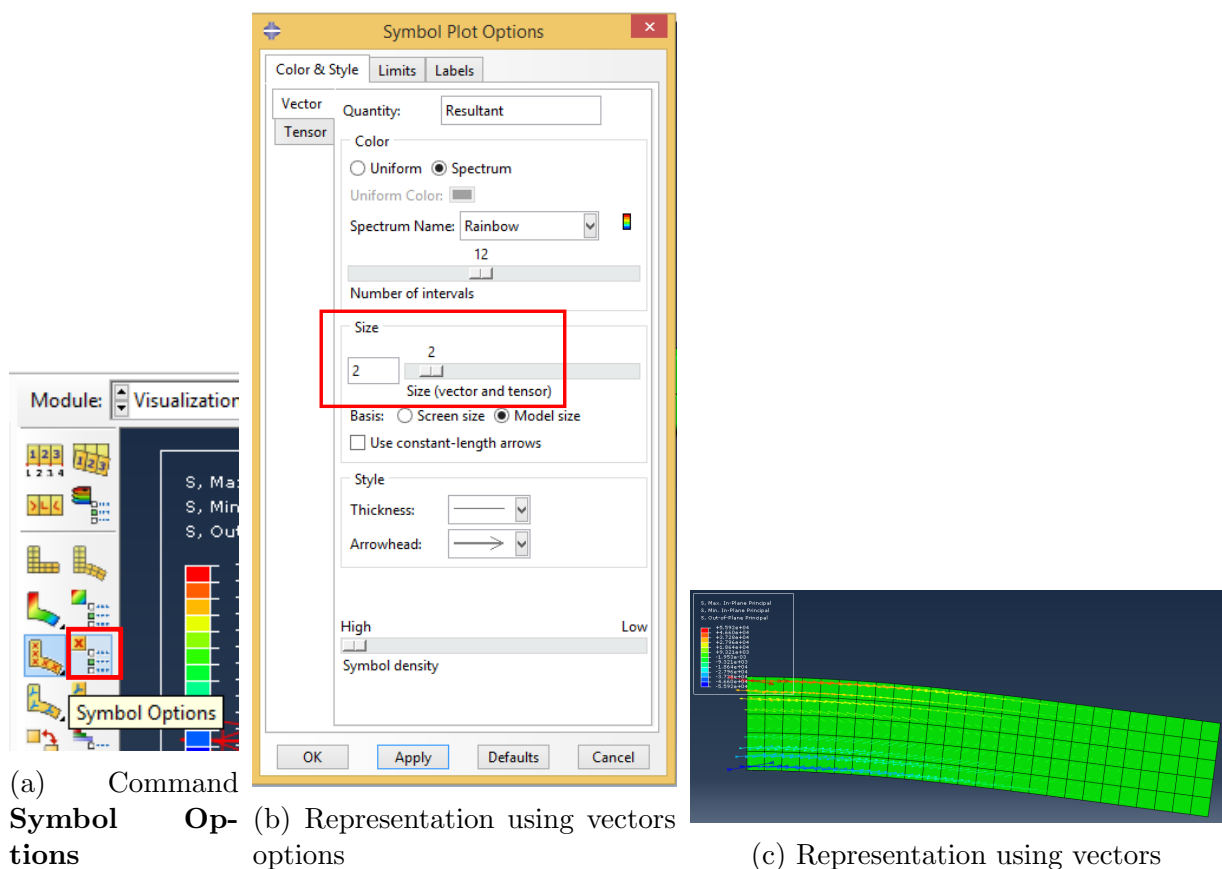


Figure 47: Distribution using vectors of a vectorial/tensorial magnitude (II)

4. Get values in nodes or elements

There are times when we are interested in knowing solution values in specific elements or nodes. Let's first obtain the solution in elements. To do so, we must activate first the **Plot Contour** with the field that we want to know. Draw, as in Fig. ??, the field of the σ_{xx} component of the stress tensor. Then press **Tools/-Query** (see Fig. ??) and select **Probe values** (see Fig. ??)

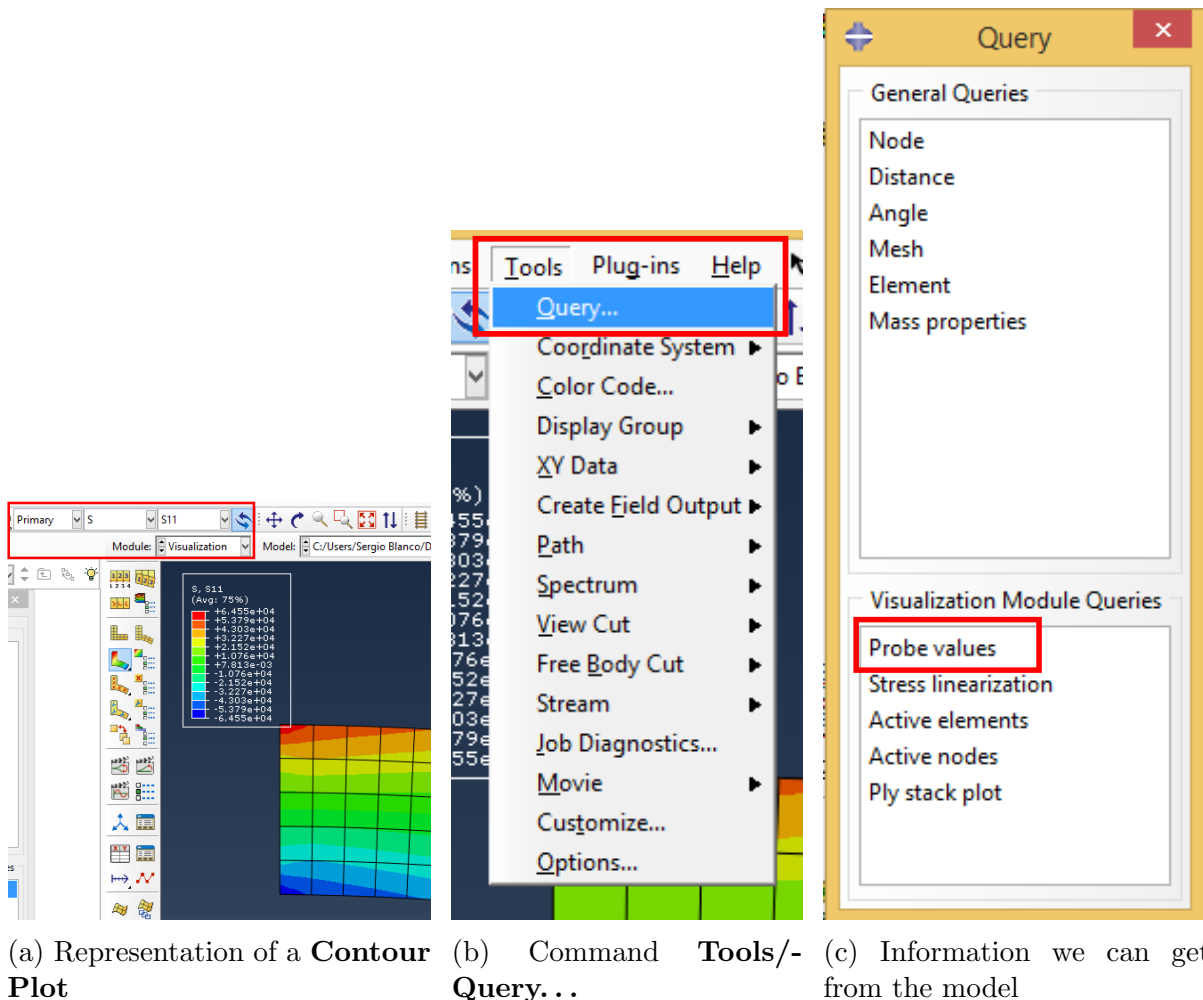


Figure 48: Obtaining values in elements (I)

A dialog box appears where we will indicate that we want to extract values from the **Elements** (which we select from the viewport), we want **All Directions** from the field selected and we want the values at the **Integration Points** as shown in Fig. ?. We select one by one the elements that are in contact with the left edge of the beam (see Fig. ?)

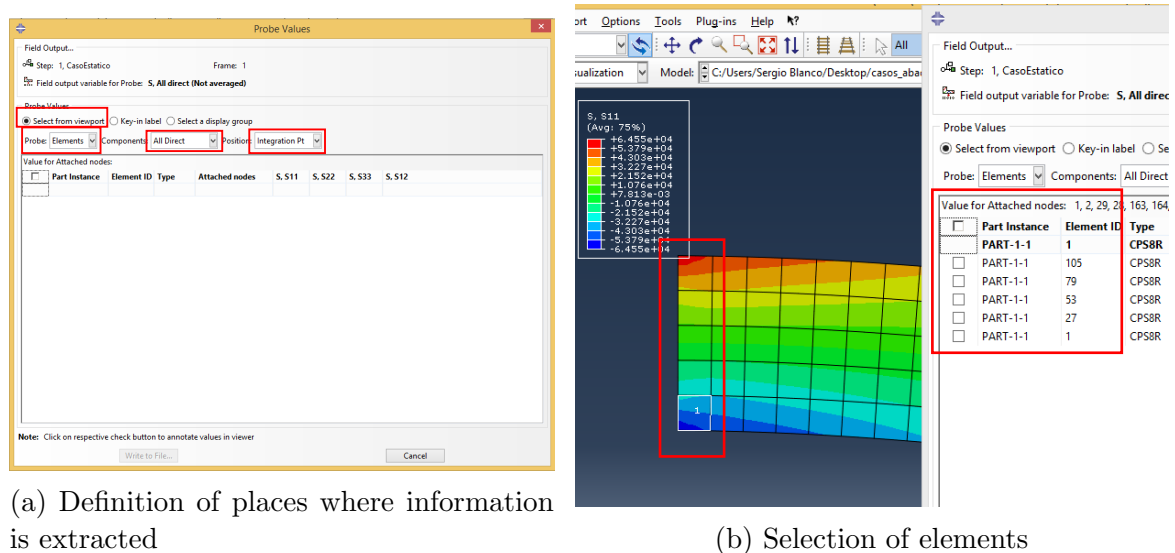


Figure 49: Obtaining values in elements (II)

Once we have selected the elements we press **Write to File** (see Fig. ??) and write the results in a file `tensiones.rpt` (see Fig. ??). If we open the file `tensiones.rpt` with a text program, we can see how we have stored the values of the components of the stress tensor in the four integration points of each of the selected elements (see Fig. ??).

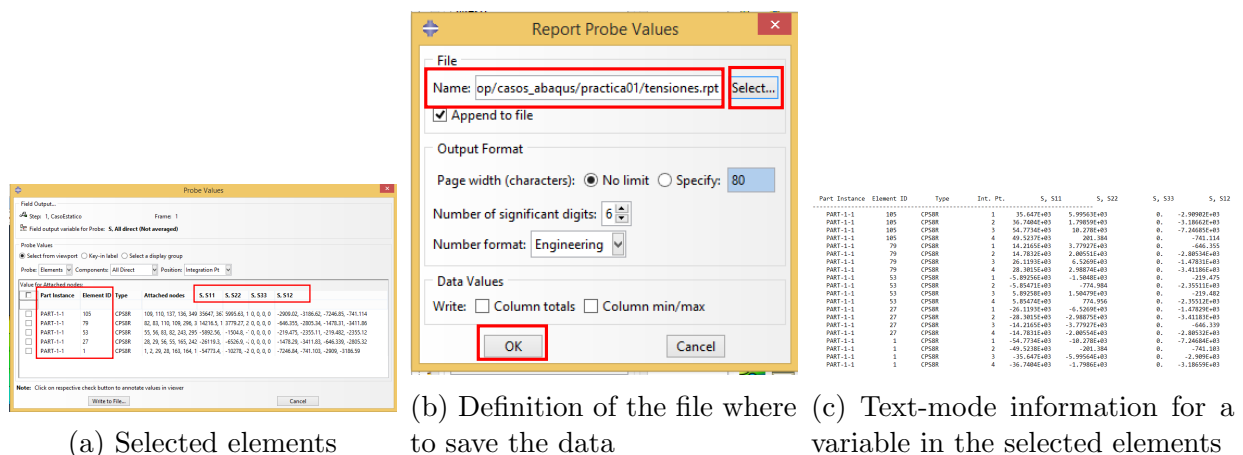


Figure 50: Obtaining values in elements (III)

We will now get the solution on nodes. Just as before we activated the **Plot Contour** with the field that we want to know about. In this case draw, as in Fig. ??, the field of the component R_x of the reaction forces. Remember that in order to activate the **Probe Values** dialog box, you must press **tools/Query** (see Fig. ??) and select **Probe values** (see Fig. ??). Now, set that we want to get the solution in **Nodes** and that we want all the components. Finally, select all the left edge nodes (see Fig. ??). Once selected click **Write to File** and save the data to a file called **Reactions**. Finally open the file and check that the values of the reactions have been stored in the selected nodes (see Fig. ??).

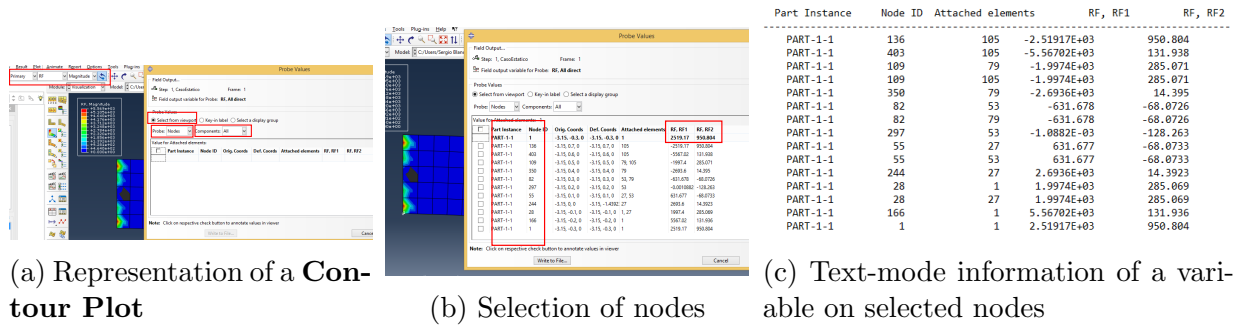


Figure 51: Obtaining values in nodes (I)

5. Get curves

Finally we will draw the distribution of a solution variable following a path in the geometry. To do so, we first have to define a **Path** inside the model and then obtain an X-Y curve in which the abscissa X is the distance of a point following the path and the ordinate Y is the value of the variable in that point.

Draw again the field of the R_x component of the reaction forces (see Fig. ??) and press **Tools/Path/Create** (see Fig. ??). In the following dialog box, assign the name *path-empotramiento* to the **path** and indicate that it is a **Node list** as shown in Fig. ??.

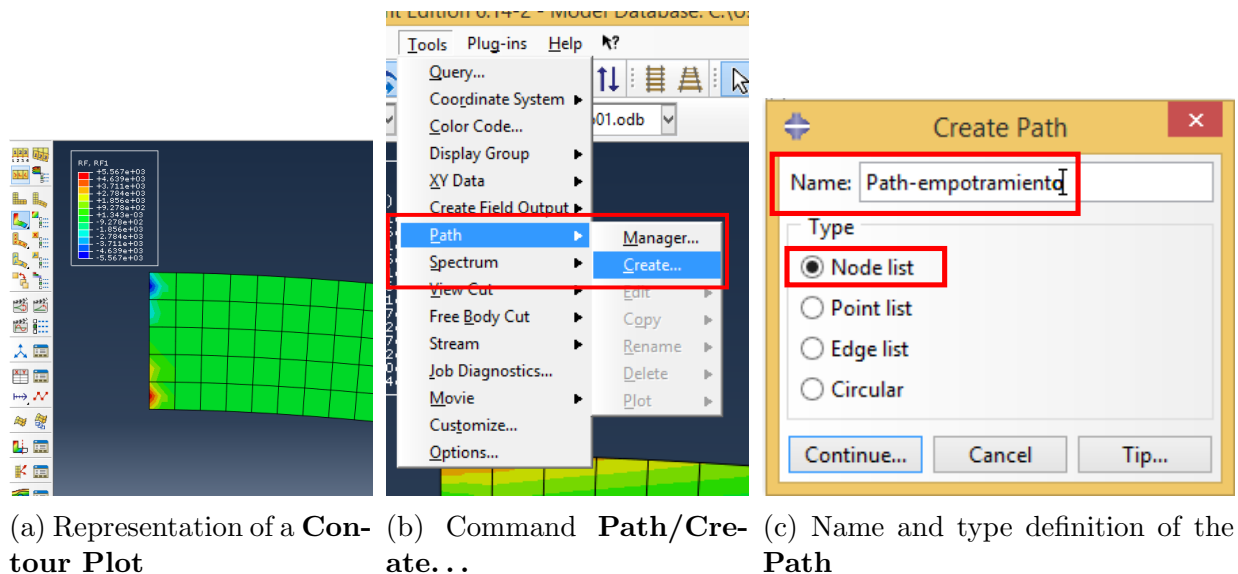
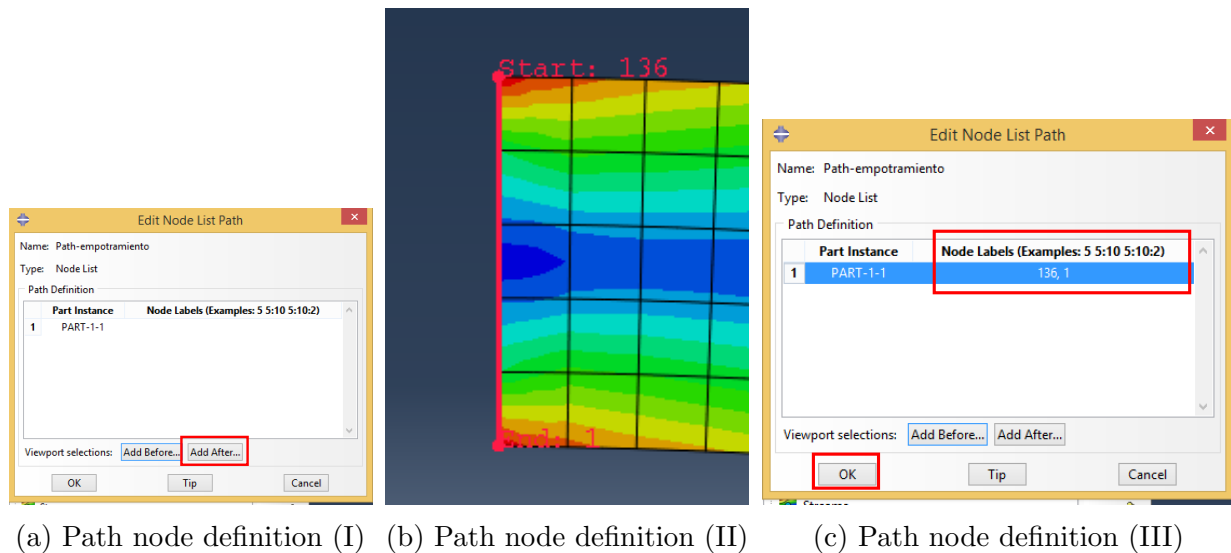


Figure 52: Definition of a Path (I)

In the next dialog, we must start assigning the nodes. Select **Add After** (although in this case it would be the same as **Add Before** because we do not have a previous selection) as indicated in Fig. ?. Mark the start point (upper end of the left side) and the end point (lower end of the left side) to define the path (see Fig. ?). Finally you should get the dialog box in Fig. ?.

Figure 53: Definition of a **Path** (II)

Once created the **path** we can define the X-Y graph. Press **Tools/XY Data/Create** as shown in Fig. ?? and set that the source for the abscissa will be a **Path** (see Fig. ??). In the next dialog, set that the path is the one we created previously *Path-embedded*, we will follow the deformed shape of the path (**Deformed**), we want to include the intersections between path and elements and that the value of the abscissa is the true distance following the path (check at Fig. ??). In order to select the variable we want to draw, press **Field Output variable** (see Fig. ??).

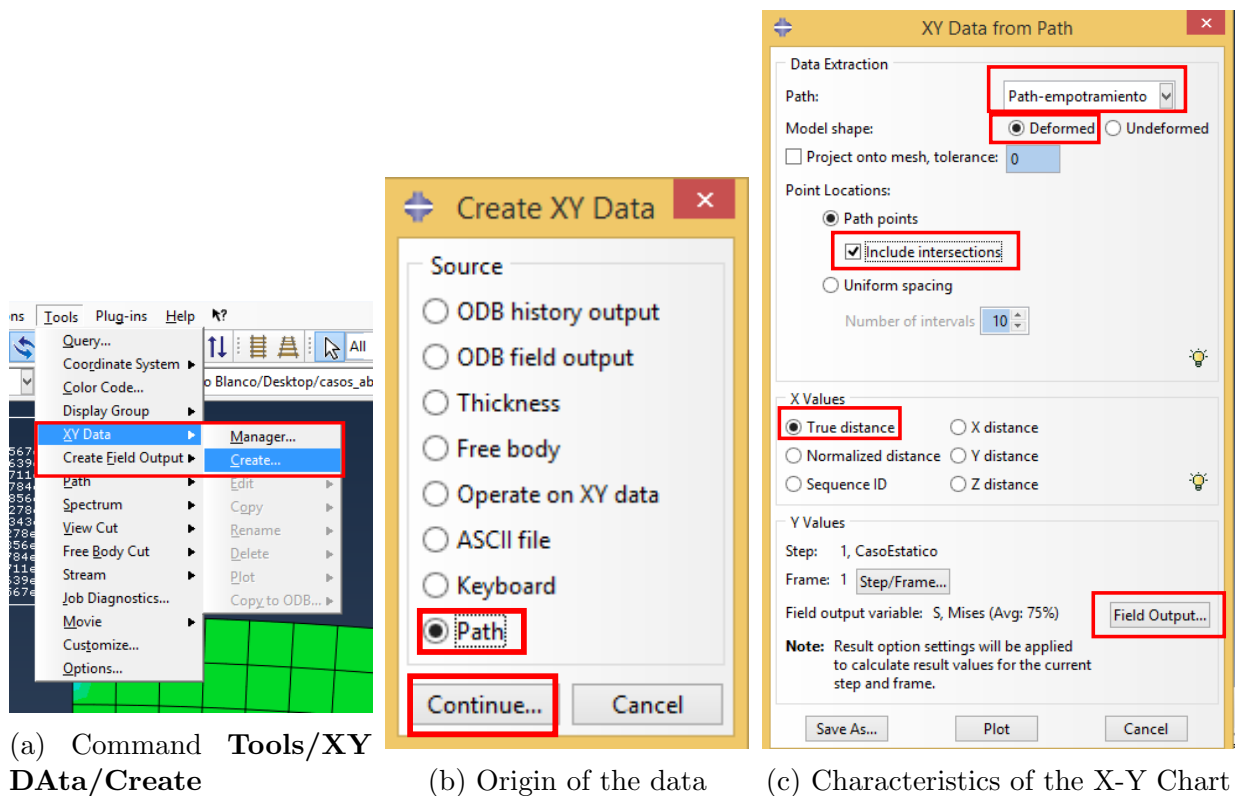
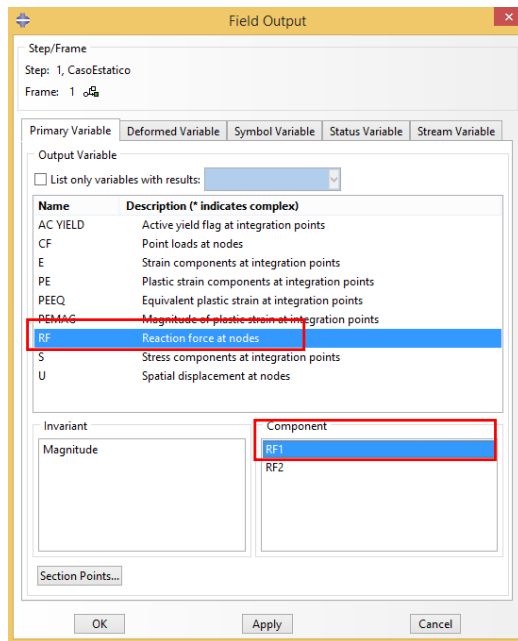
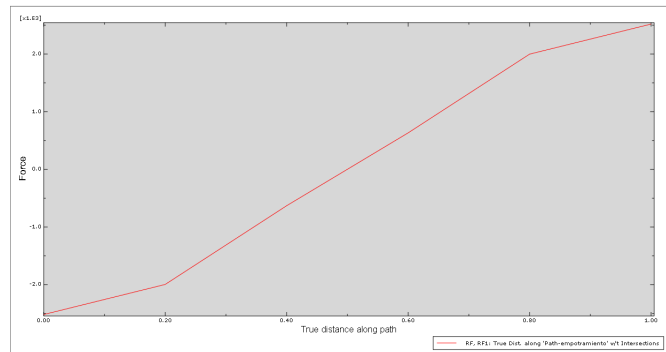


Figure 54: X-Y curve definition (I)

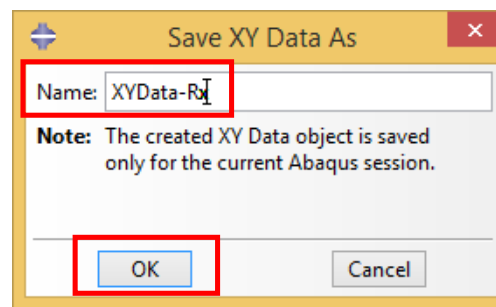
From the options provided by the following window (see Fig. ??) we choose the component **RF1** of the reaction forces at the nodes (**RF**). Again in the dialog box **XY Data from Path**, press **Plot** and we must obtain the curve shown in Fig. ?. Finally, we press **Save as...** and save the graph with the name *XYData-Rx* (see Fig. ??).



a: Selection of the variable to draw



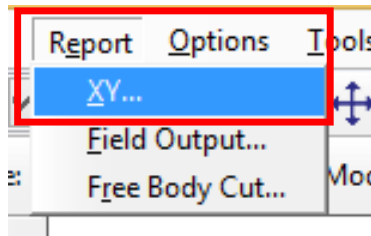
b: Rx values on the encastre side



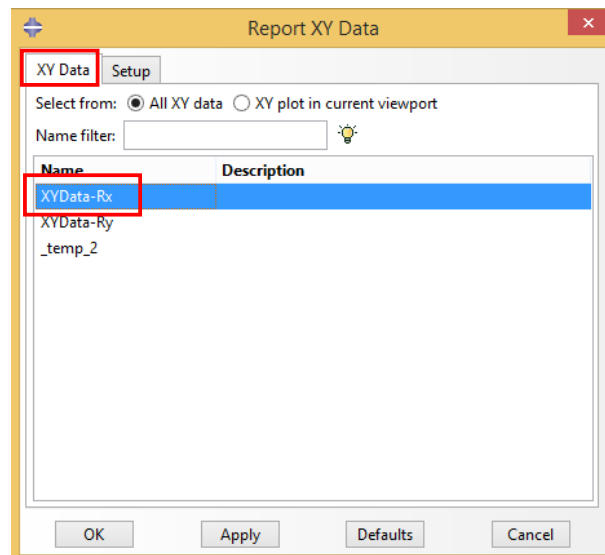
c: Name of the graph (is saved for the session only)

Figure 55: X-Y curve definition (II)

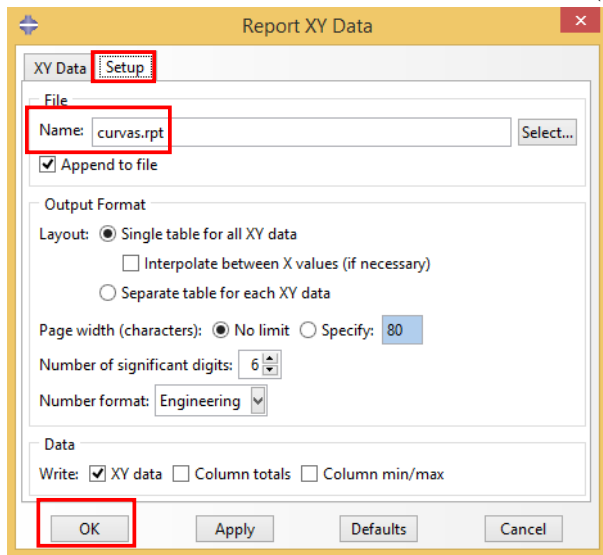
Finally we need to save the data of the curve we have created in a text file so it can be used by another program. Select **Report/XY...** (see Fig. ??) and, in the following window select the graph that we created *XYData-Rx* in the **XY Data** tab (See Fig. ??) and the filename **curvas.rpt** in the **Setup** tab (see Fig. ??). If you open the text file you should get something similar to the data in Fig. ??.

(a) Command **Report/XY**

...



(b) Writing of X-Y chart data (I)



(c) Writing of X-Y chart data (II)

X	XYData-Rx
0.	-2.51917E+03
200.E-03	-1.9974E+03
400.E-03	-631.678
600.E-03	631.677
800.E-03	1.9974E+03
1000.E-03	2.51917E+03

(d) X-Y chart data in plain text format

Figure 56: X-Y curve definition (III)

2 Proposed exercises

2.1 Exercise 1

In this exercise the classical Cook's Membrane is proposed to be modeled. Geometry is shown in Figure 57 (mm). The membrane is fixed (encastre) in its left side and a vertical load is distributed along the right side with value $F = 1.8 \text{ kN}$. The elastic properties of the material are $E = 70 \text{ GPa}$ and $\nu = 0.33$, and its thickness (direction orthogonal to the membrane) is 1 mm. Plane stress hypothesis is considered. We are going to use a structured mesh, subdividing each boundary of the membrane in 16 parts. The elements considered in this exercise are the same ones used in the main practical lesson.

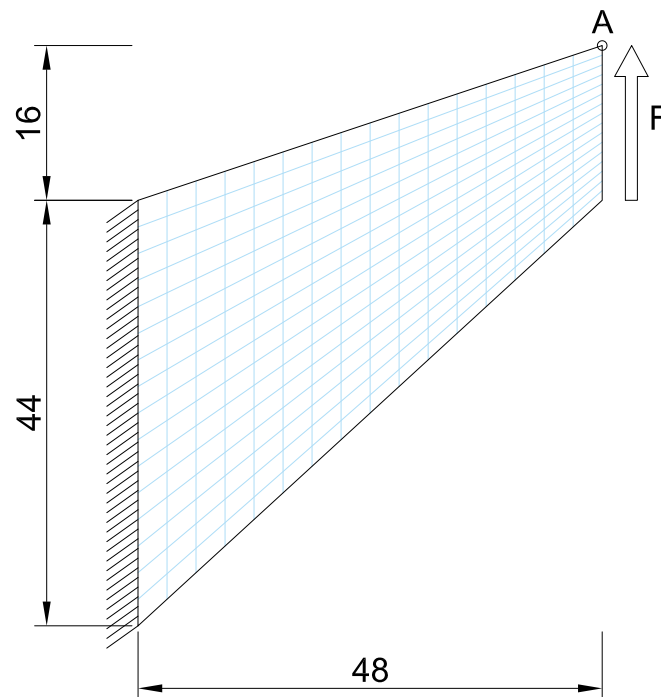


Figure 57: Model description

1. The displacement at point A is:

A: 0.807 mm

B: 1.125 mm

C: 0.689 mm

D: 1.333 mm

2. Maximum value of the reaction in the left side (encastre boundary) is:

A: 1203.0 N

B: 857.6 N

C: 560.3 N

D: 251.6 N

3. Maximum value of Von Mises's stress is:

A: 1321.0 MPa

B: 854.2 MPa

C: 1598.3 MPa

D: 381.6 MPa

2.2 Exercise 2

We want to study the mechanical behaviour of the model presented in Fig. ??, where the circular segments have a radius equal to 1 m. The piece has a state of *plane stress* with the following boundary conditions:

- All displacements are set to zero in segment AB .
- In segment EF we impose the value of the displacement $u_x^* = 0.01$ m.
- There is a concentrated vertical force of value $F_y = 200$ N.

The material behaviour is isotropic linear elastic with the elastic modulus $E = 1000000$ Pa. and $\nu = 0.25$. In order to build the discretized problem we use a mesh with the following parameters: *quad-dominated*, *quadratic* and with a global size of 0.9 .

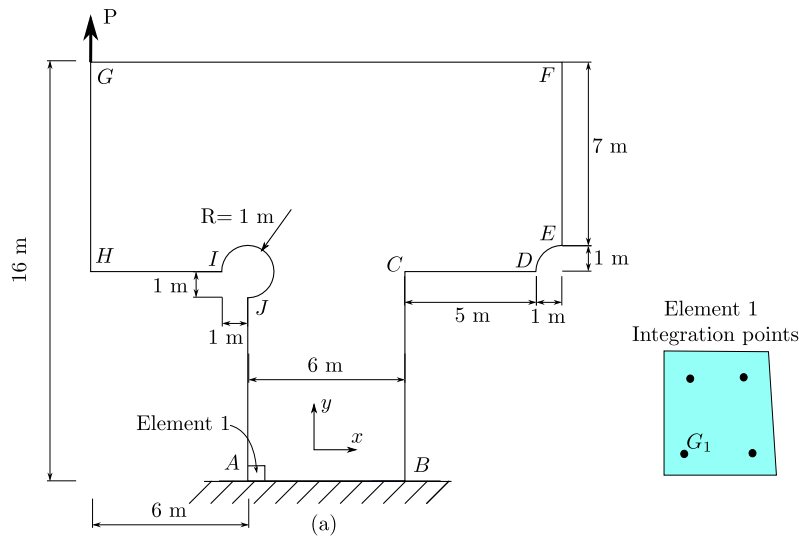


Figure 58: Model description

For this model answer the following questions (in this problem you must use some commands we have not explained in the previous example, try to figure it out with the *Help* assistance):

1. What is the vertical displacement of point H?

A: 0.0094 m.

B: 0.0025 m.

C: 0.0125 m.

D: 0.0003 m.

2. What is the horizontal displacement of point J?

A: 0.0012 m.

B: 0.0050 m.

C: 0.0006 m.

D: 0.0181 m.

3. What is the σ_{22} component of the stress tensor in the integration point G_1 of element 1 (see Figure)?

A: 710 Pa.

B: 1412 Pa.

C: 172 Pa.

D: 904 Pa.

4. What is the maximum value of the maximum principal stress in the path AF ?

A: 51 Pa.

B: 894 Pa.

C: 1384 Pa.

D: 323 Pa.

5. What is the sum of the horizontal reaction forces at the base AB ?

A: -231.75 N.

B: -124.87 N.

C: -13 N.

D: -323 N.

2.3 Exercise 3

We present here an example that formally is very similar to the exercise solved in the previous section. It is an L-shaped piece that is clamped in one of its ends and there is a force distributed in half of its upper edge. Fig. ?? summarizes the example.

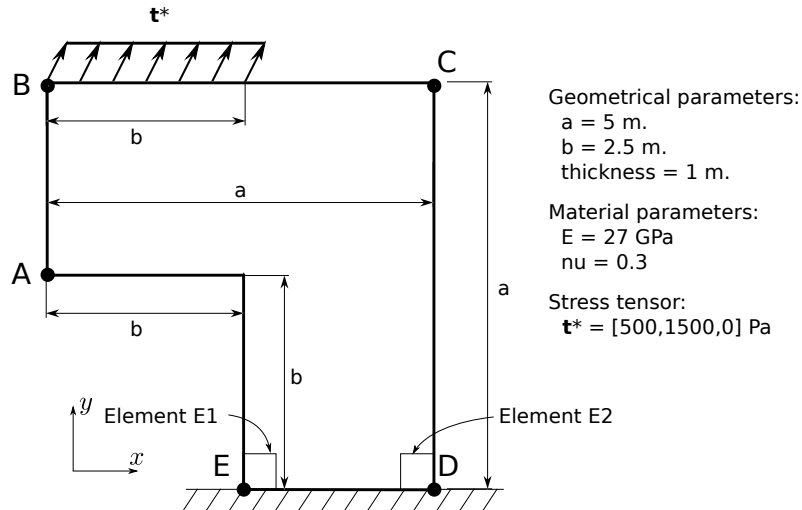


Figure 59: Model description

For this model, following the working schema of Section 1, using the same type of quadrilateral element and a value of element global size of 0.3 meters, answer the following questions:

1. What is the vertical displacement of point B?

A: $2.821 \cdot 10^{-6}$ m. B: $6.024 \cdot 10^{-6}$ m. C: $2.123 \cdot 10^{-5}$ m. D: $3.254 \cdot 10^{-9}$ m.

2. What is the horizontal displacement of point C?

A: $1.627 \cdot 10^{-6}$ m. B: $4.741 \cdot 10^{-6}$ m. C: $2.321 \cdot 10^{-5}$ m. D: $5.237 \cdot 10^{-9}$ m.

3. What is the σ_{22} component of the stress tensor in the centroid of the element E1?

A: 1782 Pa. B: 17413 Pa. C: 7174 Pa. D: 14904 Pa.

4. What is the maximum value (in absolute value) of the minimum principal stress σ_3 in the path EC?

A: -851 Pa. B: -8801 Pa. C: -1984 Pa. D: -3366 Pa.