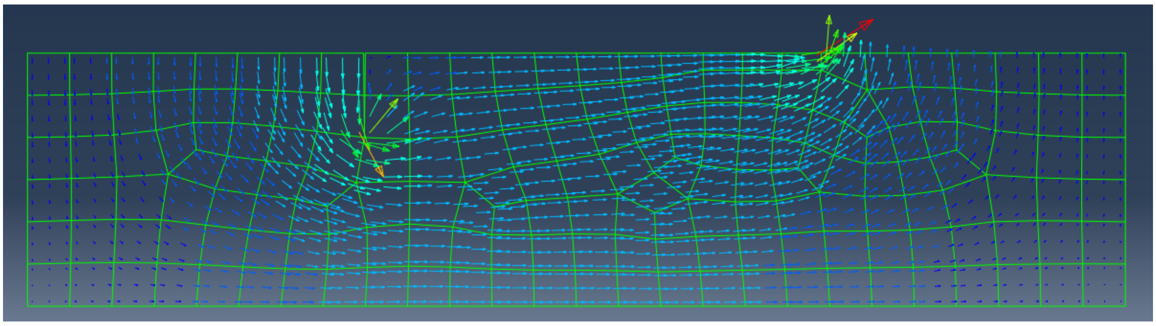
Universidad Politécnica de Madrid   Máster en Ing. de Caminos, Canales y Puertos

Computational Methods in Civil Engineering. Practical cases.

Chapter 2-3. Steady-state diffusion problems.

Grupo Mecánica Computacional



Introduction

The goals of this practice are:

1. Reinforce the practical concepts associated with (i) the numerical analysis by the finite element method and (ii) the use of the Abaqus program.

2. Review the problem of stationary diffusion, its numerical resolution and apply these concepts to the problem of fluid flow in a porous medium.

The practice consists of three sections:

Section 1. Brief introduction to the problem of steady-state fluid flow in a porous medium.

Section 2. Description of the finite element analysis procedure of a steady state diffusion problem with the Abaqus program.

Section 3. Four exercises to be solved by the student.

1 Introduction to the steady-state diffusion problem

1.1 The problem of steady-state flow in a porous medium

Next, as a review, we summarize the most important aspects of the steady-state flow problem in a porous medium of an incompressible fluid:

- The fluid flow in a porous medium takes place when there is an energy difference between two points of the fluid. The flow takes place from the point of higher energy (upstream) to the point of less energy (downstream).

- The energy level of an incompressible fluid at a point for a steady state is expressed by the Bernoulli's Equation, which calculates the total head of a fluid h at a point as the sum of three components: pressure head, head of elevation and velocity head:

$$h = h_p + h_e + h_v = p/\gamma_w + z + v^2/(2g) \approx p/\gamma_w + z \tag{1}$$

  being p the fluid pore pressure, γ_w its specific weight, z its elevation head measured with respect to a selected horizontal line (the points under this line will have a negative elevation head). The velicity head has been neglected considering the velocity of the fluid is negligible compared to the other terms.

- When the fluid passes from a point A with total head h_A to a point B with total head $h_B < h_A$, there is a loss of energy due to the friction to the flow offered by the soil. The hydraulic gradient i_h is defined as the total head loss per unit length. If the distance between A and B according to their current line is L_{AB} the mean hydraulic gradient between these two points is:

$$i_h = \frac{h_B - h_A}{L_{AB}} = \frac{(p_B/\gamma_w + z_B) - (p_A/\gamma_w + z_A)}{L_{AB}} \tag{2}$$

  And the hydraulic gradient at a point is:

$$\mathbf{i}_h = \boldsymbol{\nabla} h \tag{3}$$

- When the flow through the soil is laminar, Darcy's law is applied (equivalent to the Fick's law for the diffusion problem explained on page 5 on the presentation of the theory class by making $u = h$ and $\boldsymbol{C} = \boldsymbol{K}$):

$$\boldsymbol{q} = -\boldsymbol{K} \cdot \boldsymbol{\nabla} h \tag{4}$$

  Where $\boldsymbol{q}$ is the flow vector (its units are fluid volume per unit area and per unit time).

- The balance equation of the laminar (stationary) flow problem in a porous medium of an incompressible fluid is equivalent to the (stationary) Diffusion Equation described on page 9 of the presentation of the theory class.

1.2 Abaqus modeling strategy of the flow in a porous medium using the thermal problem

In Abaqus the problem of fluid flow in a porous medium is within a coupled formulation that solves:

- The mechanical problem of soil deformation (which can be considered rigid or deformable)
- The problem of fluid transport within the soil

Abaqus has also implemented the heat conduction problem, which formally has the same formulation as the problem of fluid flow in a porous medium (both problems are idealized with the equation in partial derivatives that we have called Diffusion Equation in theory class, see page 12). We can strategically use the Abaqus modulus of heat conduction to reproduce the fluid flow in a porous medium by making the following equivalence:

Parameters	Heat Conduction	Flow in porous medium
h	Temperature	Total (hydraulic) head
k	Conductivity coefficient	Permeability coefficient

Table 1: Equivalence thermal problem - flow problem in porous medium

Therefore, in this practice we will solve the problem of fluid transport in a porous medium using the Abaqus heat conduction module.

2 Worked exercise

We solve now the flow problem summarized in Fig. 1. We want to reproduce the water flow in an isotropic soil with a thickness of 9.2 meters, which is formed by silty sand with permeability coefficient ($k_x = k_y = k_z = 5 \cdot 10^{-5}$ m / s) and borders on its bottom edge with a layer of impermeable clay. In this sandy stratum a sheet pile of 4.6 m. height and an assumed infinite length (in the direction perpendicular to the plane of the drawing) has been nailed. To the left of the sheet piling (upstream) a height of 3 meters of water has accumulated and to the right (downstream) the runoff makes no accumulation of water. For the problem thus defined and assuming a stationary regime:

1. Obtain the outflow water flow downstream (per unit length in the y direction).
2. Obtain the evolution of the total head of the fluid in the BCD path and estimate the value of the hydraulic gradient at point D.

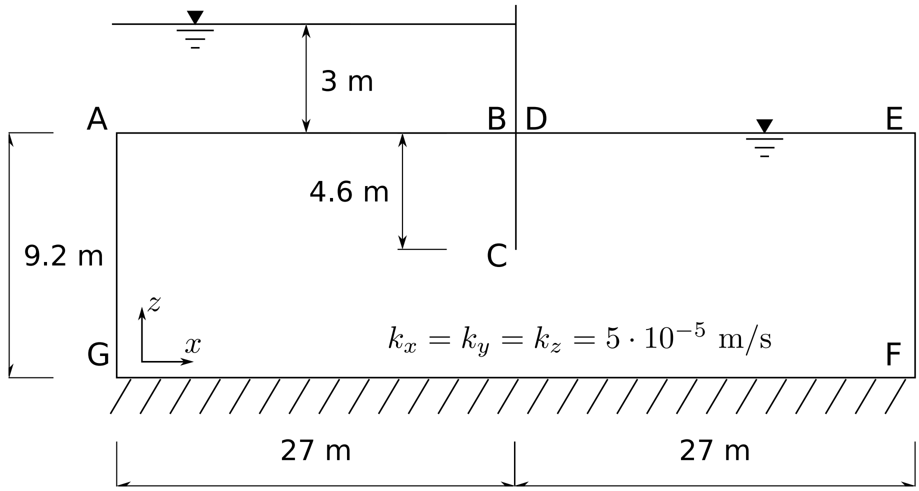


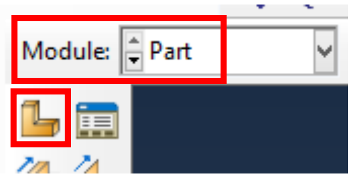
Figure 1: Model description

The units we are going to use are summarized in Table 2:

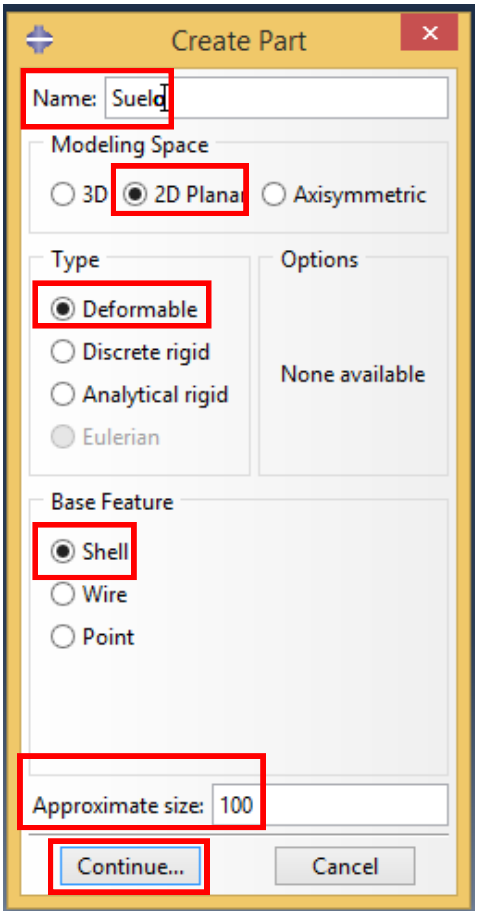
Magnitude	Units
Length	m
Total Fluid Head h	m
Coefficient of permeability k	m/s
Flow vector $\mathbf{q}$	$m^3/s/m^2$
Hydraulic gradient $\mathbf{i}_h$	m/m

Table 2: Units

To solve this problem we start Abaqus as in previous practice 1 and define a work directory called *Practica02*. In the remaining of this section we describe the needed actions to be made in each of the Abaqus modules to perform the analysis.



(a) Start **Part** module



(b) **Create Part** dialog box

Figure 2: Start **Part** module and **Create Part** dialog box

2.1 Module Part. Create the geometry of the elements

We activate the *Part* module (see Fig. 2a) and create a new object **2D Planar**, **Deformable** and **Shell** (see Fig. 2b).

With the *Sketcher* tool, we define the geometry of the problem as shown in Fig. 3a, finally obtaining the new *part* shown in Fig. 3b.

2.2 Module Property. Define materials and sections

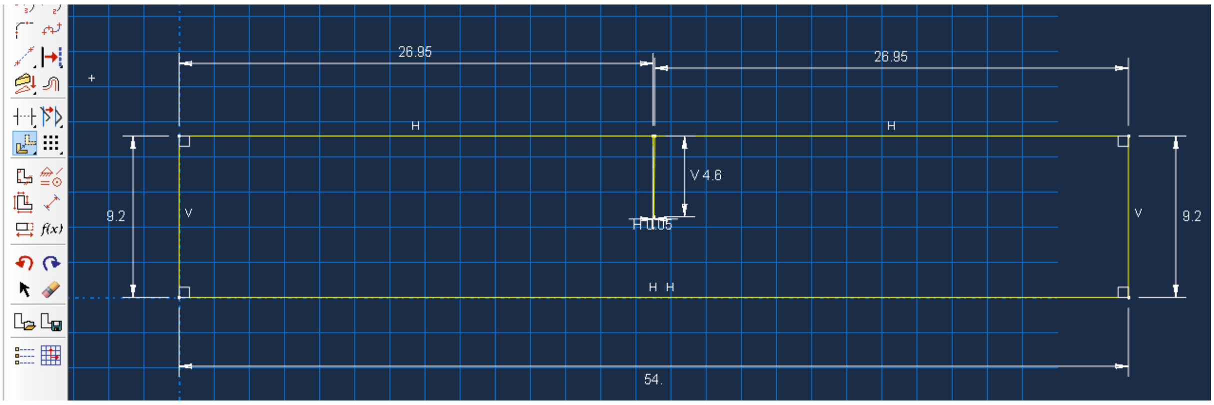
We now activate the **Property** module and define a new material **Thermal**, **Isotropic** and with a conductivity of $5 \cdot 10^{-5}$ as shown in Figs. 4a to 4c.

Once the material has been defined, we must create a new section of **Solid** and **Homogeneous** type following the steps of Figs. 5a to 5c.

Finally we assign the created section to the **part** as summarized in Figs. 6a to 6c.

2.3 Module Assembly. Assemble the model

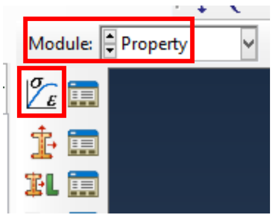
In the *Assembly* module we assemble our model by making a dependent copy of our part as shown in Figs. 7a and 7b.



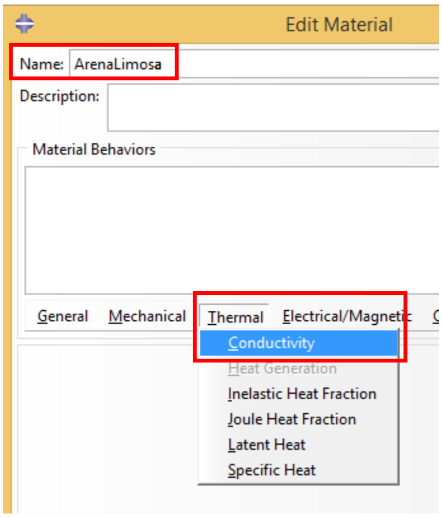
(a) Definition of geometry

(b) New **Part**

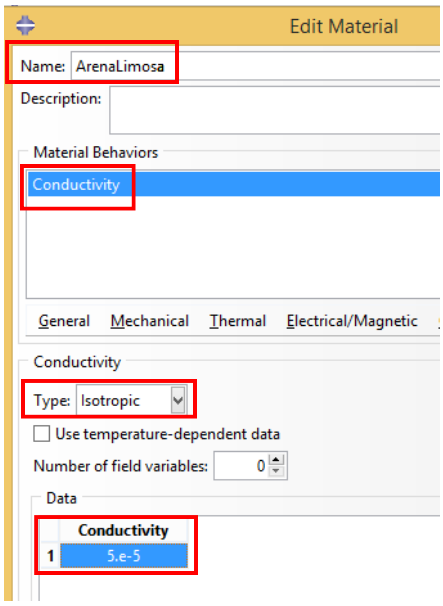
Figure 3: Construction of a new **Part**



(a) Command **Create Material**



(b) Behaviour selection



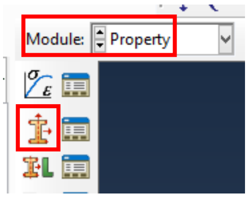
(c) Material parameter definition

Figure 4: Definition of a new material

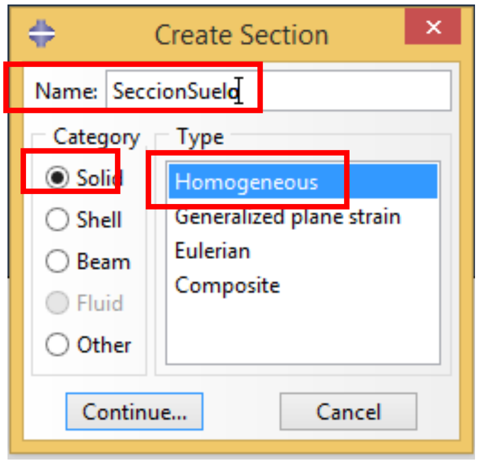
2.4 Module Step. Configure the analysis procedure

We need to create a calculation step (**Heat Transfer** type) with the option **Steady-state** (see Figs. 8a to 8c) .

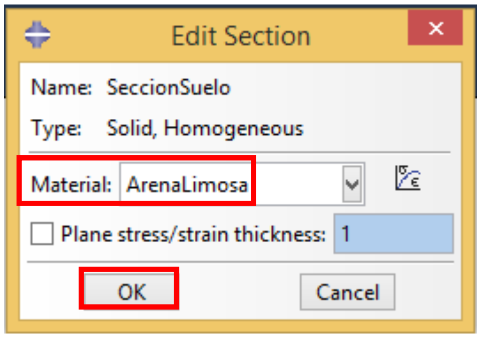
As we have defined a steady state regime, Abaqus does not generate a **History Output**. Let's review the data we are going to save for the post-processing by opening the **Field Output** that Abaqus creates by default, and making sure that we store the **Nodal temperatures**, the **Heat flux vector** and the **Reaction fluxes** (fluxes calculated where



(a) Command **Create Section**

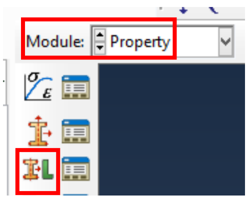


(b) Defining the properties of the section

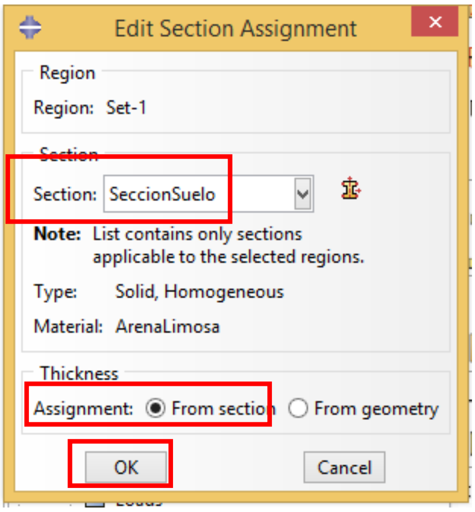


(c) Allocation of the material to the section

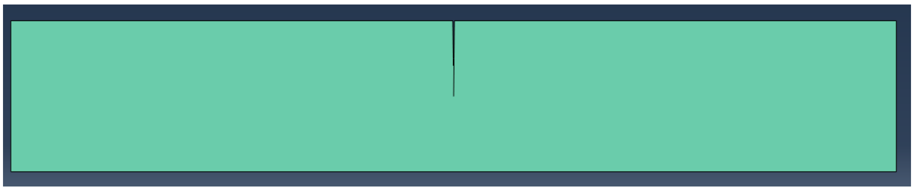
Figure 5: Definition of section *SeccionSuelo*



(a) Command **Assign Section**



(b) Linking the section to a *Part*



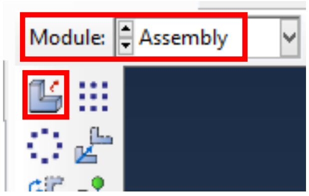
(c) *Part* with assigned section

Figure 6: Assigning the section to a **Part**

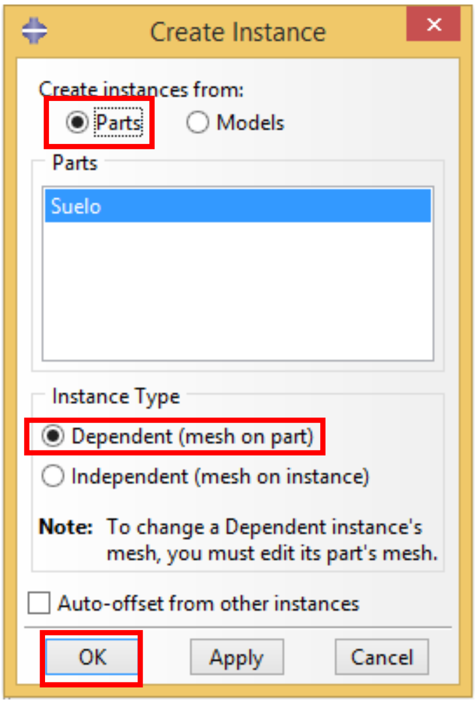
we apply a temperature Dirchlet-type boundary condition) as indicated in Figs. 9a and 9b.

2.5 Module Load. Apply the boundary conditions

In the case of confined flow problems we can find the following two boundary conditions (in our problem we will impose a constant value to the boundary conditions because we assume we are in a steady state):



(a) Command **Create Instance**



(b) Creation of a **Dependent** instance

Figure 7: **Create Instance**

Essential-type Boundaries where we impose the value of the total head of the fluid $h = h^*$.

Natural-type Boundaries where we impose the value of the flux $(-\mathbf{k} \cdot \boldsymbol{\nabla} h) \cdot \mathbf{n} = \mathrm{cte}$, being $\mathbf{n}$ the normal outside the boundary where we impose the value of the flow.

If we impose nothing on an boundary of our domain, Abaqus understands that we are imposing a null flow across that border (impermeable border), which means that we are imposing $\boldsymbol{\nabla} h \cdot \mathbf{n} = 0$.

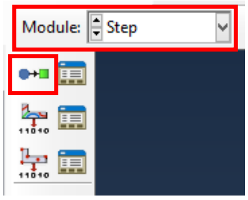
In our case we have to impose:

1. Two essential boundary conditions at the top of the domain (assuming the datum is on the horizontal line AE).

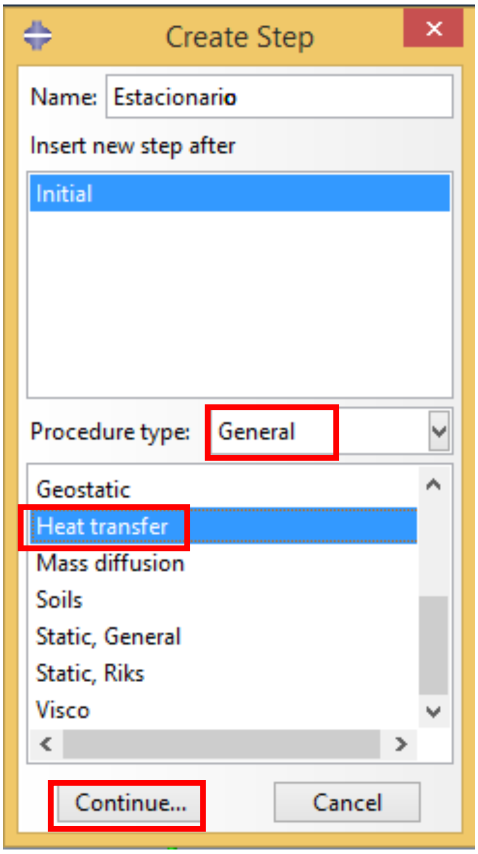
   - $h = u_A/\gamma_w + z_A = \dfrac{\rho_w\ g\ 3}{\rho_w g} + 0 = 3$ m in the equipotential AB.
   - $h = u_E/\gamma_w + z_E = \dfrac{\rho_w\ g\ 0}{\rho_w g} + 0 = 0$ m in the equipotential DE

   We will impose these conditions in the created step called *estacionario*.

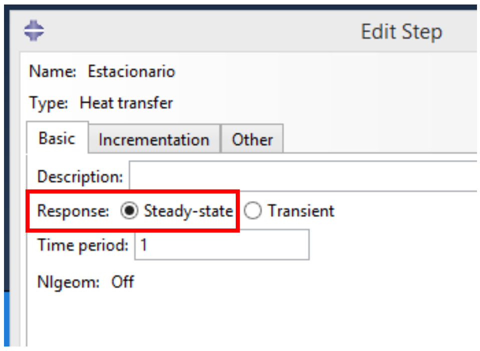
2. The natural-type boundary condition "impermeable boundary" $\boldsymbol{\nabla} h \cdot \mathbf{n} = 0$ in the AGFE and BCD boundaries. This boundary condition is kept fixed throughout the analysis and we could define it in the step *Initial* created by Abaqus to be propagated to the rest of steps. However, since this condition (impermeable border) is defined by default, it is not necessary for us to define anything.



(a) Command **Create Step**

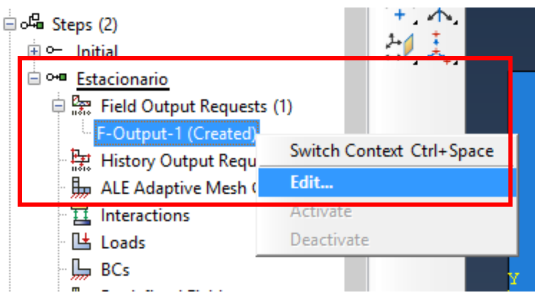


(b) Properties of new step

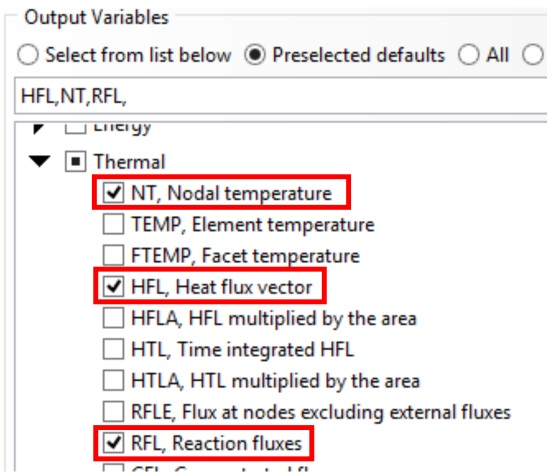


(c) Steady-state selection

Figure 8: Creating a new calculation step



(a) **Field Output** selection

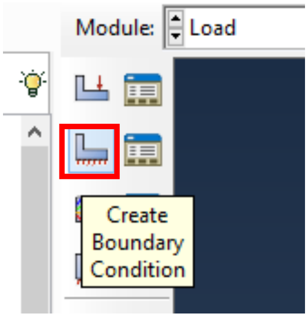


(b) Post-process values to be saved

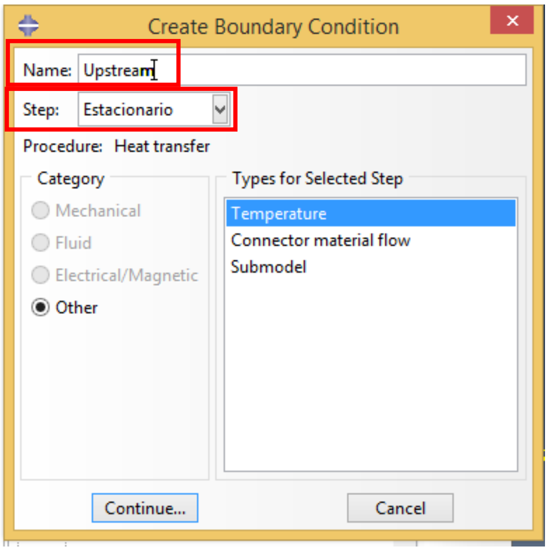
Figure 9: *Field Output* definition

To impose the essential boundary condition $h = 3$ in the boundary AB, we activate the **Load** module and press the **Create Boundary Condition** command (see Fig. 10a). Then follow the instructions given in Figs. 10b to 10d.

Likewise, to impose the essential-type boundary conditions $h = 0$ in the boundary DE follows the instructions given in Figs. 11a to 11c. At the end Abaqus indicates the boundaries where you have imposed boundary conditions as shown in Fig. 11d.

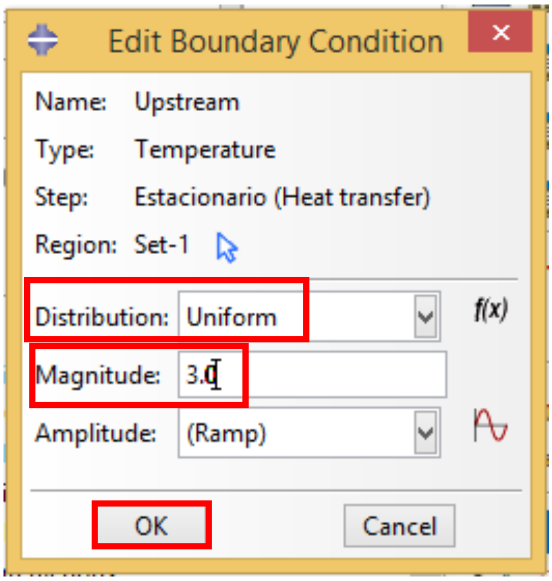


(a) Command **Create Boundary Condition**



(b) Definition of essential-type boundary condition

(c) Boundary selection



(d) Definition of imposed total head

Figure 10: Definition of the boundary condition in the equipotential AB

2.6 Module Mesh. Create the mesh.

In order to build the mesh, let's remember the actions we studied in previous practice 1:

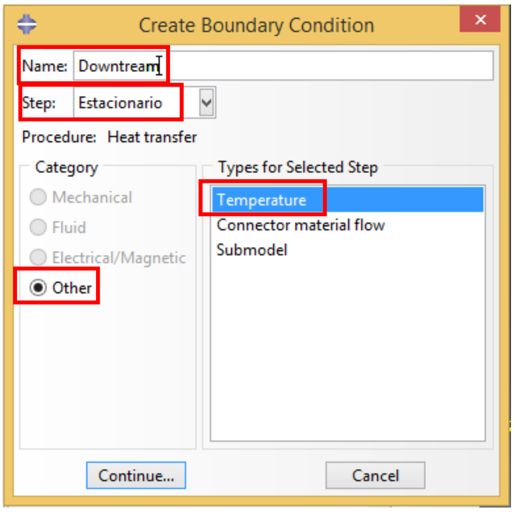
1. Activate the **Mesh** module and, since we have assembled the model with a dependent copy, we impose that we will mesh the **part** as indicated in Fig. 12.

2. Define the element shape (quadrilaterals) (see Fig. 13).

3. Set the global size of the element to 1.5 meters (see Fig. 14)

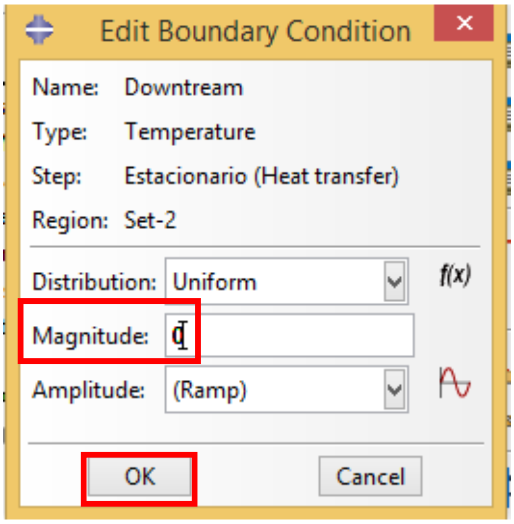
4. Finally we define the type of interpolation as indicated in Fig. 15.

Final mesh is shown in Fig. 16.



(a) Definition of essential-type boundary condition

(b) Boundary selection



(c) Definition of imposed total head

(d) Definition of imposed total head

Figure 11: Definition of the boundary condition in the equipotential DE

Figure 12: Start **mesh** module

2.7 Module Job. Create the job and run the analysis.

Once we have defined the model we only need to create and launch the **job**. To do this activate the **job** module, click on the **Create job** icon (see Fig. 17a) and follow the instructions in Figs. 17b and 17c.

Finally launch the job and, once the numerical analysis converges, activate the post process (see Figs. 18a and 18b).

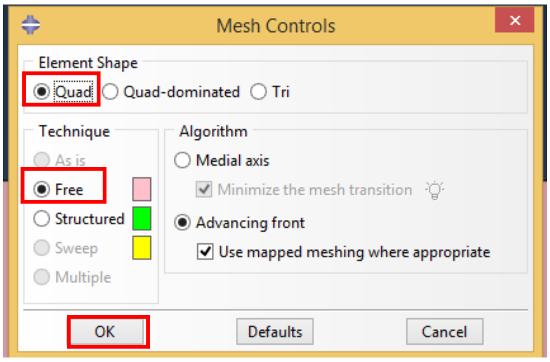


Figure 13: Shape element definition

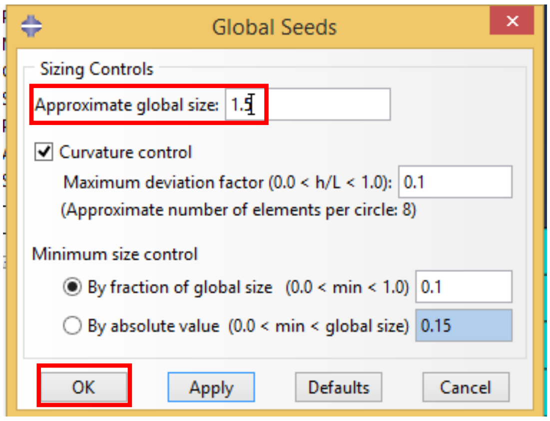


Figure 14: Element size definition

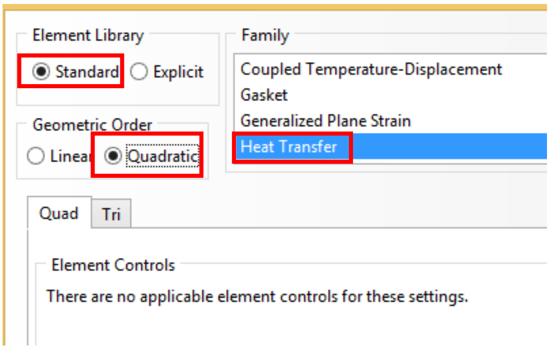


Figure 15: Type of element definition

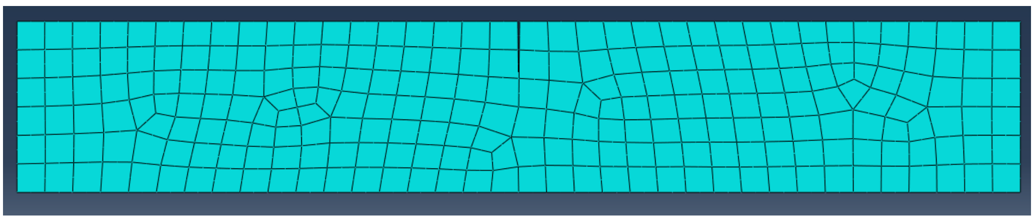
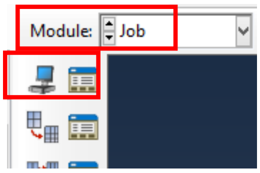
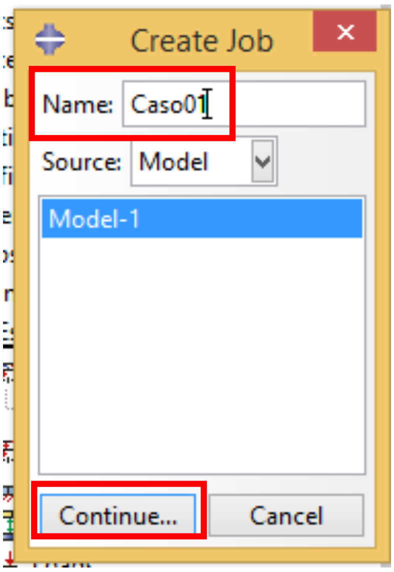


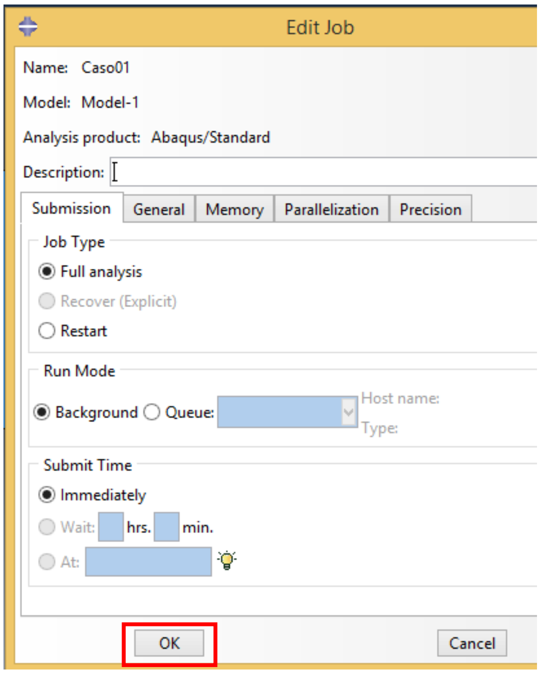
Figure 16: Final mesh



(a) Command **Create Job**

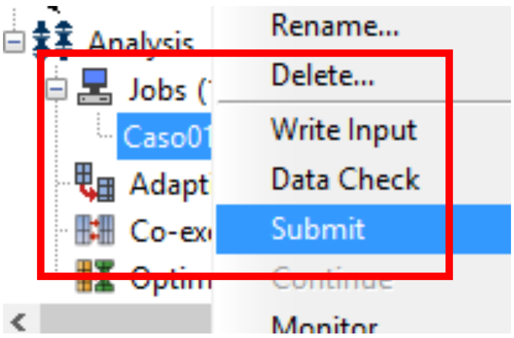


(b) *Create Job* dialog box

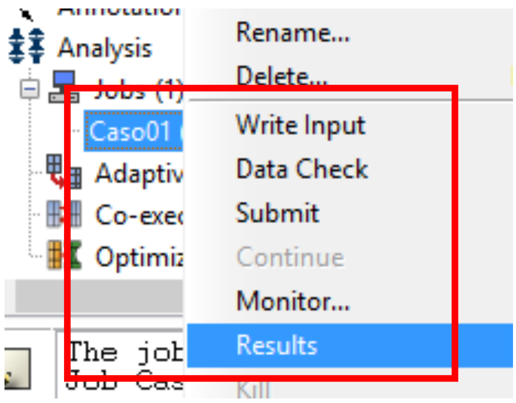


(c) Parameters of the analysis

Figure 17: Job definition



(a) Job submission



(b) Post-process activation

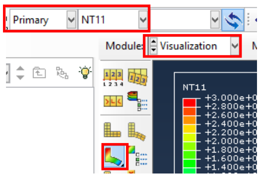
Figure 18: Job launch

2.8 Module Visualization. Post-process.

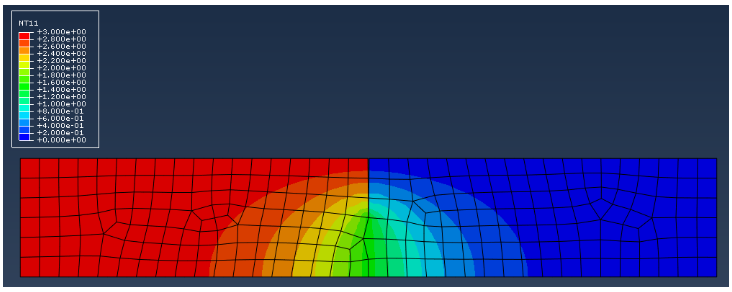
Next, the needed post process actions are summarized in order to answer the questions of the exercise:

- Let us first draw the field of the total head of the fluid at each point (see Figs. 19a and 19b). If we would like to obtain the equipotential lines, we activate the **Contour Plot Options** command and set that the type of contour is **line** as indicated by Figs. 19c and 19d.

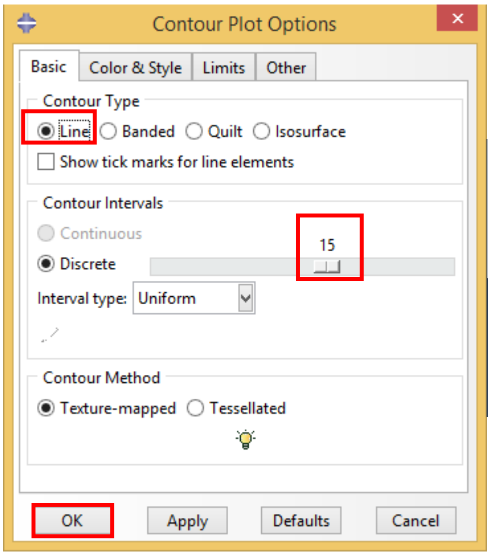
- We want to answer now the first of the exercise questions, to obtain the outflow downstream. To do so we must obtain the distribution of the outflow in the line DE and integrate it. Let us first define a **path** in the contour DE that we will call *CorrienteAbajo* as shown in Figs. 20a to 20e.



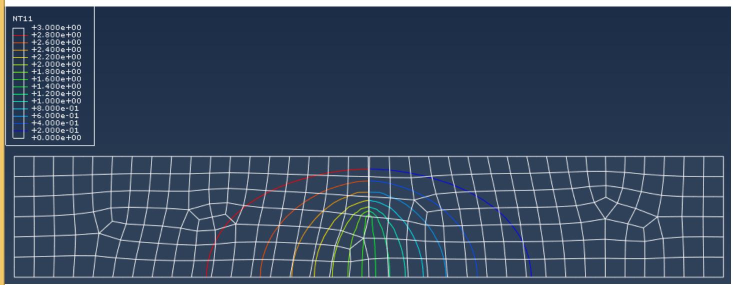
(a) Command **Plot Contours on deformed shape**



(b) Distribution of total fluid head



(c) **Contour Plots** dialog box



(d) Distribution of total fluid head (equipotentials)

Figure 19: Display of total fluid head

- Once the *path* is defined, we look for the distribution of the total head along it. To do so we must create an object **XY Data** as summarized in Figs. 21a to 21e.

- The distribution of the outflow in the DE boundary is shown in Fig. 22a. In order to export it to an excel sheet (and integrate it) we must save it as shown in Figs. 22b and 22c.

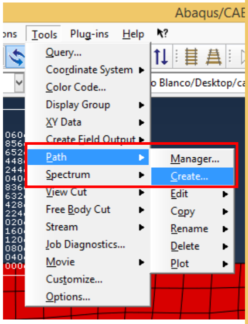
- Next we export the data from the XY Data object we just created as indicated by Figs. 23a to 23c.

- In Figs. 24a to 24d how to import the text file we just saved in Excel is explained. This information is valid for Excel2013.

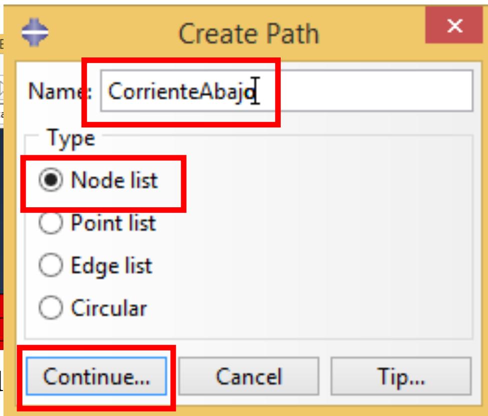
- With the imported data, we integrate the curve flow *vs.* distance using the trapezoid method, finally obtaining a value of the outflow of $7.67 \cdot 10^{-5}$ m^3/s per meter in the y direction (see Fig. 25).

  Equivalently (and faster) we can integrate the curve of Fig. 22a using Abaqus's `python` console. The XY-Data object we have named XYData-HFL2 (see Fig. 22c) is an object that Abaqus stores and can be accessed and operated with, as shown in Fig. 26 (try as an exercise to understand the logic of the python code).

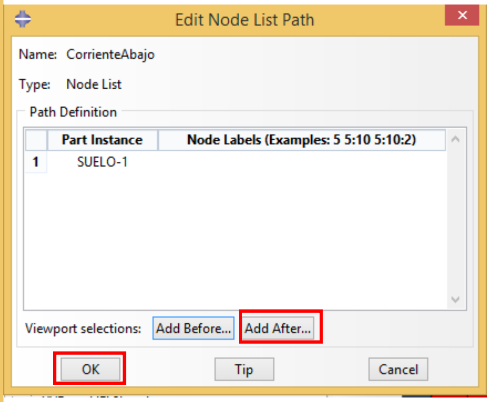
  Finally, there is a third way to integrate this curve using the Abaqus commands themselves, which allows us to operate with previously created *XYData* objects.



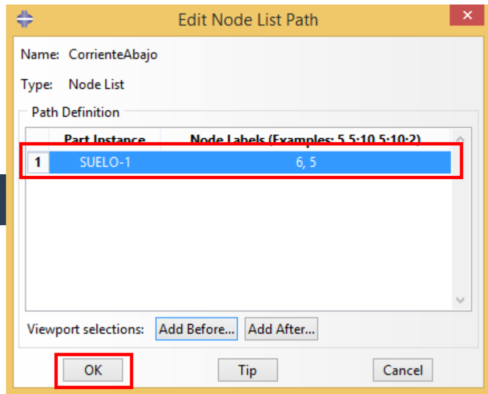
(a) Command **Path/Create**



(b) *Create Path* dialog box



(c) *Path* geometrical definition



(e) *path* geometrical definition

(d) Downstream boundary

Figure 20: *AguasAbajo* path definition

Press *Tools/XYData/Create*, but instead of creating a new object according to a *path* we will choose the option *Operate on XYData*, as shown in Fig. 27.

From all possible operations we select the command *integrate(X)* taking as argument the object *XYData* where we have saved the normal flow to the surface as summarized in Fig. 28.

Finally we can draw the value of the flow and its integration along the curve as shown in Fig. 29 and access the value of the total flow for the final ordinate as shown in Fig. **??**.

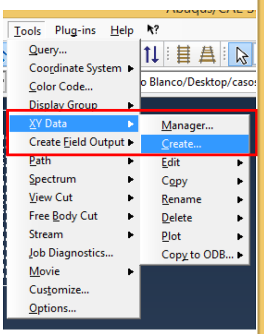
- In order to answer the second question we need to obtain the evolution of the total head of the fluid in the BCD path and to estimate the value of the hydraulic gradient in the point D. Therefore we must obtain the distribution of the total head in the BCD path and estimate its slope at point D. Similar to the previous point, first define a **path** following the BCD path as summarized in Figs. 31a to 31d.

- Once the path *PathBCD* is defined, we must create the XY-Data object with the distribution of the total head along the BCD path as summarized in Figs. 32a to 32d. Fig. 32d is the requested distribution.

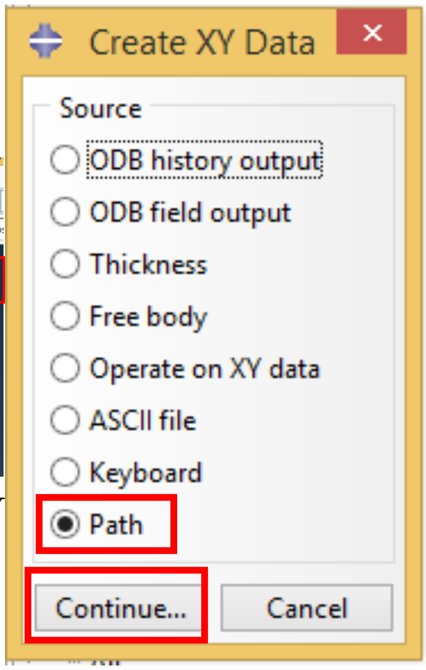
- We could estimate the hydraulic gradient in Fig. 32d but if we wanted to calculate it accurately we must save the data into an XY-Data object and then export it to a text file as summarized in Figs. 33a to 33c. From Fig. 33c we can calculate the requested gradient as:

$$i_h = \frac{0.3234 - 0}{9.2 - 7.67} = 0.211 \tag{5}$$

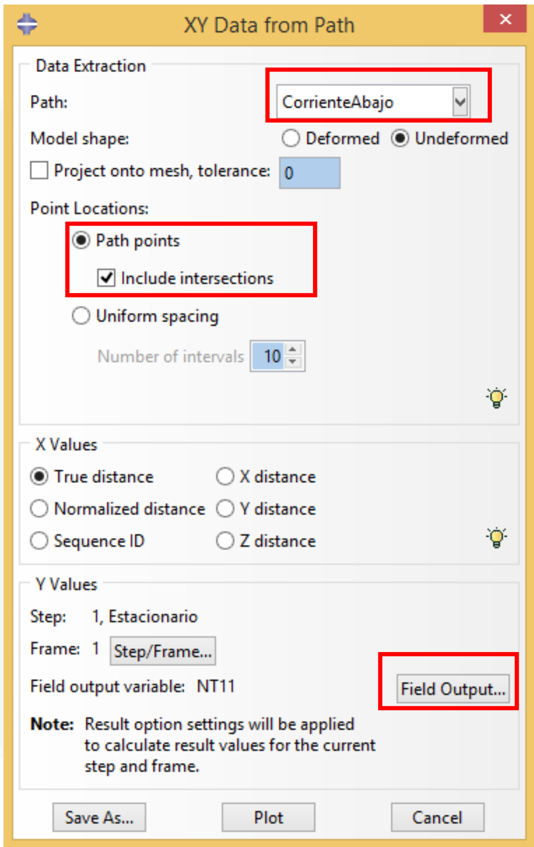
In this case it would have been easier to request the information about the created XY-Data object (in the **Model Tree**), see Figs.34a and 34b) instead of saving it to disk as a file text.



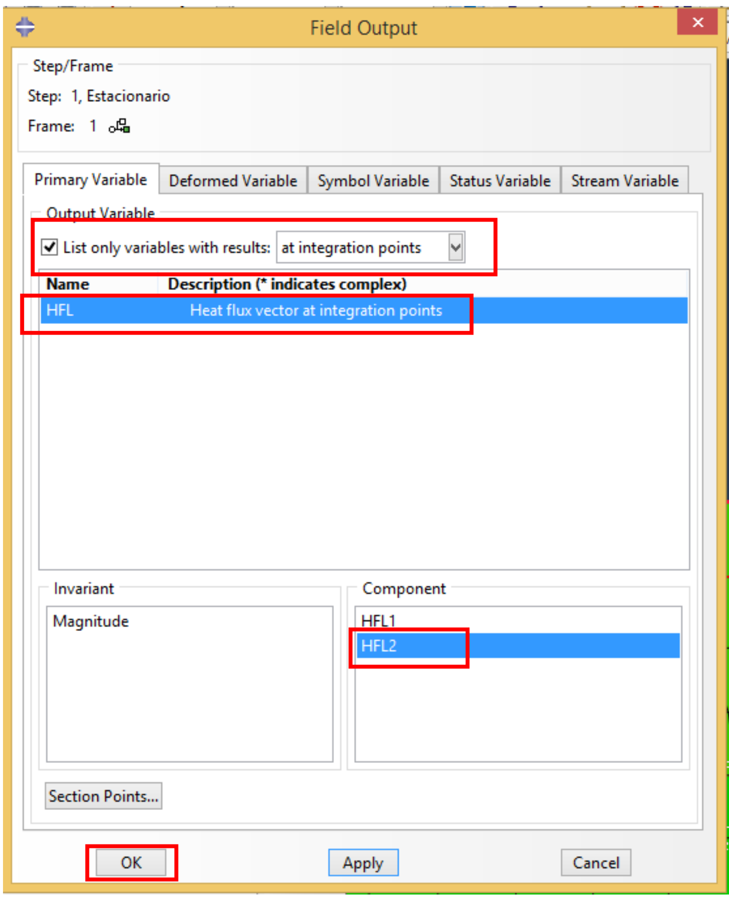
(a) Command **XY Data/Create**



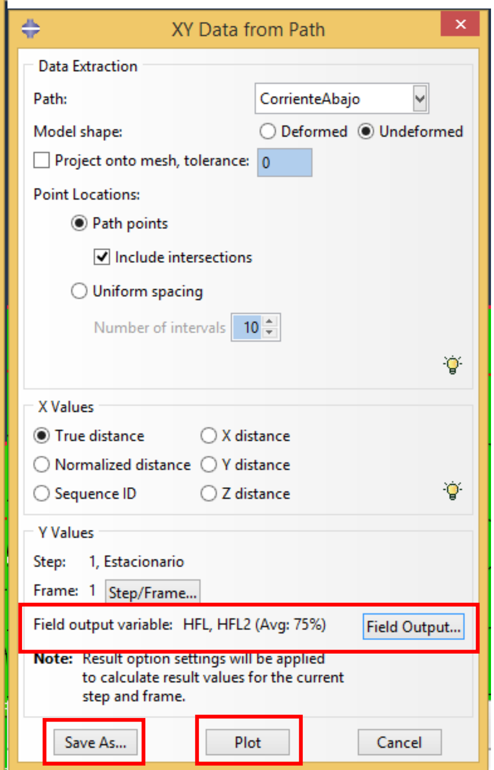
(b) *XY Data* dialog box



(c) *XY Data from path* dialog box

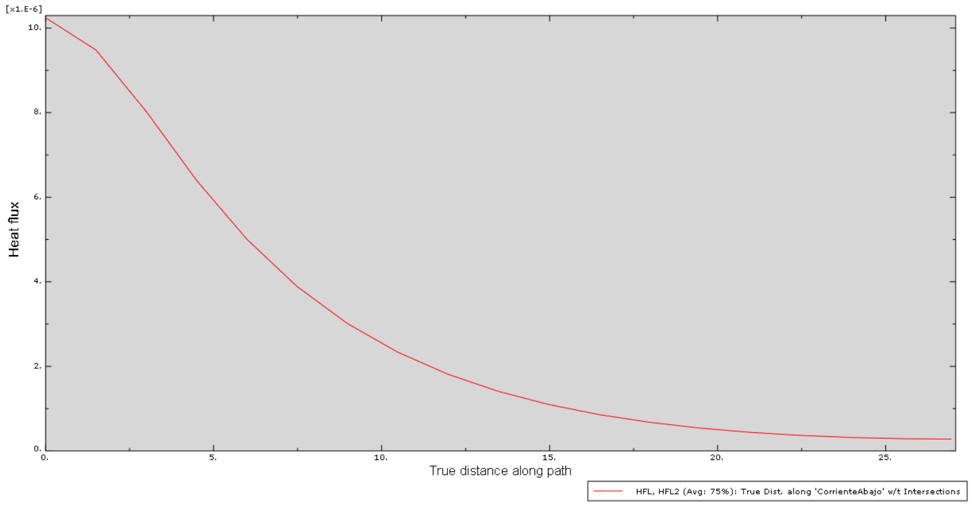


(d) Selecting the variable to draw in the path

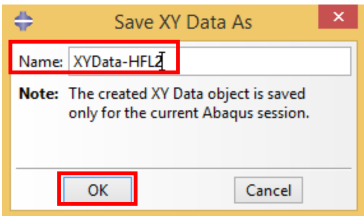


(e) *XY Data from path* dialog box

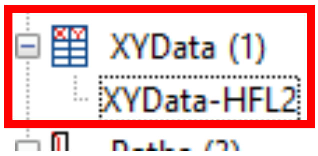
Figure 21: Definition object XY Data with the total head on the boundary DE



(a) Distribution of the outflow in straight line DE

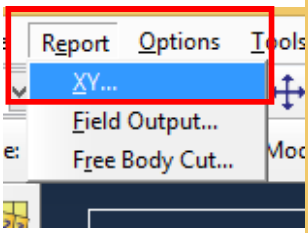


(b) *Save XY Data As* dialog box

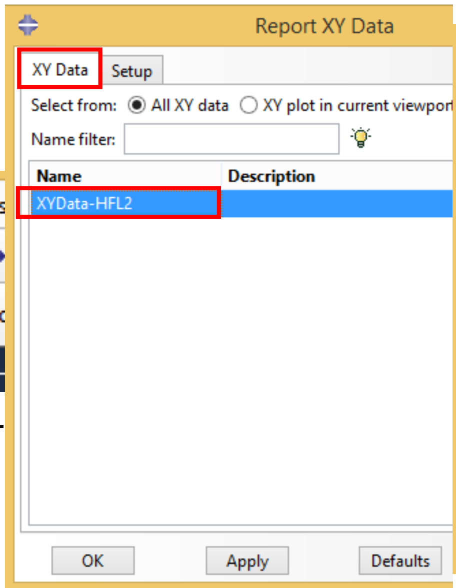


(c) XY Data objects in the *Model Tree*

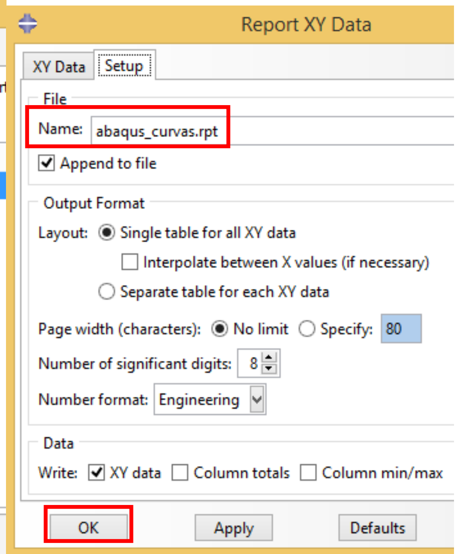
Figure 22: XY Data objects of the outflow in the line DE



(a) Command **Report/XY**

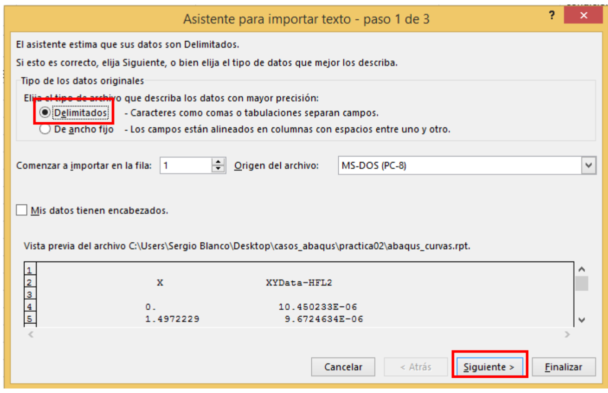


(b) *report XY Data* dialog box

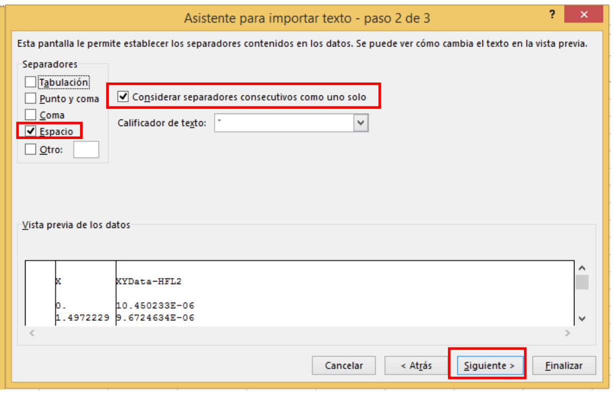


(c) *Report XY Data* dialog box

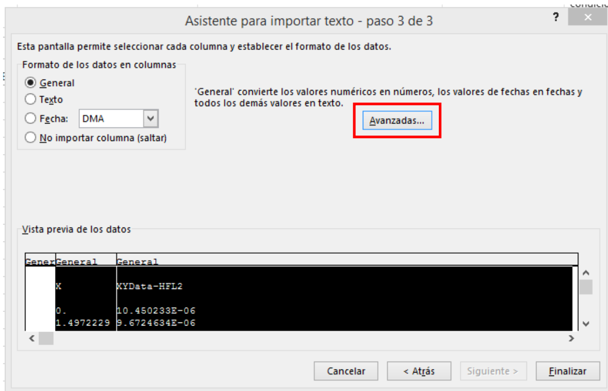
Figure 23: Save in a plain text file the object XY data



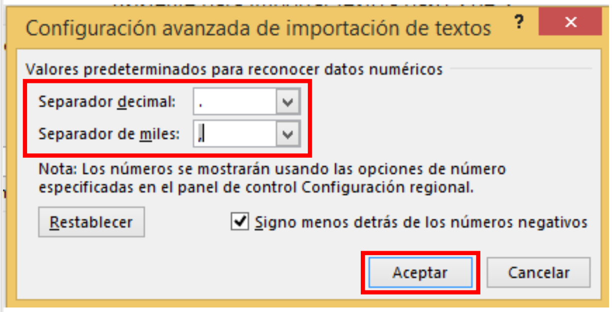
(a) Import data from text file into excel (I)



(b) Import data from text file into excel (II)

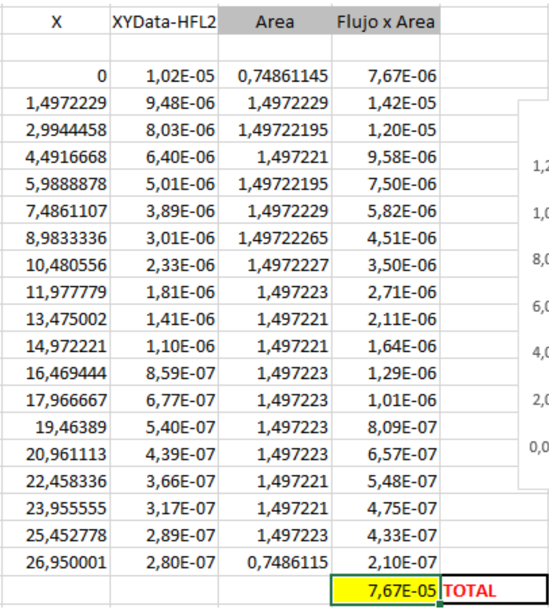


(c) Import data from text file into excel (III)



(d) Import data from text file into excel (IV)

Figure 24: Import data from text file into excel



X	XYData-HFL2	Area	Flujo x Area
0	1,02E-05	0,74861145	7,67E-06
1,4972229	9,48E-06	1,4972229	1,42E-05
2,9944458	8,03E-06	1,49722195	1,20E-05
4,4916668	6,40E-06	1,497221	9,58E-06
5,9888878	5,01E-06	1,49722195	7,50E-06
7,4861107	3,89E-06	1,4972229	5,82E-06
8,9833336	3,01E-06	1,49722265	4,51E-06
10,480556	2,33E-06	1,4972227	3,50E-06
11,977779	1,81E-06	1,497223	2,71E-06
13,475002	1,41E-06	1,497221	2,11E-06
14,972221	1,10E-06	1,497221	1,64E-06
16,469444	8,59E-07	1,497223	1,29E-06
17,966667	6,77E-07	1,497223	1,01E-06
19,46389	5,40E-07	1,497223	8,09E-07
20,961113	4,39E-07	1,497223	6,57E-07
22,458336	3,66E-07	1,497221	5,48E-07
23,955555	3,17E-07	1,497221	4,75E-07
25,452778	2,89E-07	1,497223	4,33E-07
26,950001	2,80E-07	0,7486115	2,10E-07
			7,67E-05 TOTAL

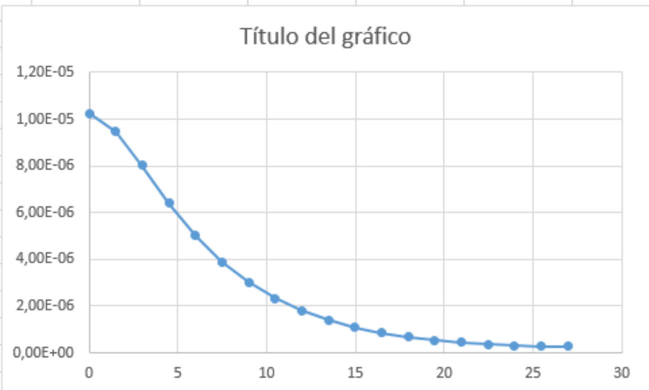


Figure 25: Integration of the total flow using the trapezoid method

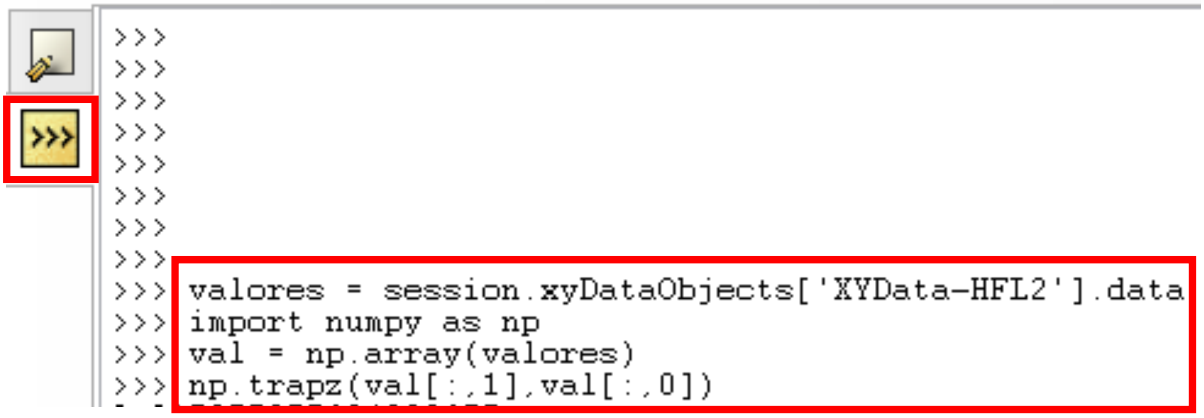


Figure 26: Integration of the total flow using the trapezoid method using `python`

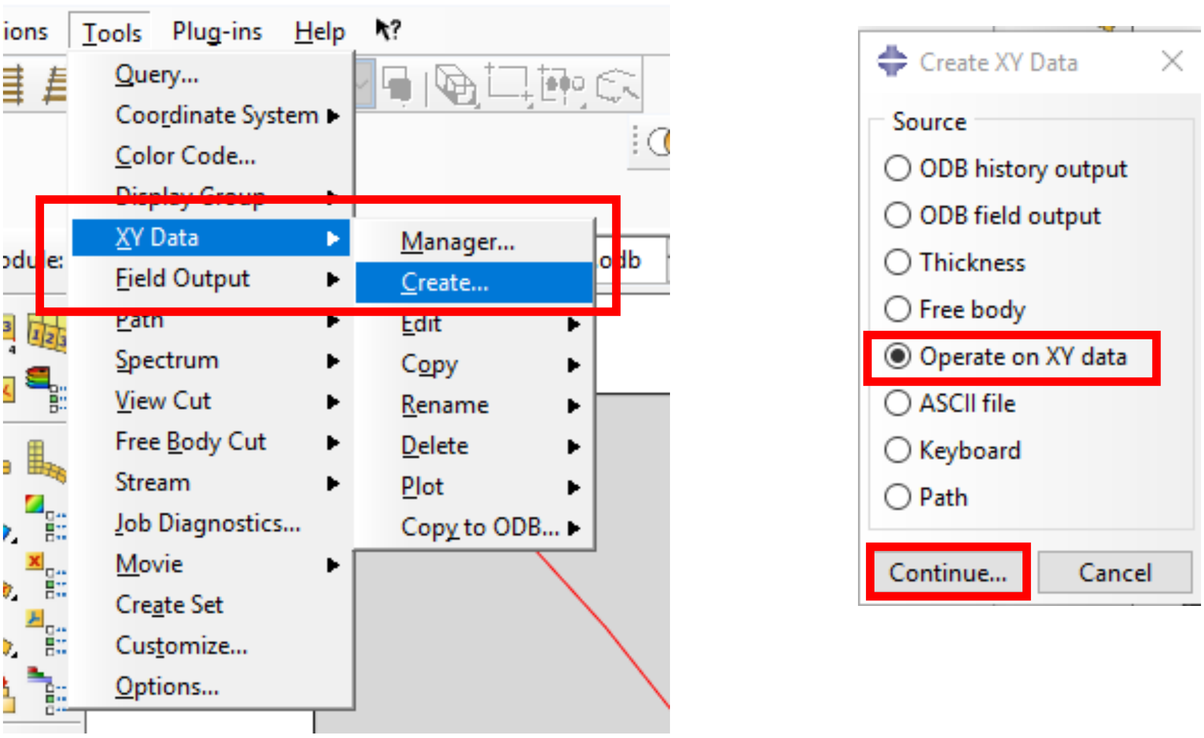


Figure 27: Selection of tool *Operate on XYData*

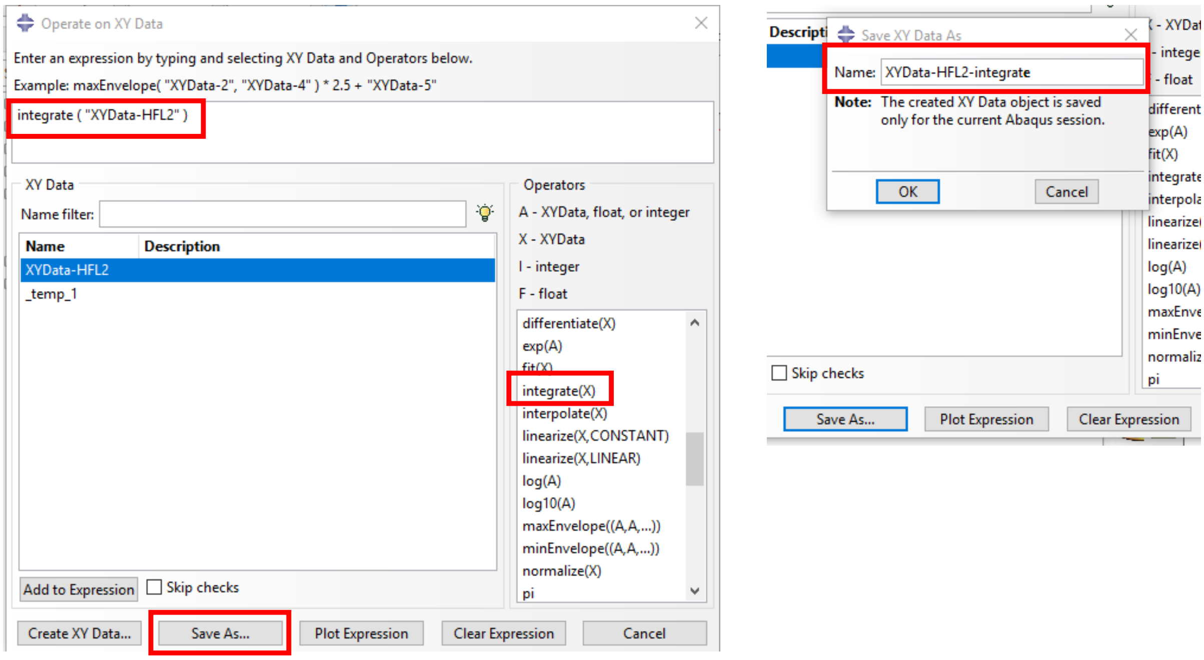


Figure 28: Integration of object *XYData*

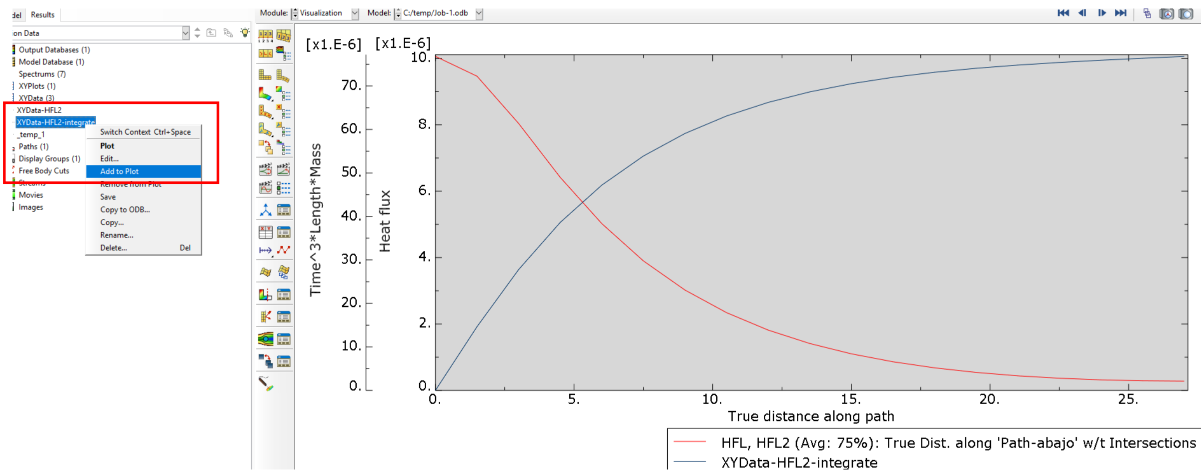
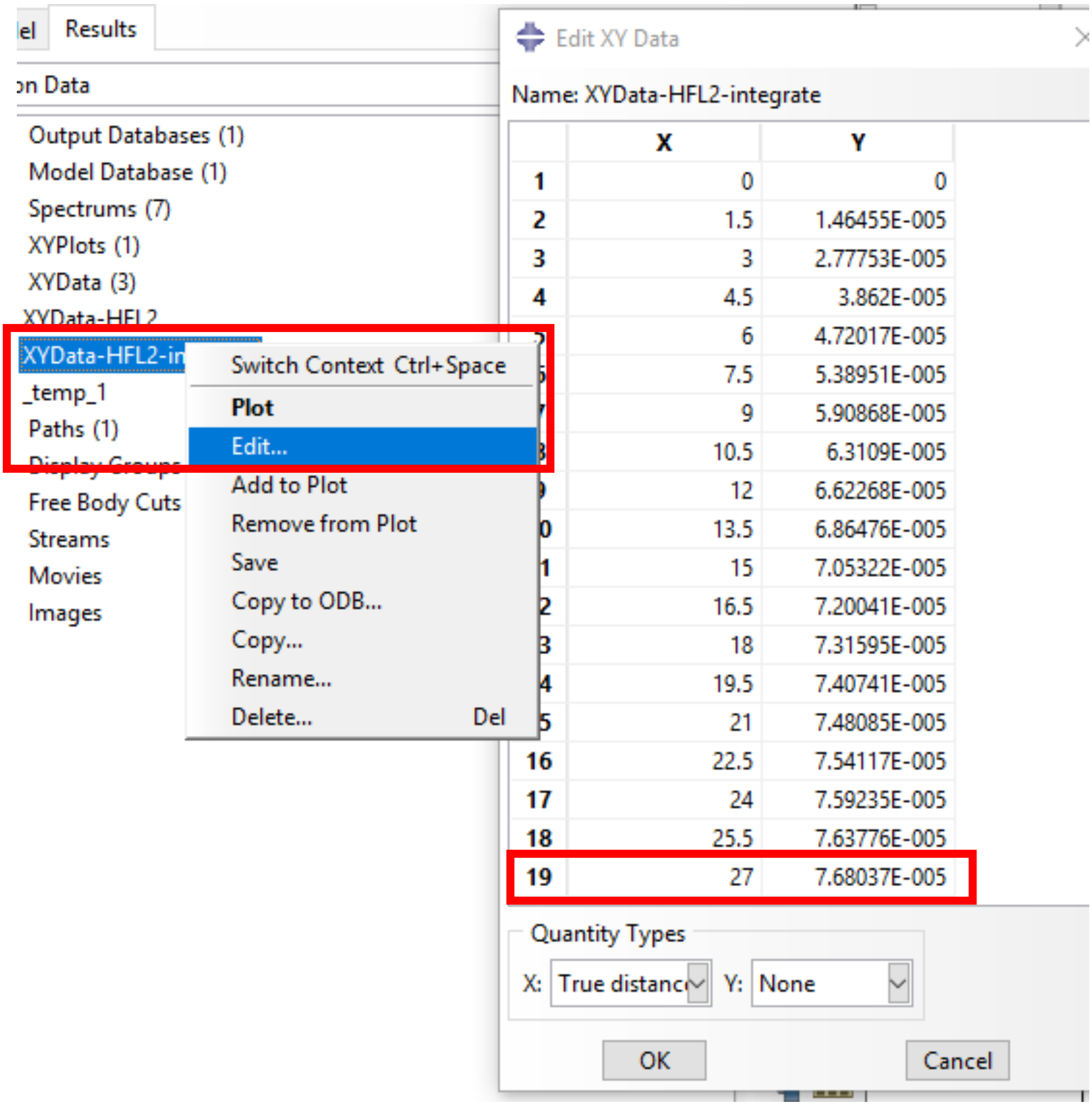
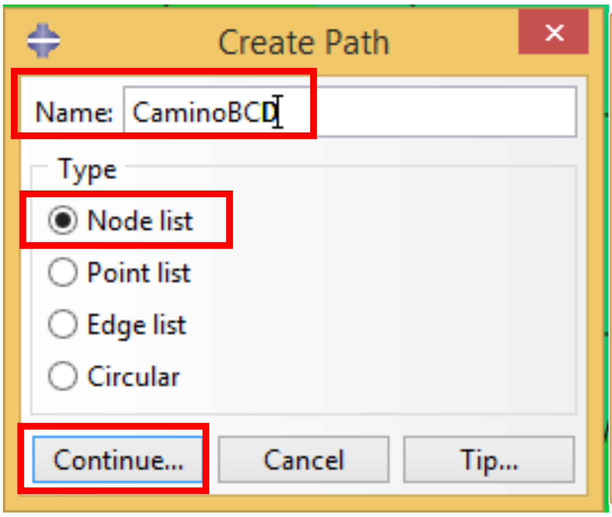


Figure 29: Variación del flujo y su integración a lo largo del *path*

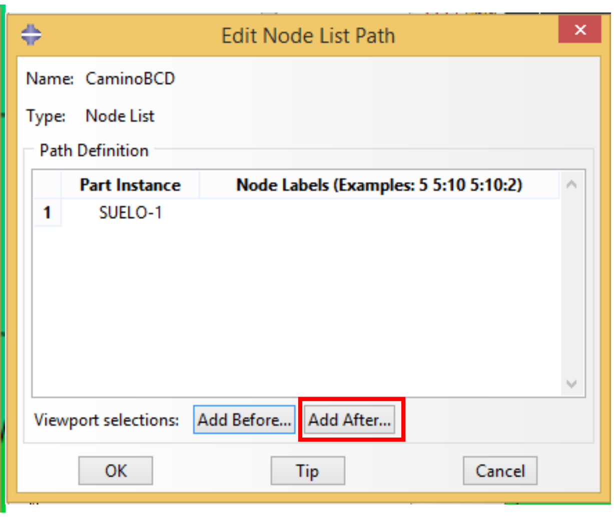


	X	Y
1	0	0
2	1.5	1.46455E-005
3	3	2.77753E-005
4	4.5	3.862E-005
	6	4.72017E-005
	7.5	5.38951E-005
	9	5.90868E-005
	10.5	6.3109E-005
	12	6.62268E-005
	13.5	6.86476E-005
	15	7.05322E-005
	16.5	7.20041E-005
	18	7.31595E-005
	19.5	7.40741E-005
	21	7.48085E-005
16	22.5	7.54117E-005
17	24	7.59235E-005
18	25.5	7.63776E-005
19	27	7.68037E-005

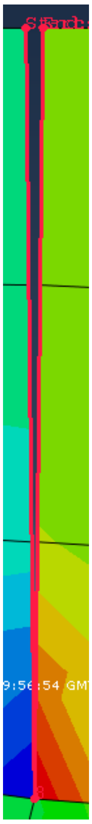
Figure 30: Integración del flujo a lo largo del *path*



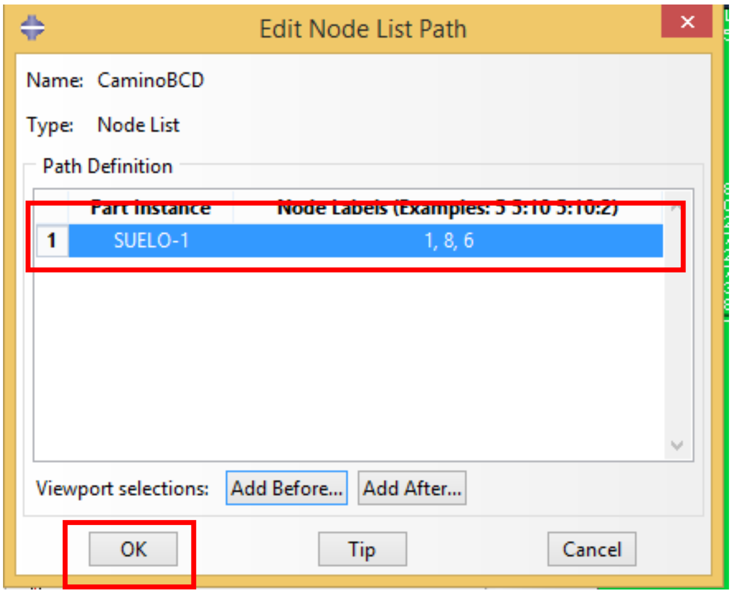
(a) *Create Path* dialog box



(b) *Path* geometrical definition

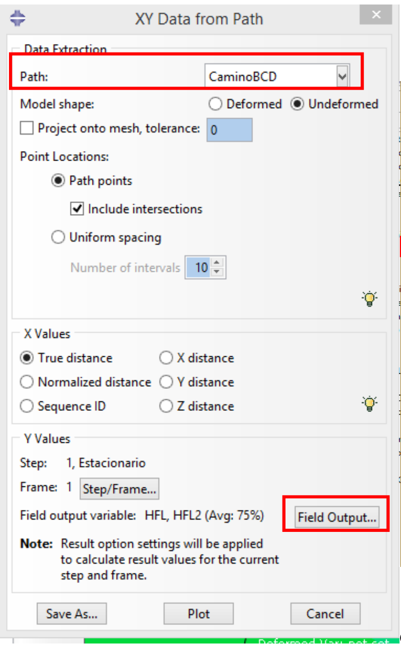


(c) Boundary BCD

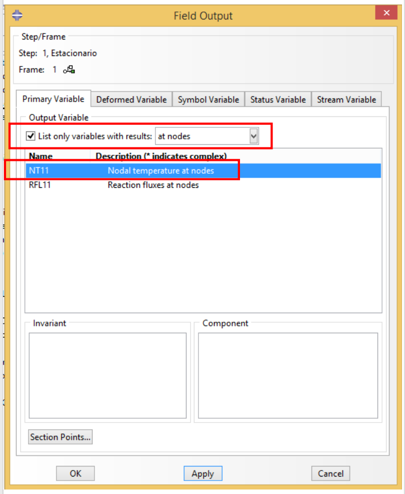


(d) *Path* geometrical definition

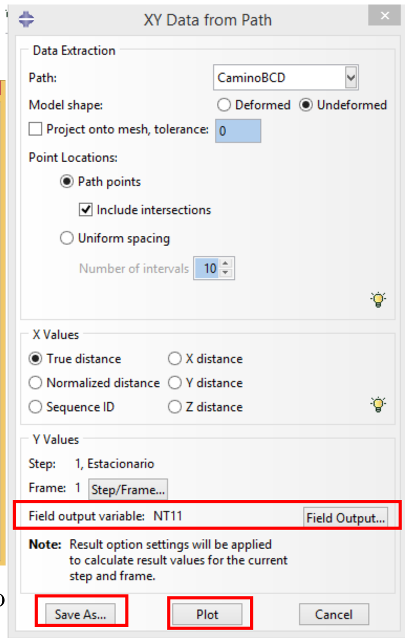
Figure 31: *CaminoBCD* path definition



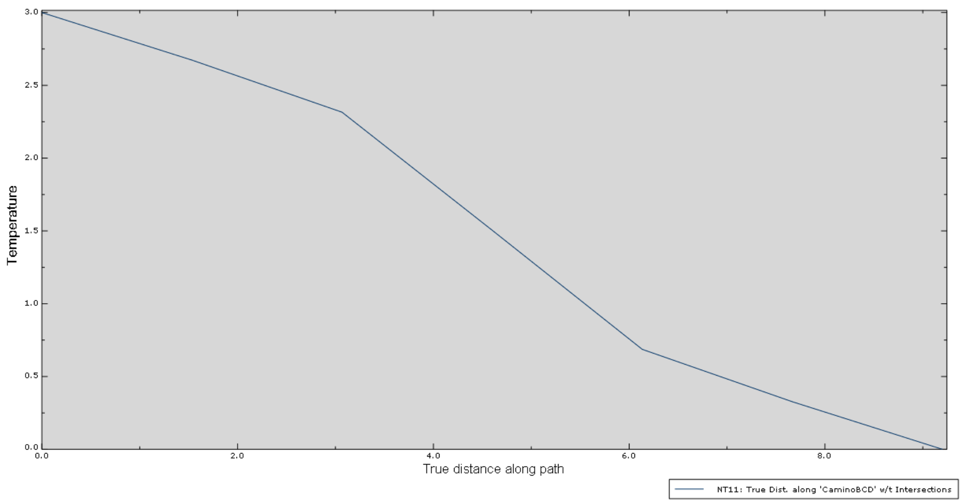
(a) *XY Data from path* dialog box



(b) Selecting the variable to draw in the path

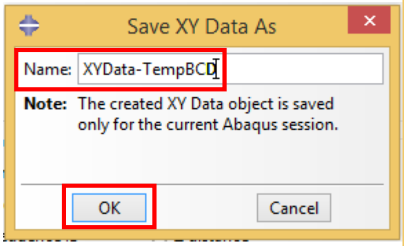


(c) *XY Data from path* dialog box

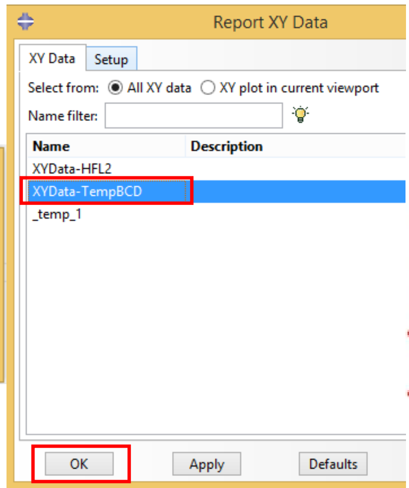


(d) Total head distribution in BCD

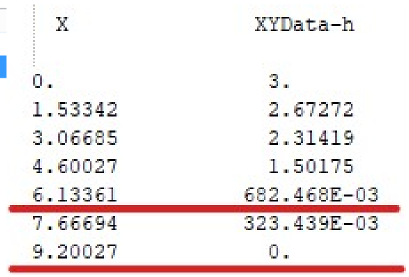
Figure 32: XY-object Data of the total head in the BCD path



(a) Save XY-Data object

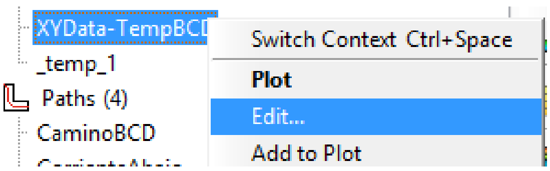


(b) Export XY-Data object to text file

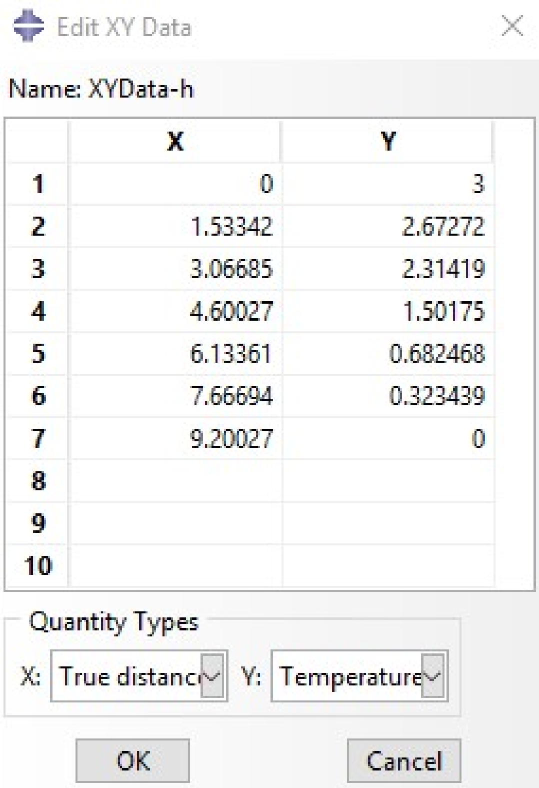


(c) Text file

Figure 33: Save XY Data object as text file



(a) Direct access to a XY-data object in the **Model Tree**



(b) Data in *XYData-h* object

Figure 34: Information stored in *XY Data* object

3 Proposed Exercises.

3.1 Proposed exercise 1

Let be a homogeneous and isotropic sand stratum with a thickness of 5.2 m and a permeability coefficient of $k_x = k_y = k_z = 1 \cdot 10^{-2}$ m/s. Underneath there is another homogeneous and isotropic stratum formed by silty sands with a thickness of 4 meters and with a permeability coefficient of $k_x = k_y = k_z = 5 \cdot 10^{-5}$ m / s. Under the silty sands there is a layer of clays that is assumed to be impermeable. In order to reduce the filtration due to the first layer, a sheet pile (of infinite length in the direction perpendicular to the plane of the drawing) has been embedded, which, by execution error, has only penetrated 4.6 m. To the left of the sheet piling (upstream) a height of 3 meters of water has accumulated and to the right (downstream) the runoff makes no accumulation of water. For the problem thus defined (see Fig. 35) and assuming a steady state, answer the question you are being asked in class.

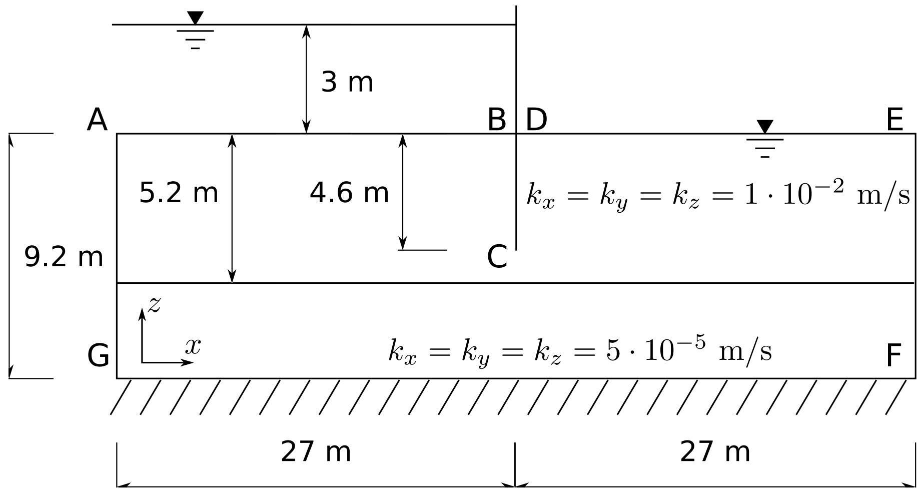


Figure 35: Model description

NOTE: Use the same type of elements as in the previous solved exercise and a global mesh size of 1.4 meters.

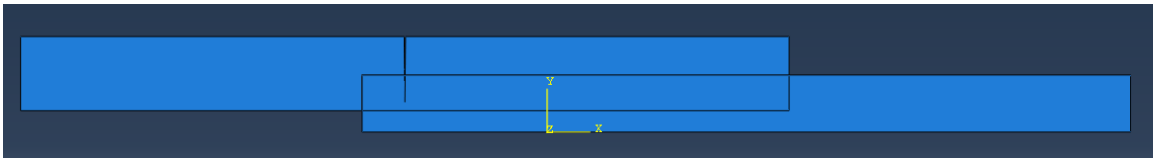
HELP: In this exercise there are two different materials that share a common border. To reproduce it in Abaqus follow the following steps:

- Create a different *part* for each stratum and assign the corresponding material. When you are in the **Property** module and you are assigning a section to a *part* (**Assign Section** command), remember not to create a *set*, as shown in Fig. 36 (we want to avoid problems in the subsequent **Merge** operation).

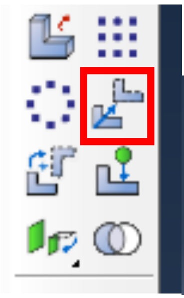
- In module **Assemble**, assemble a model with a dependent copy of each part as indicated in Fig. 37a. Then press the command **Translate Instance** (see Fig. 37b) until you leave them in their final position (see Fig. 37c).

- Finally we try to create a new *part* as the union of the two previous but conserving the common border between them. To do so press **Merge/Cut** (see Fig. 38a)

Figure 36: *Prompt* associated to **Assign Section**



(a) Initial position of the *parts* copies



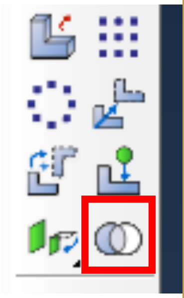
(b) Command **Translate Instance**

(c) Final position of the *parts* copies

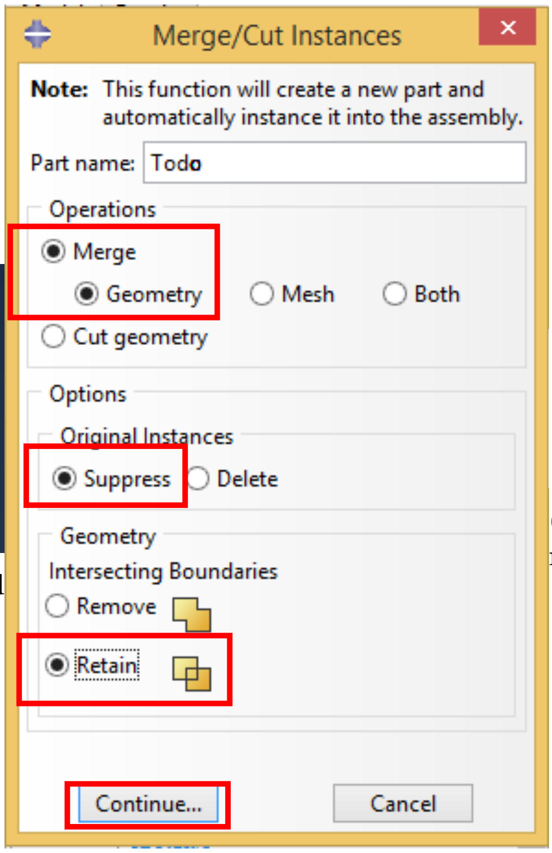
Figure 37: Assemble a model with two *parts* (I)

making sure you activate the **Retain Boundary** option (see Fig. 38b). Check that the *Assembly/Instances* node of **Model Tree** has created the new copy (see Fig. 38c)

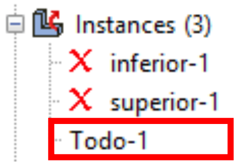
- Remember that when you build the mesh you should do it on the new **Part** created as indicated in Fig. 39.



(a) Command **Merge/Cut**



(b) **Merge/Cut** dialog box



(c) Copies from the model

Figure 38: Assemble a model with two *parts* (II)

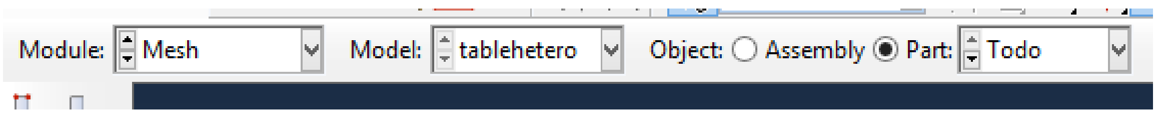


Figure 39: Mesh of new *part*

3.2 Proposed exercise 2

We have a dam with a base 27 meters wide built in concrete (which we will assume is impermeable) of infinite length in the direction perpendicular to the plane of the drawing. Under that base there is a homogeneous and isotropic stratum of silty sand with a permeability coefficient of $k_x = k_y = k_z = 5 \cdot 10^{-5}$ m/s and a thickness of 15 m. Under this layer there is a packet of clays that we will assume are impermeable.

A 5-meter-high membrane cut-off at the upstream edge of the dam is built to avoid filtration and, to avoid underpressure, a 7 m long and 1 m high toe-drain with permeability of $k_x = k_y = k_z = 1 \cdot 10^{-1}$ m/s is built too.

Upstream of the dam a height of 10 meters of water is accumulated and downstream the runoff makes no accumulation of water. For the problem thus defined (see Fig. 40) and assuming a steady state, answer the following questions:

1. Obtain the water outflow downstream (per unit length in y direction).
2. Obtain the resultant of the underpressure (vertical force) on the line DE per unit length in the y direction.

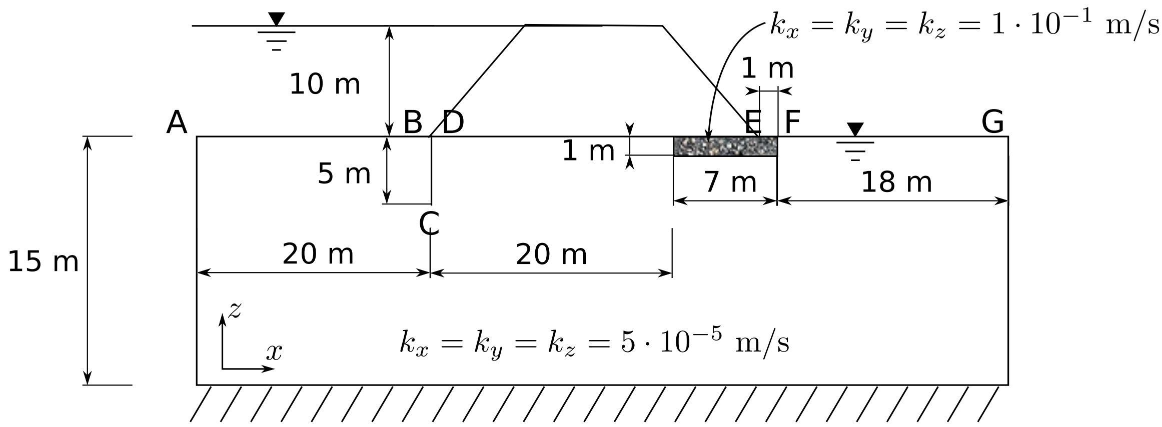


Figure 40: Model description

NOTE: Use the same type of elements as in the previous solved exercise and a global mesh size of 2.5 meters. The solution to question 1 is 1.942e-4 m^3/s per meter in the y direction. The solution to question 2 is 699.8 kN per meter in the y direction.

HELP: In this exercise there are two different materials that share a common border. To reproduce it in Abaqus follow the steps described in the proposed exercise 1.

3.3 Proposed exercise 3

We have a dam built in concrete (which we will consider impermeable) of infinite length in the direction perpendicular to the plane of the drawing. Under its base there is a layer formed by two homogeneous and isotropes materials of silty sand and under that layer there is a packet of clays that we will assume are impermeable. A membrane cut-off is built to reduce filtration at the upstream end of the dam (consider the membrane 0.05 m thick). The geometry and permeability properties of the materials are described in Fig. 41. We will assume that the vertical boundaries of the stratum at the ends are so far from the dam as to be considered impermeable. At point D we want to simulate the effect of a pump that extracts a flow of 0.0001 m^3/s per linear meter in the direction z. To intruduce that Neumann-type boundary condition, consult in the Abaqus' help the command **Concentrated heat flux** (think about the sign that you must set to simulate that we are extracting flow).

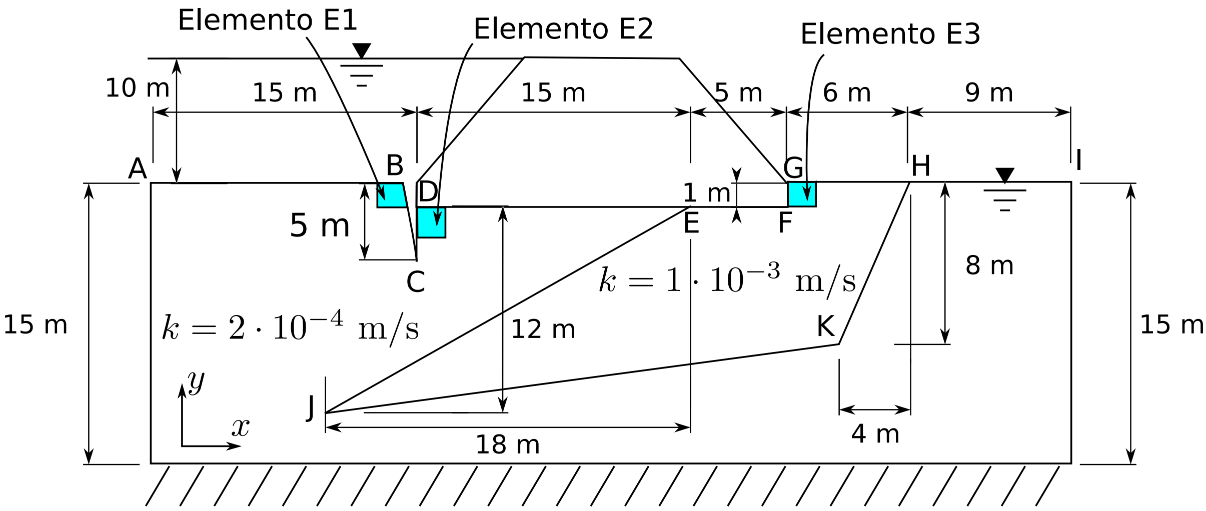


Figure 41: Problem description

Answer the questions that follow with the following considerations:

- Use, when defining the mesh, quadrilateral elements, mesh technique *Free*, with a global mesh size of 1.8 meters and quadratic interpolation (element *DC2D8*). Do not subdivide the parts to get a more regular mesh.
- Consider the fluid is fresh water with density $\rho_w = 1000$ kg/m^3 and the acceleration of gravity is $g = 9.81$ m/s^2.
- To facilitate the calculation of the underpressure, consider that the origin of the heights (*z=0* m.) is at the base DF of the dam.

For the problem thus defined, calculate the outgoing flow in the straight line GH per unit of meter in the direction z (the solution is 0.00108 m^3/s/ml) and the vertical force of sub-pressure under the dam (the solution is 613.37 kN/ml).

3.4 Ejercicio Propuesto 4

Let be a soil stratum where two steel sheets (of infinite length in the direction perpendicular to the plane of the drawing) have been embedded. This stratum is composed of three homogeneous and isotropic silty sand soils. Underneath, there is a packet of clays that we will assume is impermeable. The geometry and permeability properties of the materials are described in Fig. 42. To the left of sheet pile 1, a volume of water with a constant height above the H point of 15 meters is accumulated. Between sheet piles 1 and 2 we build a concrete plate that makes the CD contour impermeable. To the right of sheet pile 2 there is a network of injectors that apply a flow of value $q = 10^{-3}$ m^3/m^2/s in the EF contour (positive indicates flow into the surface). Finally, in the FG contour, runoff provoques that water does not accumulate above the ground. We will assume that the vertical borders of the stratum at the ends are so far away from the sheet piles as to be considered impermeable.

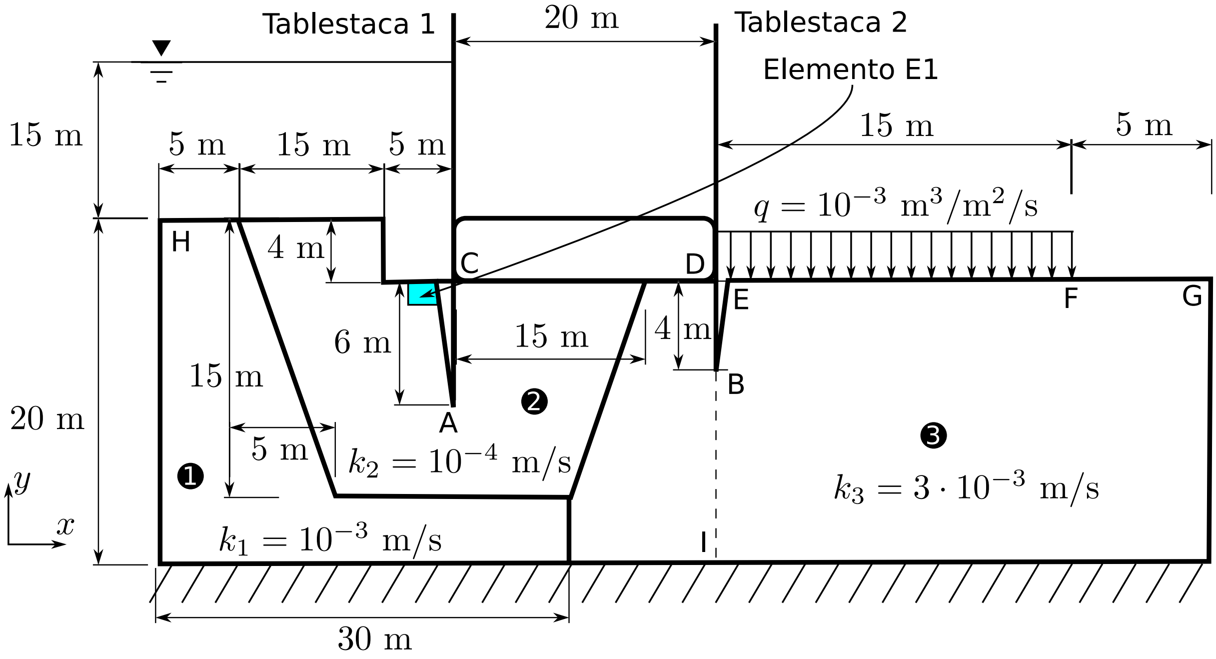


Figure 42: Problem description

Use the following considerations (since it is a plane problem, we will take a unit thickness and the values of flow and force will be per meter in the z direction):

- When defining the mesh quadrilateral elements, use meshing technique *Free*, with global mesh size 1.3 meters and linear interpolation (the element type should be *DC2D4*). Do not subdivide the parts to obtain a more regular mesh.
- Consider the fluid is fresh water with density $\rho_w = 1000$ kg/m^3 and that the acceleration of gravity is $g = 9.81$ m/s^2.
- The line that separates materials 1 and 3 is vertical.
- **Use as the origin of geometric heights the horizontal line that passes through H.**

For the problem described and assuming a steady state, calculate the vertical force of underpressure under the CD block (1454.8 kN), the pore pressure of the fluid at point A

(170.6 kPa), the total head of point B (2.67 m), the modulus of the flow vector at the centroid of the element E1 ($9.62 \cdot 10^{-5}$ $m^3/m^2/s$) and the flow through the straight line BI (0.00275 m^3/s).