Generating mesh with PDE tool in Matlab

Division of Solid Mechanics

In this assignment you should create your own mesh using the built-in PDEtool in MATLAB. PDEtool can be used directly to create simple geometries with little effort. To start just type `pdetool` into the MATLAB terminal to open PDEtool user interface. Although the tool is originally designed for solving partial differential equations only the meshing aspect are of interest in this assignment. As an example consider the segment of a dog bone geometry that should be meshed, see figure 1.

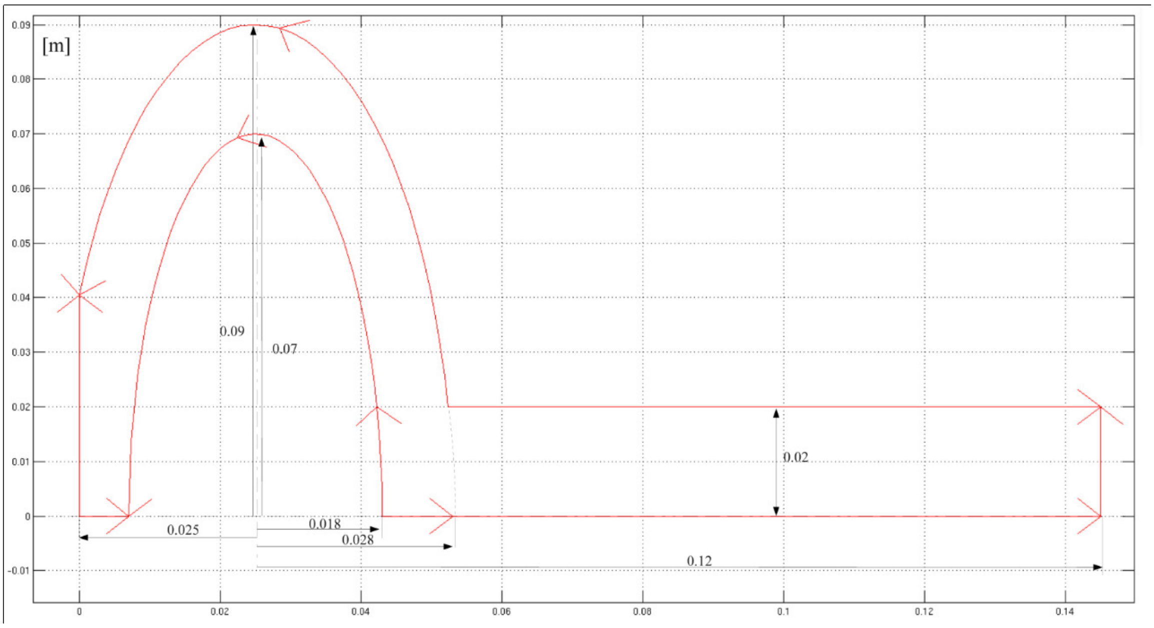

Figure 1: Dog toy geometry.

A step by step tutorial of how to use some of the basic features of PDEtool are provided here. Before starting to draw it is often convenient to activate grid and change the axis scales. This is done using <Options> menu. To build our geometry simple geometries are added one by one. The basic shapes used in PDEtool are rectangles, ellipses and polygons. Basic drawing tools can be found on the toolbar (see figure 2, note polygons are not necessary to create the geometry for the lab).

Draw a rectangle: Create a rectangle by using the rectangle tool (see figure 2) and simply hold left mouse button and move the mouse cursor to obtain the desired size. To adjust the size and position double click on the rectangle and a dialog will appear where coordinates and dimensions can be provided, see figure 3.

Draw an ellipse: Create a ellipse by using the draw ellipse tool (see figure 2) and simply hold left mouse button and move the mouse cursor to obtain the desired size. Double click to adjust position and size in similar fashion as for rectangles.

Determining combined geometry: In the box below shape-buttons the present features are

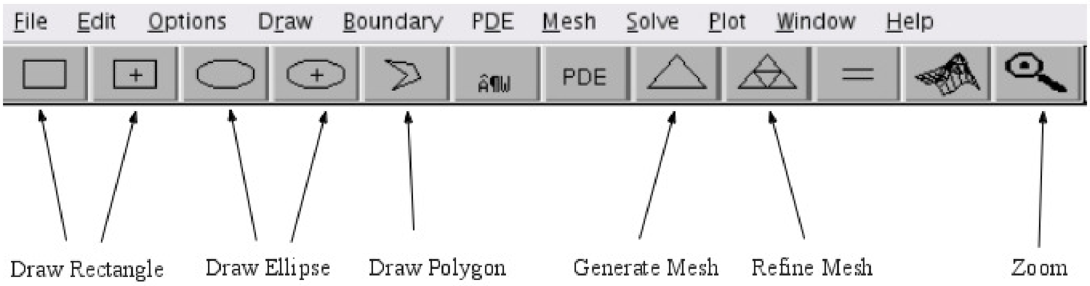

Figure 2: Basic tools for drawing in PDEtool.

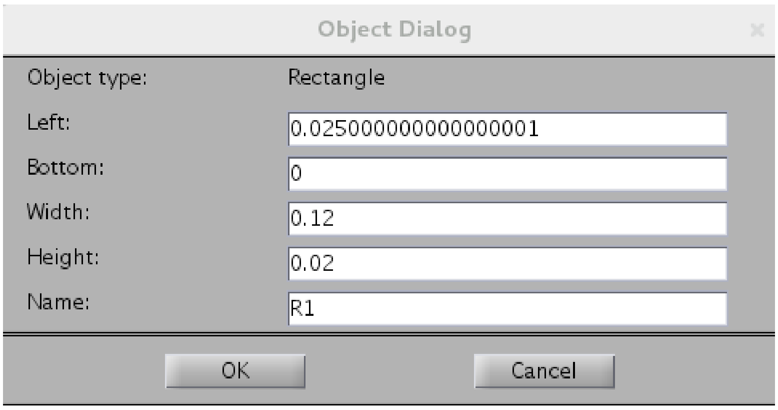

Figure 3: Rectangle object dialog.

presented by name (which may be edited using object dialog). The plus sign indicates that geometries are added and minus signs indicate that they are cut out. Because of this the order is important i.e **R1+E1-E2 $\neq$ R1-E2+E1 in general**. Below figues showing drawn features and the resulting body after combining by grouping and changing signs in the set formula row (see figure 4, 6 and 5). Note that to see the resulting geometry we need to go to either meshing or boundary view (see toolbar).

When the necessary features are added go to <Boundary> / <Boundary mode> to see which are the boundaries of your geometry. The red arrows define the main borders and the grey contours represent the so called subdomains. For instance, subdomains appear where features overlap. To get a uniform mesh it is convenient to remove the subdomains. To do so use <Boundary> / <Remove all Subdomain Borders>. If the mesh consist of different natural domains such as different materials it is favourable to keep the subdomains.

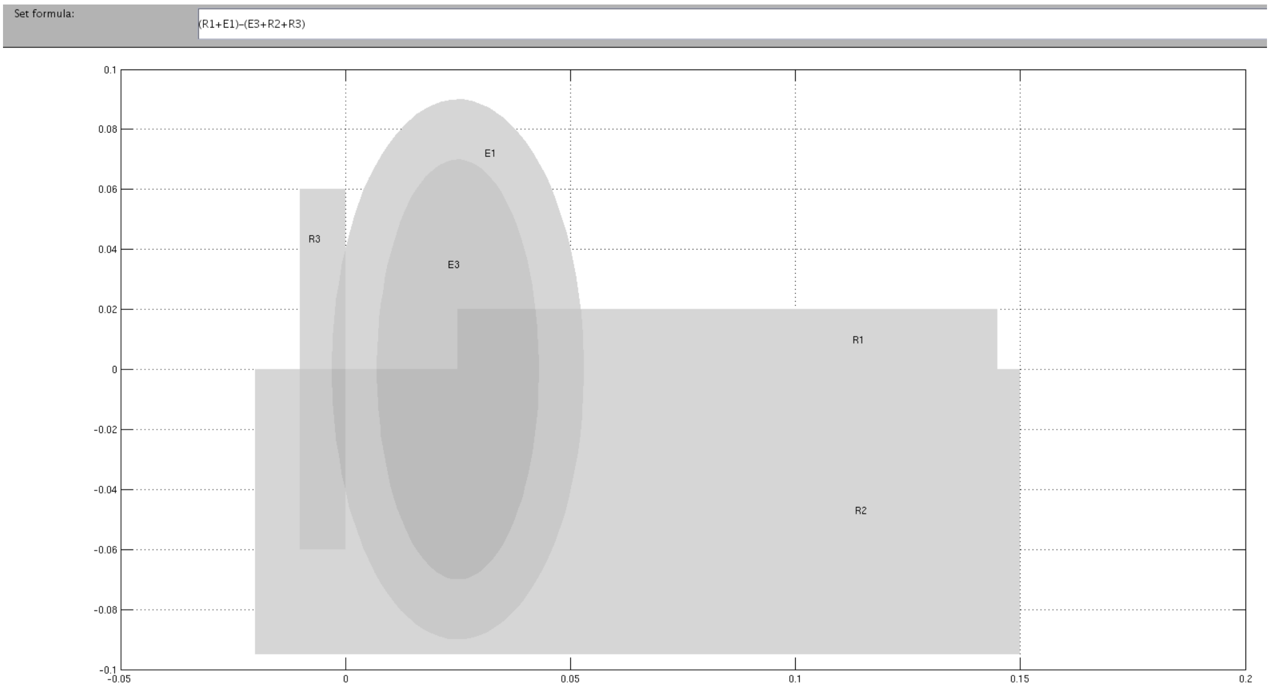

Figure 4: Set of combined shapes.

To obtain the geometry provided in figure 5 the formula used is given in figure 6 and the object names is seen in figure 4.

To generate a mesh simply press mesh tool when the geometry seen in boundary mode is satisfactory (see figure 2). To refine the mesh, i.e. create more elements, press refine mesh until enough elements are generated (remember that finer mesh means higher computational cost but better accuracy). Note that a high number of elements increases the computational time.

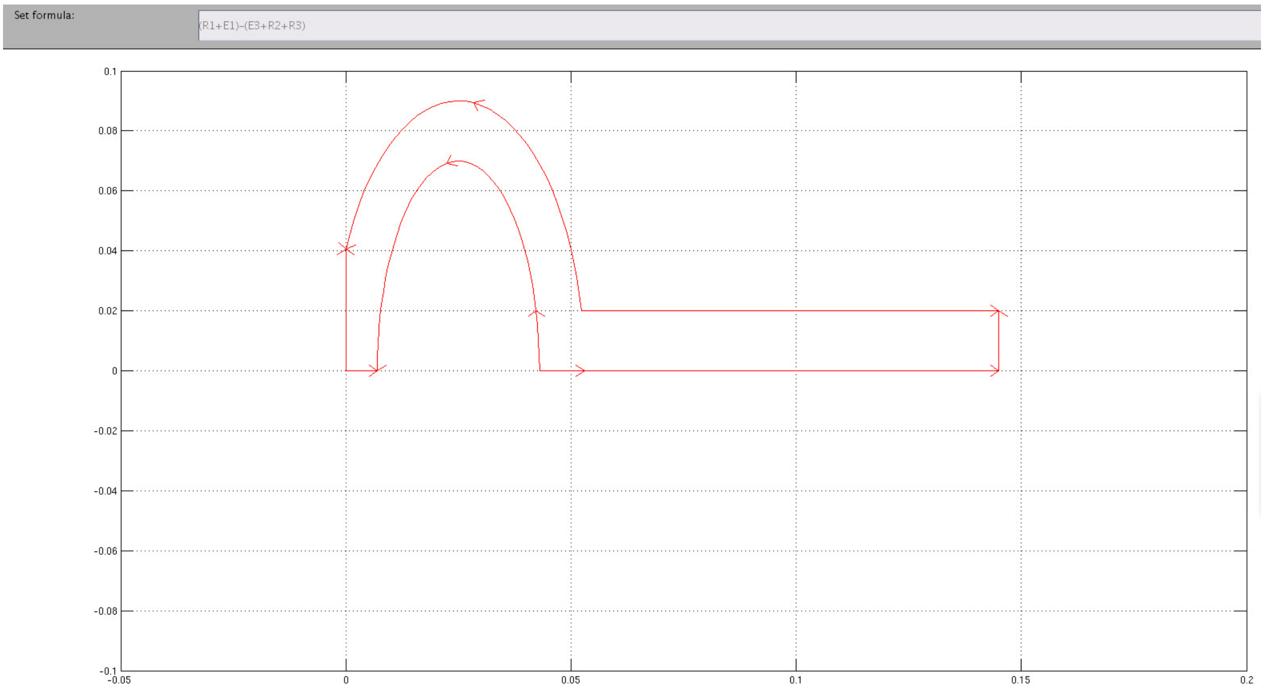

Figure 5: Resulting geometry seen in boundary mode (ctrl+B) after all subdomains have been removed.

Figure 6: Formula determining active areas.

When you are content with your mesh use <Mesh> / <Export mesh> to save the topology matrices associated with the current mesh (note you have to save the work separately to keep geometries etc. since <Export mesh> will only save matrices). The topology matrices are `p,e,t` (points, edges, triangles). The exported matrices will directly appear in the active MATLAB workspace. It is strongly advised to save them directly in a `.mat` file (mark variables in workspace, right click and save) so it can be loaded in the scrips you write. From `p,e,t` the traditional CALFEM quantities `edof`, `dof`, `coord` etc can be extracted. On the course web page for FEM FAQ are some instructions of how this is done.

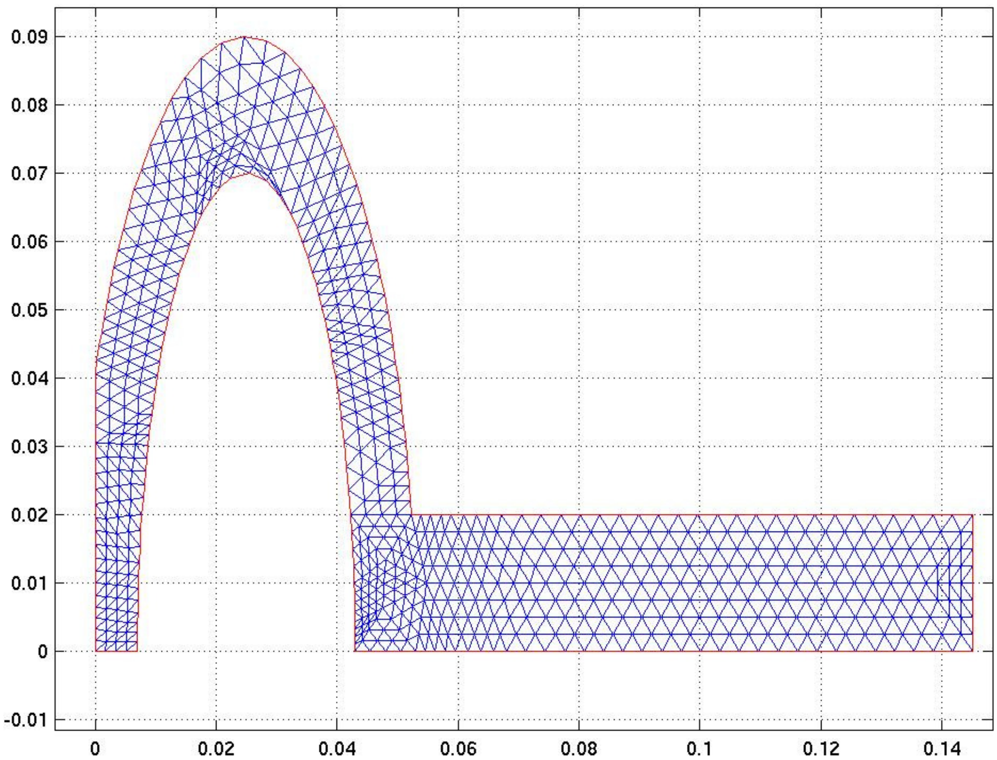

Figure 7: Mesh after two refinements.