

## Tutorial 1: Introduction to GMSH

This tutorial will make a simple demonstration of the following:

1. Creating geometry in GMSH
2. Meshing in GMSH
3. Structured Meshing in GMSH
4. Linear Elasticity in Elmer

Because this is expected to be the first tutorial, I will address many key concepts of the software as I go along - particularly the ways these tools will differ from commercial software. In general, the difference is that the free software tools expect you to know more about what you are doing - a bit about the background of FEA and the analysis you plan to perform. Particularly a background in the concept of Mesh generation is helpful. Programs such as ANSYS typically allow the user to neglect concepts and then makes decisions internally that the user did not see or control. Freeware does not do any guessing - to maintain accuracy they mostly use published Algorithms and Methods so you can do your own due diligence on getting an accurate solution. Another characteristic is that the system of Units is never expressly defined. As a user you must maintain a system of units through the work as you go along. I recommend always keeping a notepad close by to keep track of changes made (sometimes they are hard to find after-the-fact) and settings changed as you go.

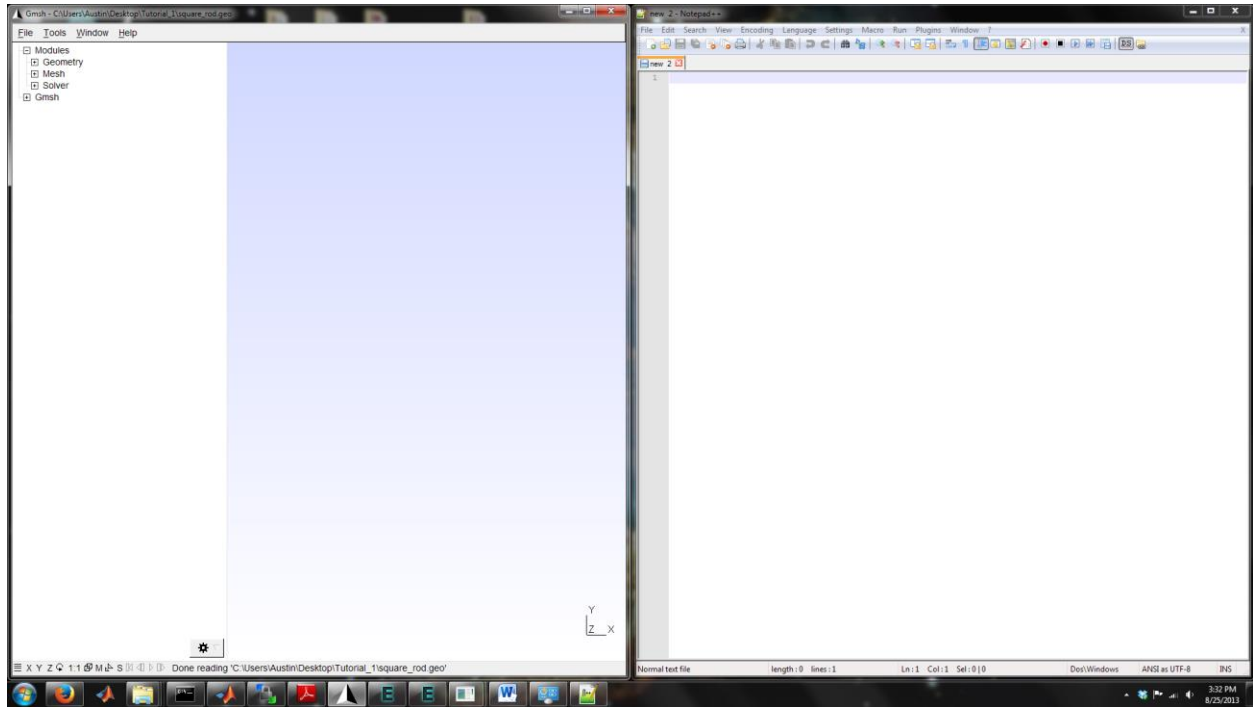
These tutorials will always assume you have the software running well and any bugs or ‘features’ unique to your machine are probably not addressed. I am running these on Windows 7 Service Pack 1. Grab a notepad (I’m serious about that) and let’s get started!

### Creating geometry in GMSH

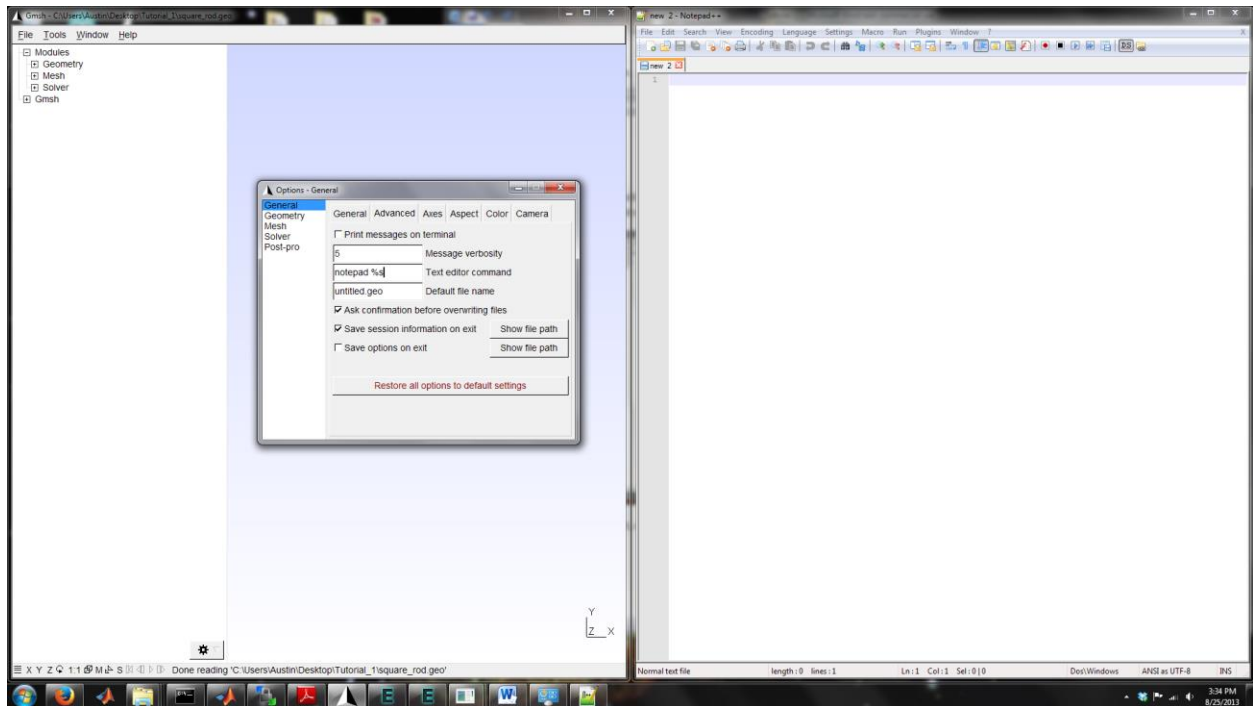
Geometry and mesh generation will be handled in GMSH. GMSH allows geometry creation from primitives, or the import of general CAD formats (.iges). The term ‘primitives’ refers to abstract geometry - points in space, lines and arcs between those points, surfaces between those lines, and so on. While slower to pick up than SolidWorks style click-and-drop modeling, the creation of geometry with these concepts is philosophically often more satisfying for two reasons. First, when you go to draw on a sheet of paper, you naturally start by thinking about what you plan to end with, then define outline points, then draw lines between them, etc. This Primitives based modeling is consistent with proper design, especially inasmuch as the first step requires - you must think (what a concept!) about what you are doing before you start drawing. The other advantage is that such a process easily lends itself to structured understanding of the parts of the model. This is especially important as the Mesh (described later) - and thus your analysis results - will only be as well defined as your geometry. This structure comes in the form of ‘scripting’ or ‘coding’ the geometry. While you rarely have to touch code to use these tools (the way I am going to present them), keeping track of the code helps you know where you are and perhaps where you went wrong.

We will create the geometry in GMSH by using the GUI (Graphic User Interface). The information about the geometry will be saved in a *part\_name.geo* file, where *part\_name* is replaced by the name you choose for the file. Every command made in the program GUI is entered onto the *.geo* script immediately after occurring. The geometry can be built either by using the GUI, or by modifying the *.geo* script

directly. First, upon opening GMSH, the following screen will appear. I have opened notepad++ to the right so the .geo file script can be followed simultaneously.



First, upon opening GMSH, select 'Tools' -> 'Options' on the toolbar. A window should appear

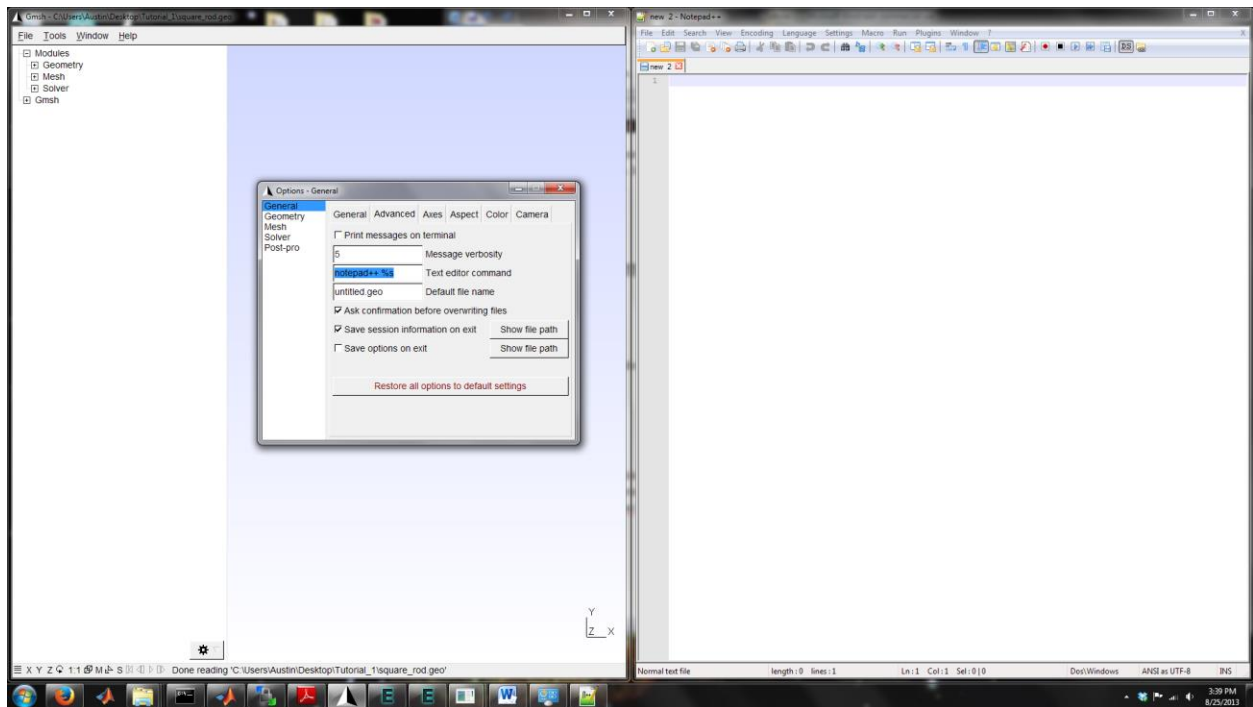


This pane gives options for every part of the program, and is particularly useful to change visualization settings. In this case, we want to change the selected text editor from notepad to notepad++.

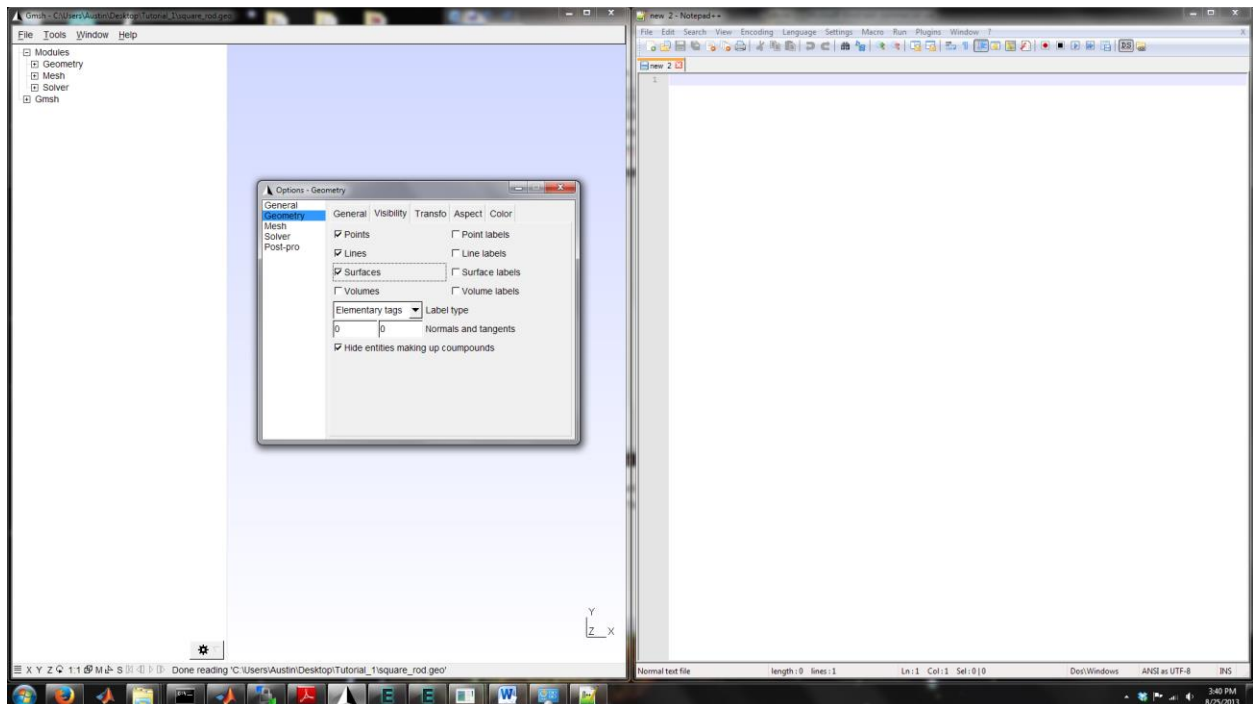
This is done by selecting (in the Options windows) 'General' from the sidebar, and replacing

notepad %s      with      notepad++ %s

(note, this will only work in Windows if the User Variable 'ROOT' has been updated to include the path to npp.exe. If that sentence meant nothing, please read the document titled 'Effective GMSH Elmer Paraview Interfacing' on the website before continuing)



After that, select 'Geometry' in the side bar, then select the 'Visibility' tab. make sure the radio button next to 'Surfaces' is checked. if not, check it by clicking in the box.

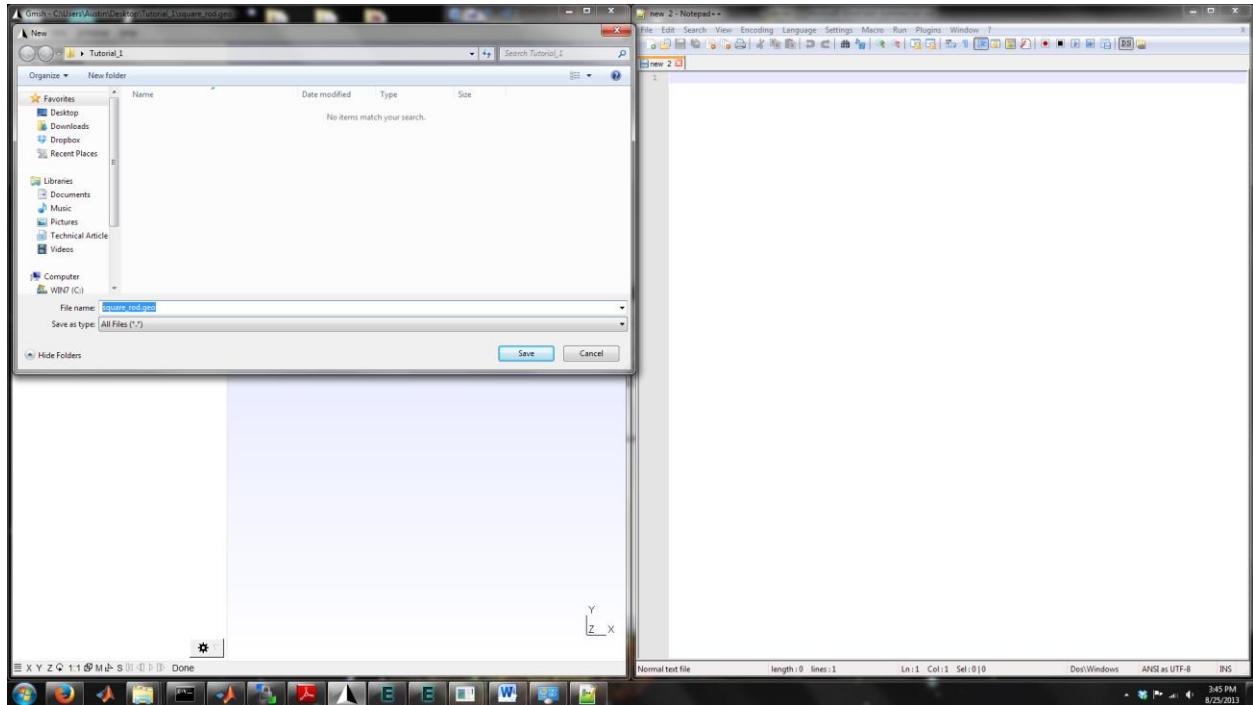


Close the 'Options' window by clicking the exit button in the top right of that window.

Now it is time to create the new geometry file. Select 'File' -> 'New...' find the directory where you wish to save the file (I am in .../Desktop/Tutorial\_1). Enter the File name, be sure to include the '.geo' after the file name. I choose to call the file 'square\_rod' and so enter:

square\_rod.geo

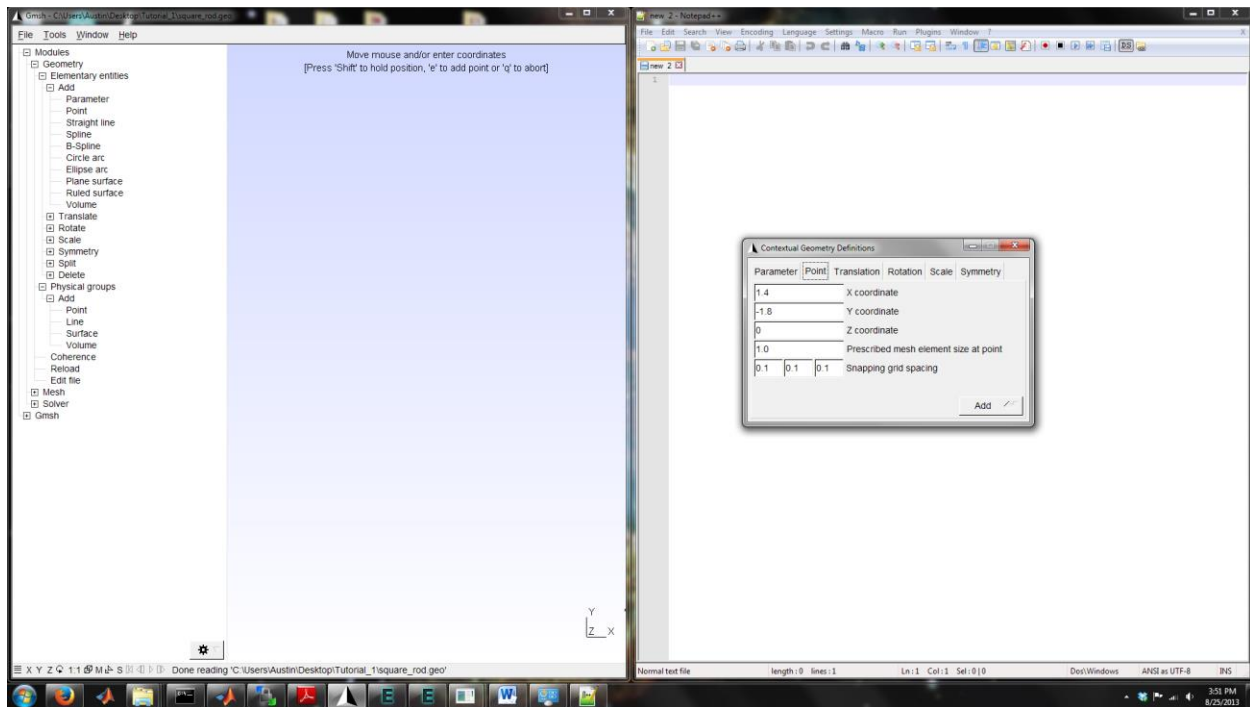
Click 'Save' to accept the name.



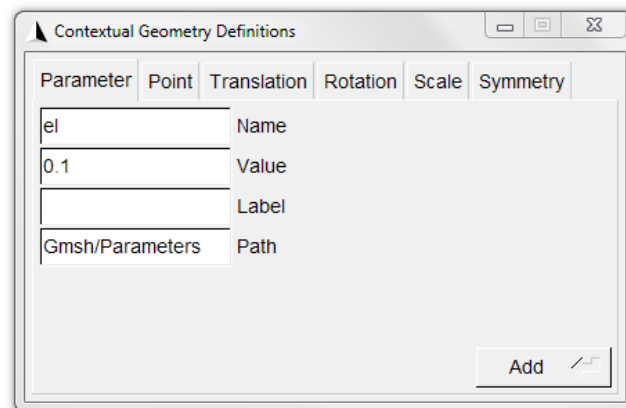
We will start by adding some points to define the geometry. In the Tree on the left of the screen, open

‘Modules’ -> ‘Geometry’ -> ‘Elementary entitites’ -> ‘Add’ -> ‘Point’

A ‘Contextual Geometry Definitions’ (‘CGD’) window appears that allows the user to enter points in space. I suggest moving the ‘CGD’ window off of the geometry viewing area (blue region) of the screen, as any mistaken clicks there will affect where the points are located.

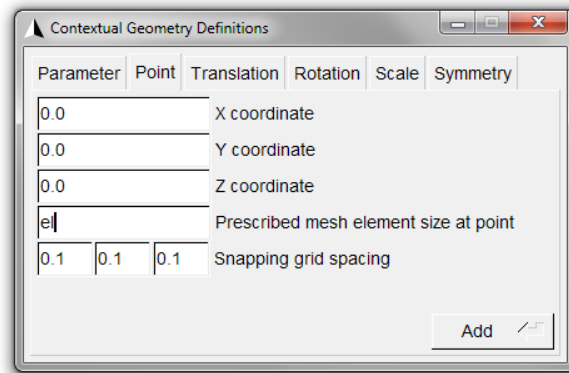


The values shown for the coordinates will be the last place the mouse pointer hovered over the geometry viewing area, disregard their values. To help keep our code clean later, we are going to add a constant now. Open the ‘Parameter’ tab in the CGD window. The variable will later define the mesh size, and will be reviewed later. For now, just enter the name as you please (I use  $\epsilon_1$ ) and give it the value 0.1 . click ‘Add’ to add the constant.



Now, return to the ‘Point’ tab. We will enter the first point at the origin. Input the following values

X coordinate - 0.0  
 Y coordinate - 0.0  
 Z coordinate - 0.0  
 Prescribed mesh... -  $\epsilon_1$   
 Click 'Add'



Click 'Add'. This will add a point at the origin  $\{0.0,0.0,0.0\}$ . For this, and all following points 'Prescribed mesh element size at point' is defined using the constant  $\epsilon_1$ . This allows the 'Prescribed mesh element size at point' value at every point to be changed simultaneously by a change of the value of the constant  $\epsilon_1$ . In this manner any values can be linked together for quick adjustment later via the .geo text file.

Now add the following points  $\{0.0,1.0,0.0\}$ ,  $\{0.0,0.0,1.0\}$ , and  $\{0.0,0.5,1.0\}$  in a similar manner:

X coordinate - 0.0  
 Y coordinate - 1.0  
 Z coordinate - 0.0  
 Prescribed mesh... -  $\epsilon_1$   
 Click 'Add'

X coordinate - 0.0  
 Y coordinate - 0.0  
 Z coordinate - 1.0  
 Prescribed mesh... -  $\epsilon_1$   
 Click 'Add'

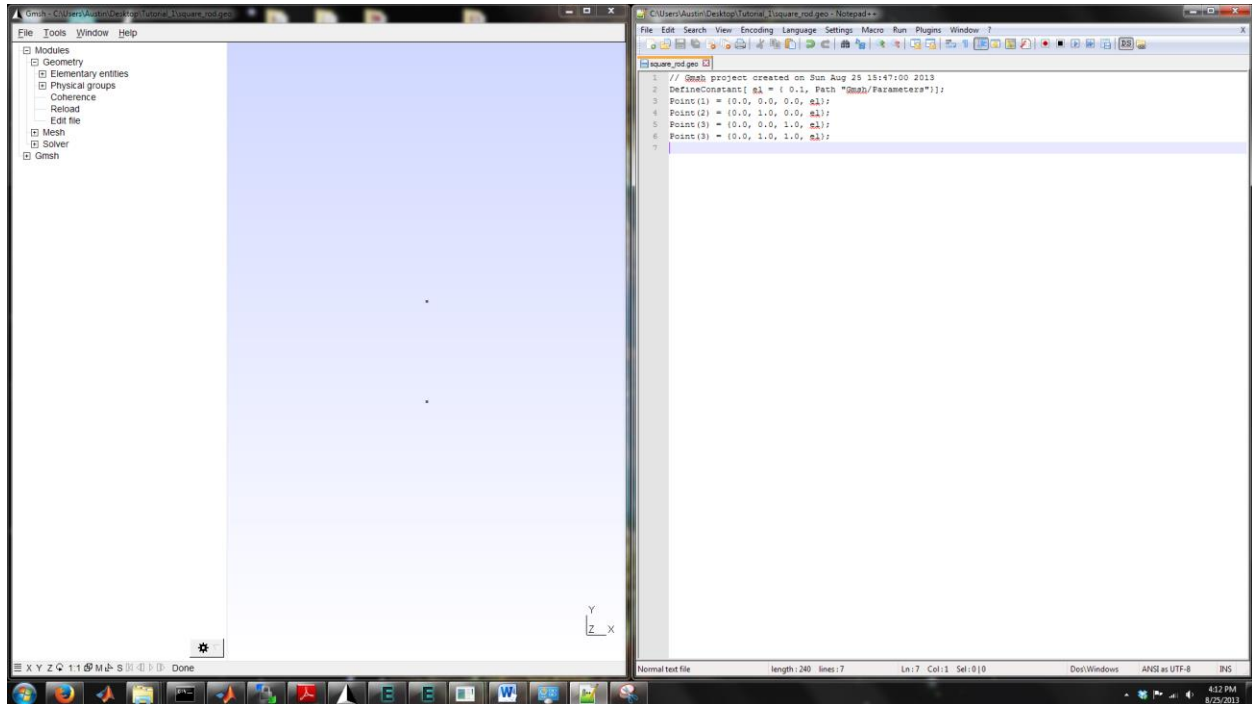
X coordinate - 0.0  
 Y coordinate - 0.5  
 Z coordinate - 1.0  
 Prescribed mesh... -  $\epsilon_1$   
 Click 'Add'

Exit the CGD Window. Note the comment in the top of the geometry viewer advising to 'press 'q' to abort'. Press q to exit the 'Add' 'Point' function that was being used.

It is now time to review the .geo file for the first time. In the sidebar Tree, select

‘Geometry’ -> ‘Edit File’

The *square\_rod.geo* file appears in the Notepad++ editor.



Here, every step we have made so far has been recorded. The constant  $\epsilon_1$  is defined, as well as each point that was drawn.

You will notice that, though we clearly entered four points, only two are shown on the screen. This is because we have ‘drawn’ in the Y-Z plane, and are currently viewing the X-Y plane. To change the view, select ‘X’ from the very bottom left of the GUI. This toolbar controls the viewing plane, zoom level, etc. this is best used to view one of the ‘plan views’, or 2D orthogonal views. Generally, a 3D model is best manipulated using the mouse (practice these):

click and hold the left mouse button in the viewing area; drag to rotate the part

click and hold the right mouse button in the viewing area; drag to drag the part

click and hold the center mouse button in the viewing area; drag to zoom in and out

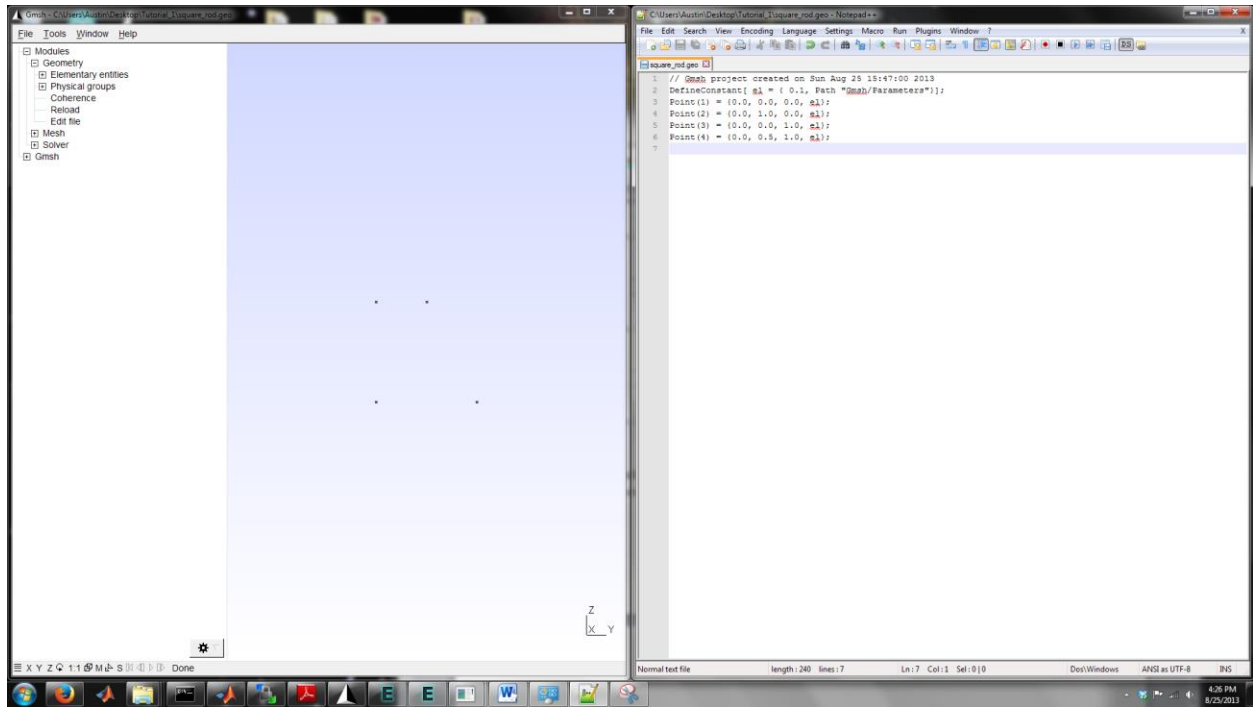
scroll the mouse scroll-wheel in the viewing area to zoom in and out towards/away from the pointer

Reset the view by selecting ‘X’ in the very bottom left toolbar.

Click ‘1:1’ in the same to fit the current drawing to the window size.

The windows should appear as follows:





Any change made in the GUI can affect the script, and vice versa. Any change made in the GUI immediately updates the .geo file. Changes made in the .geo file will not immediately affect the GUI or geometry. We now make a change in the script just to show the process. Suppose the point {4} is not in the correct place. We input it at {0,0.5,1.0} and now want to move it to {0.0,1.0,1.0}. To modify this:

Change the line in the .geo file

```
Point(4) = {0.0, 0.5, 1.0, el};
```

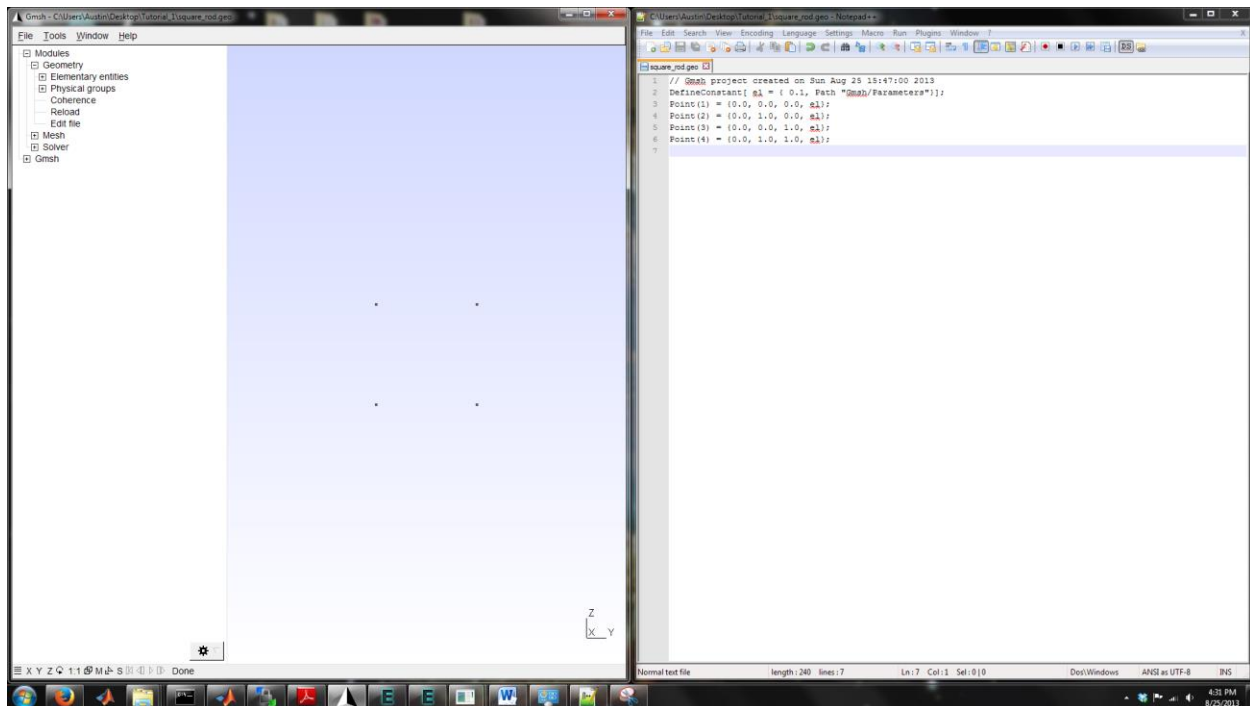
to

```
Point(4) = {0.0, 1.0, 1.0, el};
```

Save the .geo file by clicking the save icon in the toolbar of notepad++

Reload the new geometry in GMSH by clicking 'Geometry'->'Reload'

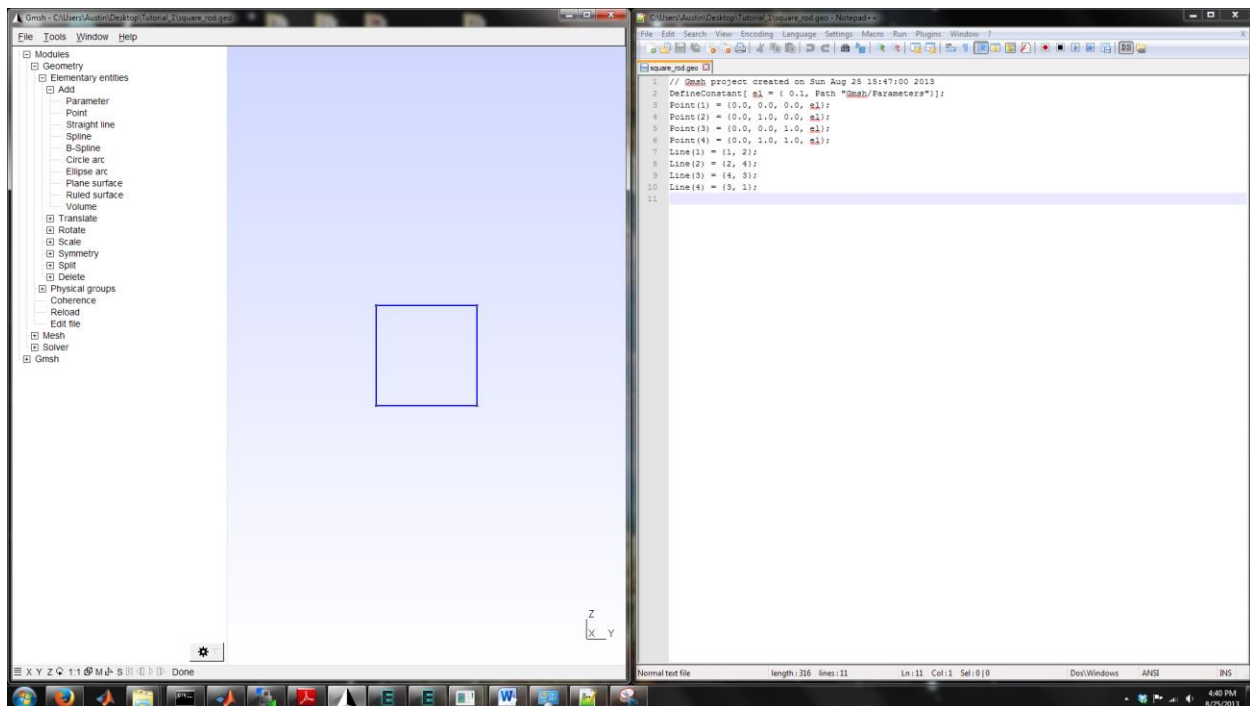
The screen should show the change (a square appears in the viewing area:



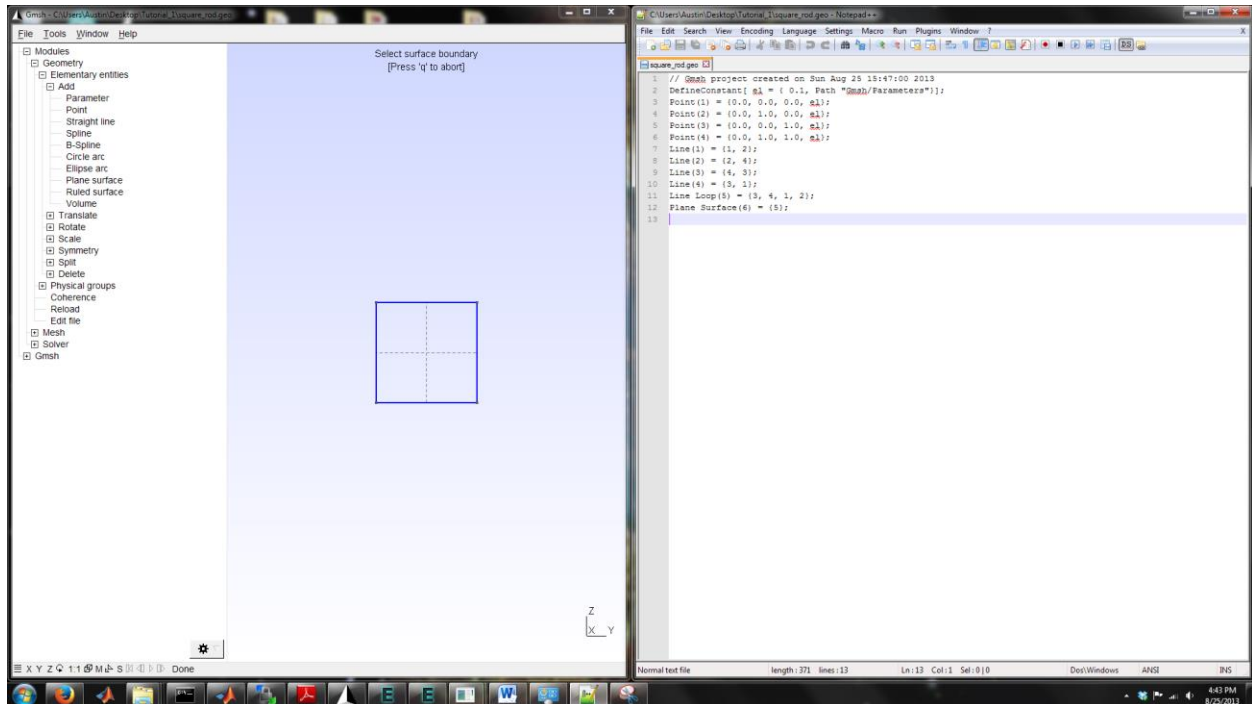
Leaving the .geo file alone, we return to work with the GUI. We want to connect the four points to make an enclosed square. Select 'Geometry' -> 'Elementary elements' -> 'Add' -> 'Straight line'

Every time a tool like this is selected, notes appear in the top of the viewing area about how to use the tool. In this case, we select each set of points we wish to connect. Hover over the bottom left point - a hover-text box should appear giving the name and location (Point 1{0 0 0}). Click this point to select it as the first point on the straight line. notice the point turns red to indicate it is selected. Select the point directly to the right of it (Point 2). A line appears between the two. Another line can be added in the same manner. Connect the four corners with three more line segments to create a square. press  $\square$  to leave the 'Add' 'Line' function.

Click in the notepad++ screen. A window appears asking to update the file; accept the update. The square and its definition should be visible.



Now, we want to create a surface of the loop of lines we have created, otherwise the software does not recognize this as a 2D shape but merely a connection of 4 lines. To accomplish this, select 'Add'-'>'Plane Surface'. click any of the lines in the loop - the square will turn red indicating the boundary of the surface that will be created. press  $\epsilon$  to accept this as the border. The surface is indicated by dotted lines crossing the center of the square. Update the .geo script as before (click in the notepad++ window, accept update).



There are two new additions: a Line Loop and a Plane Surface have been created. Because several geometries can be defined by a closed loop of lines, a Line Loop is a unique entity. The Plane Surface simply calls on such a Line Loop to define the border of the Surface. Notice that the Points are numbered 1 through 4, and all other entities are numbered consecutively. GMSH requires that the identifying numbers for each piece of geometry be unique, except for Points which maintain their own identifier set. Keep this in mind if adding geometry in the script file as it will cause an error.

We will now create a mesh to introduce the concept of structured meshing later. To mesh the surface we have created, simply click, in the Tree, 'Modules' -> 'Mesh' -> '2D'. A triangular mesh appears on the surface made earlier. If the mesh color bothers you (mine was yellow) go to the upper toolbar and follow

'Tools' -> 'Options'

select 'Mesh' in the list on the right

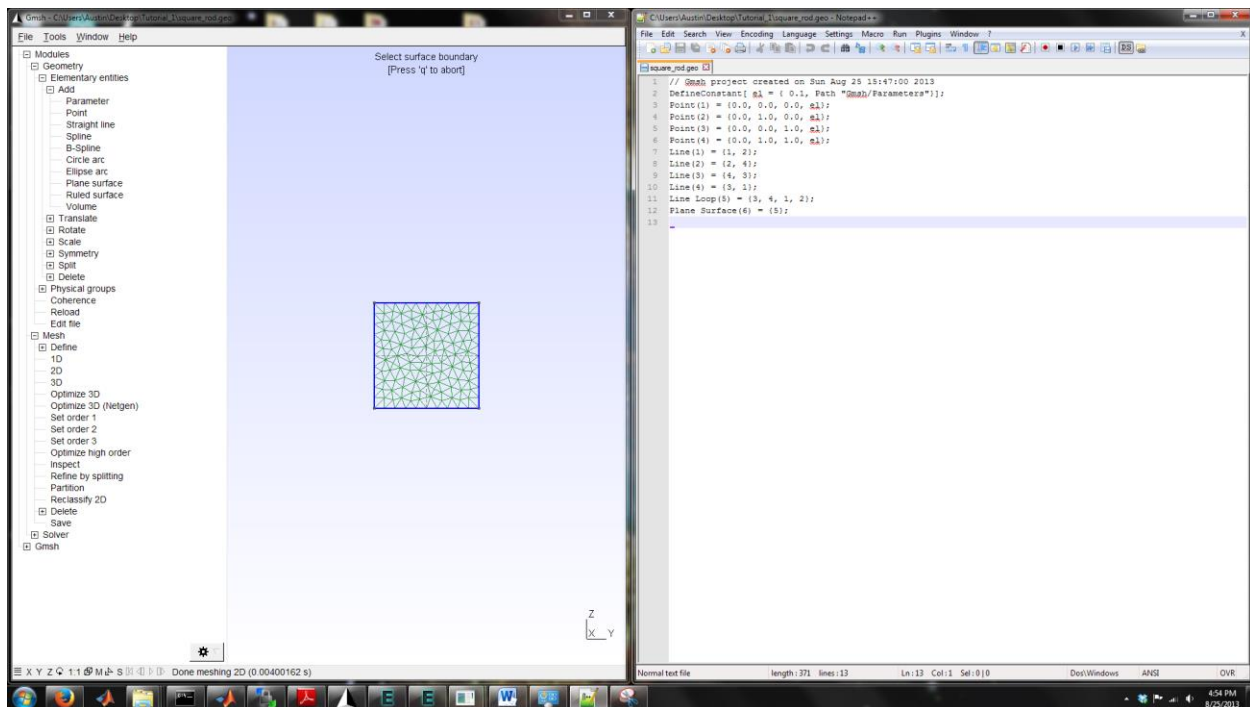
select the 'Color' tab

scroll down in the menu to 'Six', click on it

choose a color from the color wheel.

Click 'OK' and Exit.

Notice that each line of the square has ~10 nodes on it. This is because we set the 'Prescribed Mesh Element Size at a Point' to 0.1; since  $l = \text{length of the line}$ ,  $l/10$  yields 10 evenly spaced nodes on each line.



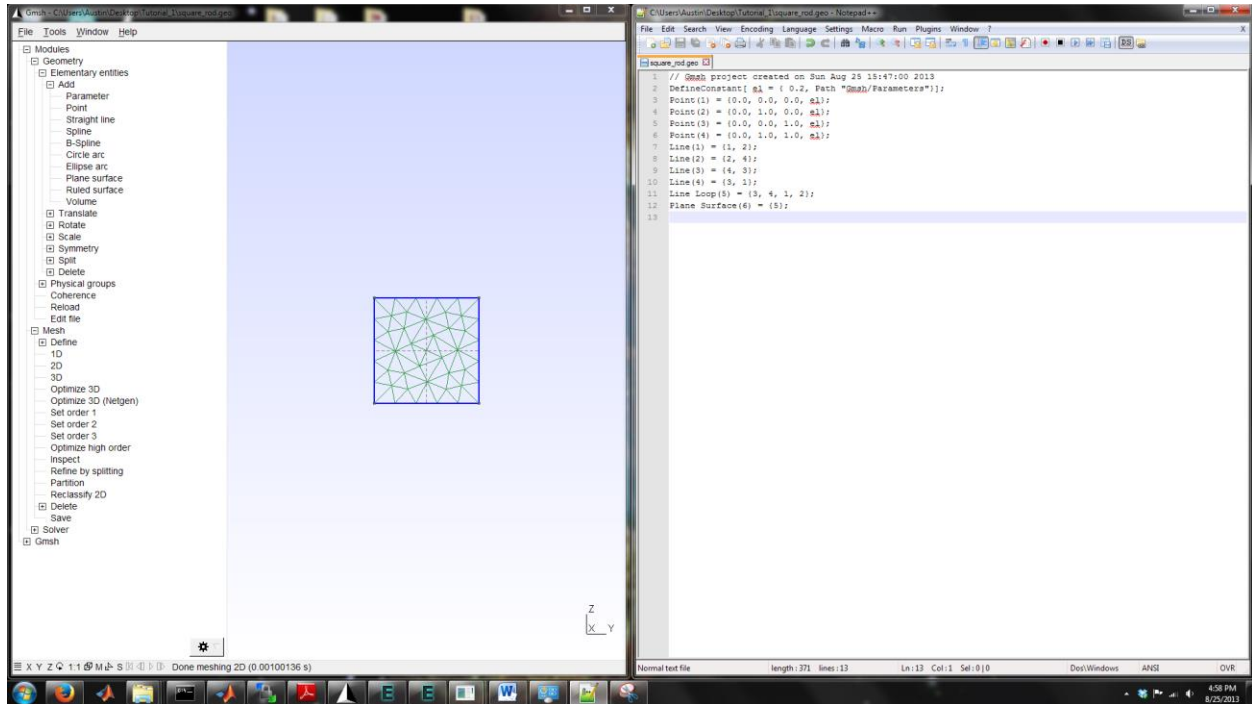
This can be changed by editing our variable `el`. change `el` in the script to 0.2,

```
DefineConstant[ el = { 0.1, Path "Gmsh/Parameters"}];
```

becomes

```
DefineConstant[ el = { 0.2, Path "Gmsh/Parameters"}];
```

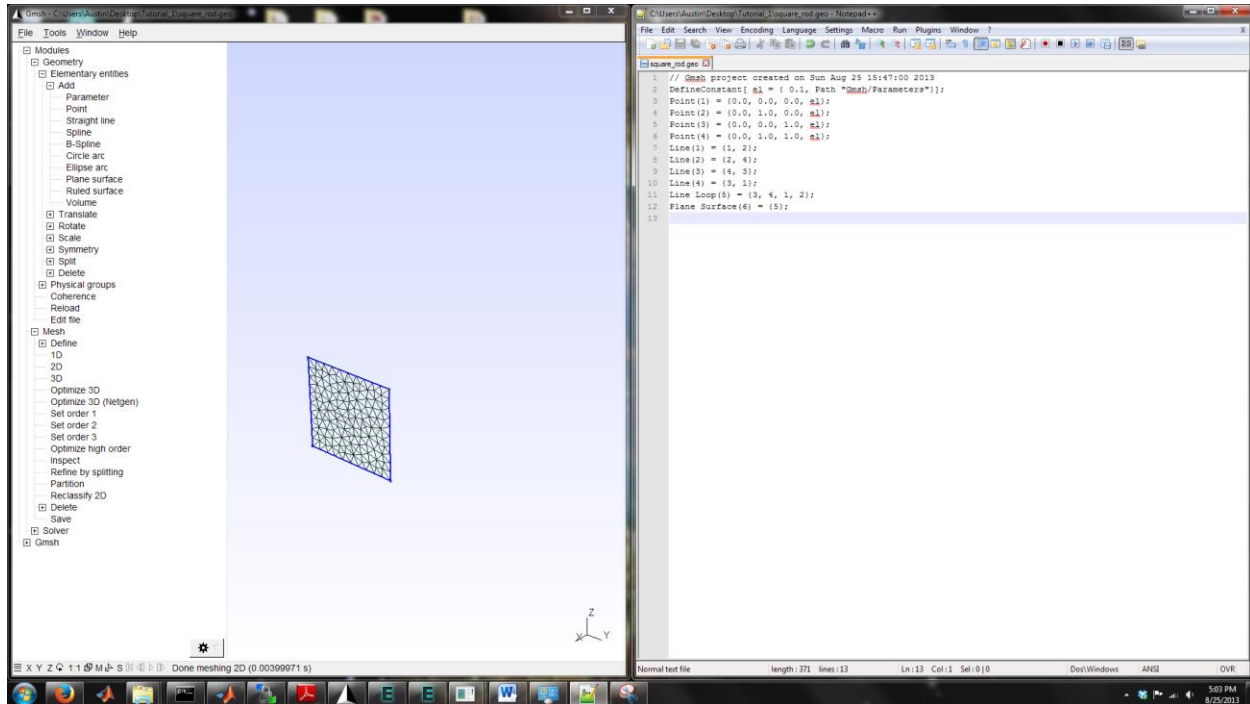
Save the .geo file, reload the part, ('Geometry'->'Reload') and re-mesh ('Mesh'->'2D'). The mesh now has the new spacing.



For now, let's revert back to the original size 0.1, following that same procedure.

We now wish to extend our square out into a rectangular solid (or prism). This could be accomplished by adding more points, lines, and surfaces, and then filling the surfaces with a body. However, we will do this in a more intelligent way, with some important effects on the Mesh.

Drag the square such that it is seen at an angle (using the left mouse button)

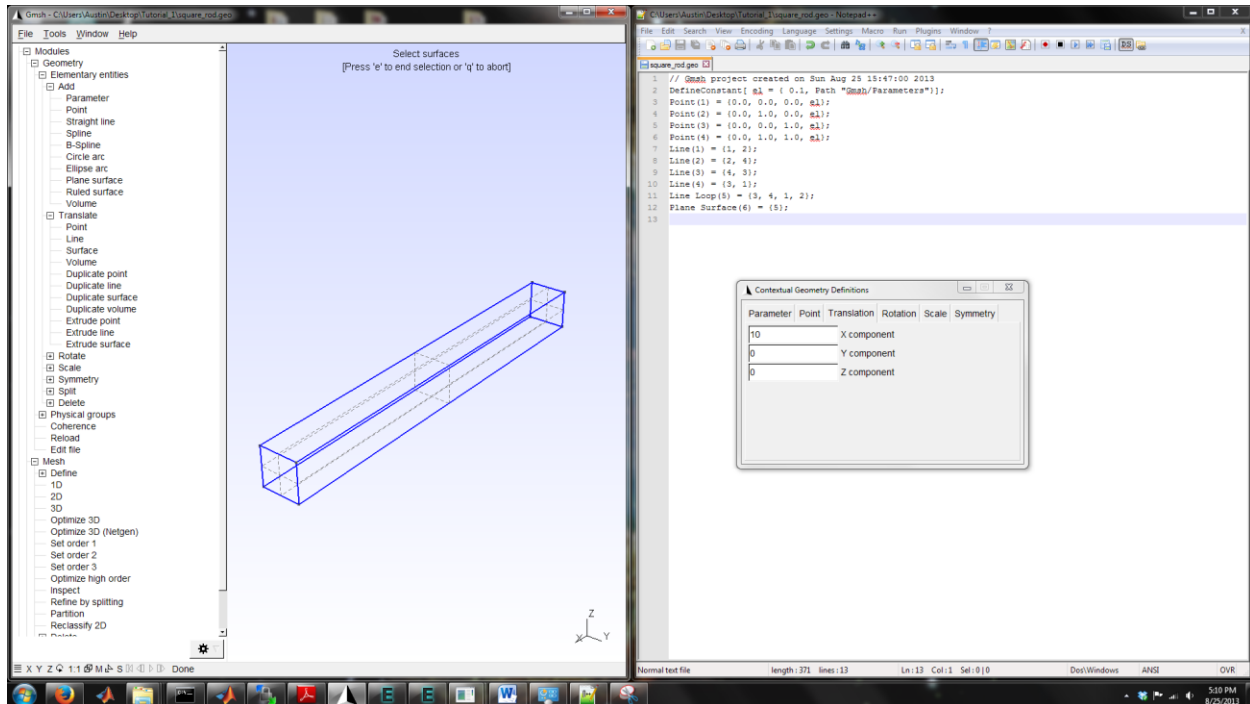


Reload the geometry to clear the Mesh ('Geometry' -> 'Reload')

We will use the 'Extrude Surface' command to create the geometry by expanding our square surface along the x-axis 10 units. Select

'Geometry'-'>'Elementary entitites'-'>'Translate'-'>'Extrude Surface'

The CGD window appears again. Select the surface we made (Plane 6) by clicking the crossed dotted lines in the center of the square. The dotted lines turn red to indicate that plane is selected. Set the X-component of the translation (in the CGD window) to 10. Make sure the Y and Z components are both zero. This will extend our square 10 units in the X direction. Click in the Geometry Viewer and press  $\epsilon$  to finish the extrusion. Your screen should look something like this:



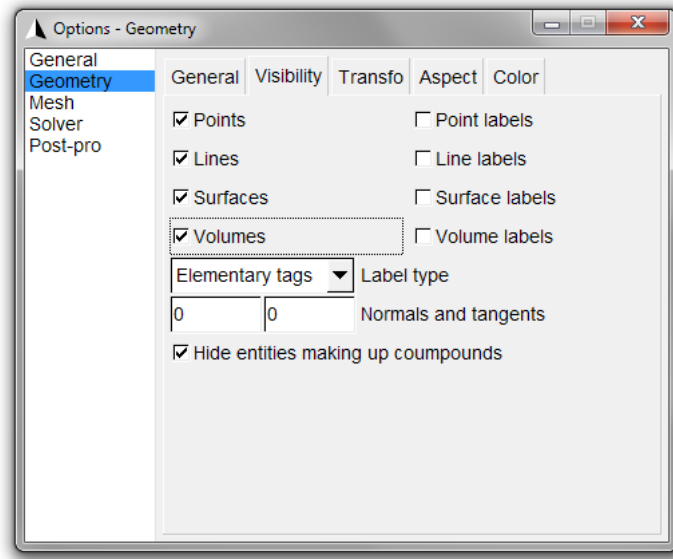


We can now mesh the part as before. However, it is now a 3D shape. To see that a solid has been created, change another setting in the Options:

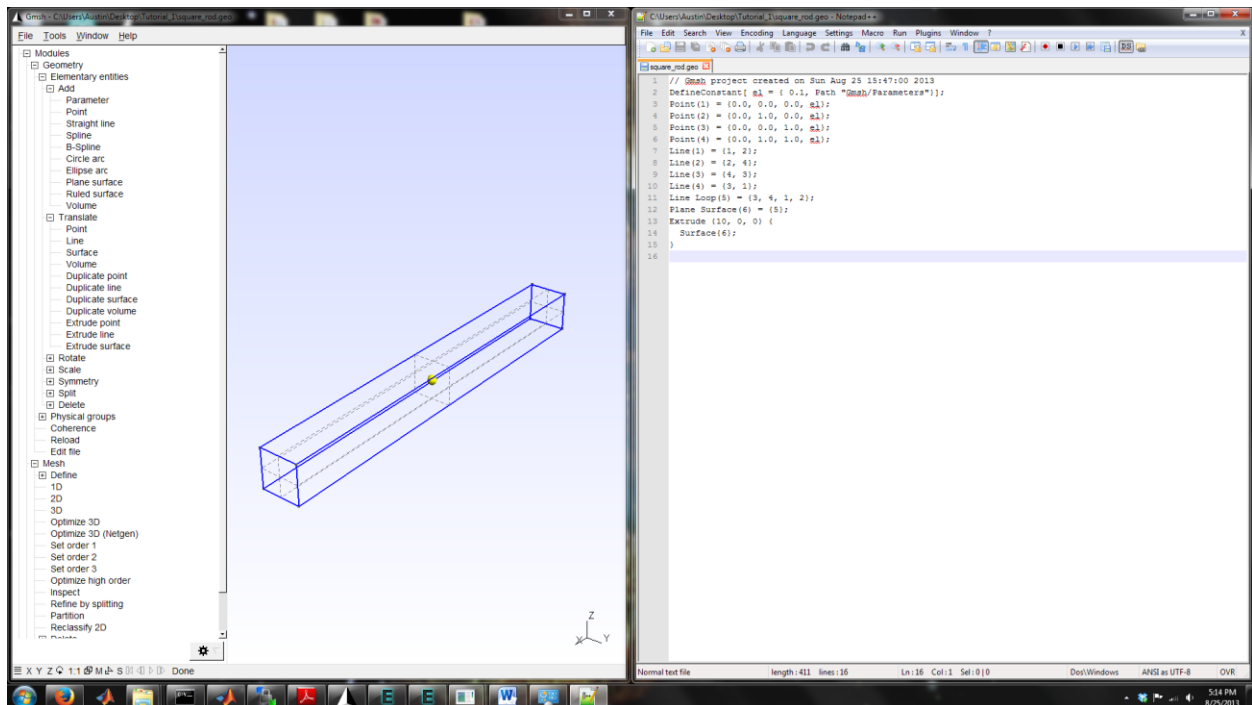
'Tools' -> 'Options'

Select 'Geometry' from the List

Check the box next to Volumes



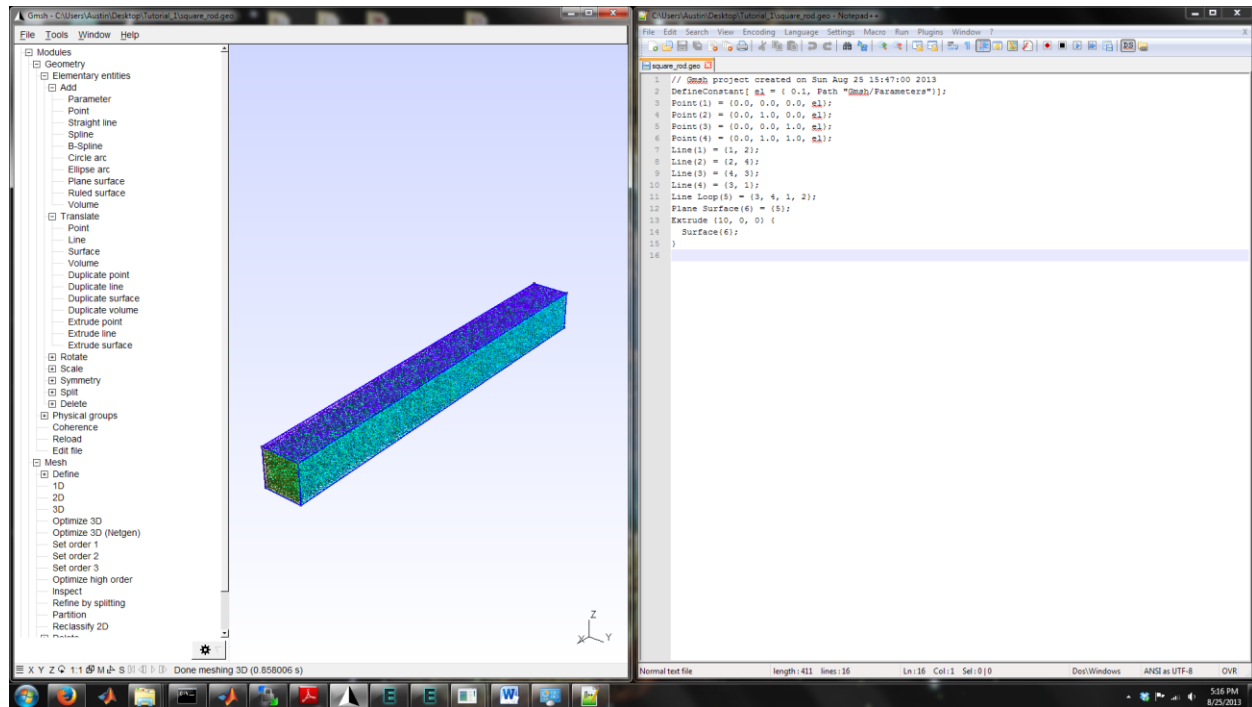
A sphere indicating the volume center appears in the Viewer:



Now, mesh as before except select 3D mesh (which performs 2D then 3D meshing immediately following)

‘Mesh’->’3D’

The part is meshed using the 3D version of a triangle called a Tetrahedron



I mentioned we would use the sweep to our advantage. A lot is about to change so look at the current mesh and get a feel for how random and unstructured it is. While the results of the analysis would be accurate (with this mesh), there are an excessive number of elements due to the random nature which are unnecessary (this mesh has ~48000 elements). We now make some major amendments to the .geo script.

first, we will define our original 'square' surface using a swept line. Delete all the code after the first two point definitions. The code should be similar to this:

```
// Gmsh project created on Sun Aug 25 15:47:00 2013
DefineConstant[ el = { 0.1, Path "Gmsh/Parameters"}];
Point(1) = {0.0, 0.0, 0.0, el};
Point(2) = {0.0, 1.0, 0.0, el};
```

Save the changes and reload the geometry.

In the lower left toolbar, click 'X' and '1:1' to see what is left; the original two points appear.

Draw a line between the two points ('Geometry'-'>'Elementary entities'-'>'Add'-'>'Straight line') as before

Now, we will extrude the line upward to create the square surface.

'Geometry'-'>'Elementary entities'-'>'Translate'-'>'Extrude line'

Select the line

Set Z Component to 1, the X and Y components to 0, and press e to finish the shape. The square has returned to the screen.

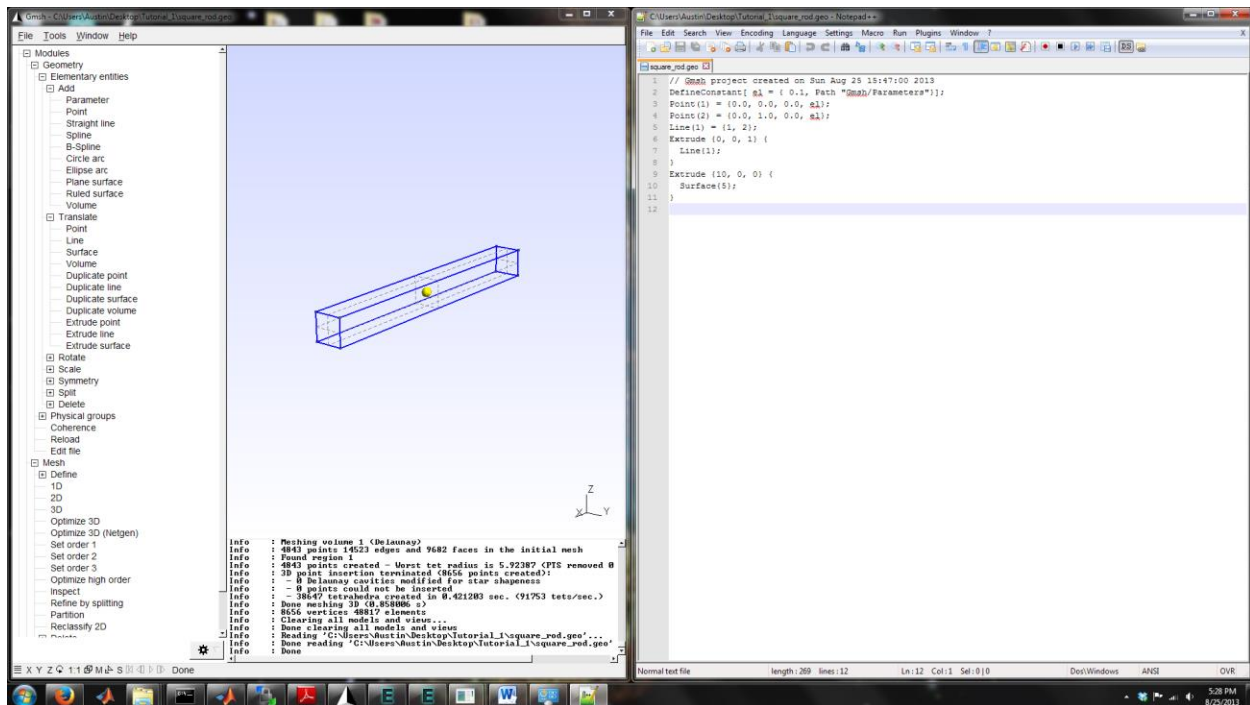
Extrude the square as before

'Geometry'-'>'Elementary entities'-'>'Translate'-'>'Extrude surface'

Select the square surface center-plane

Set X Component to 10, the Y and Z components to 0, and press e to finish the shape. The rod has returned to the screen (may need to rotate the view to see it).

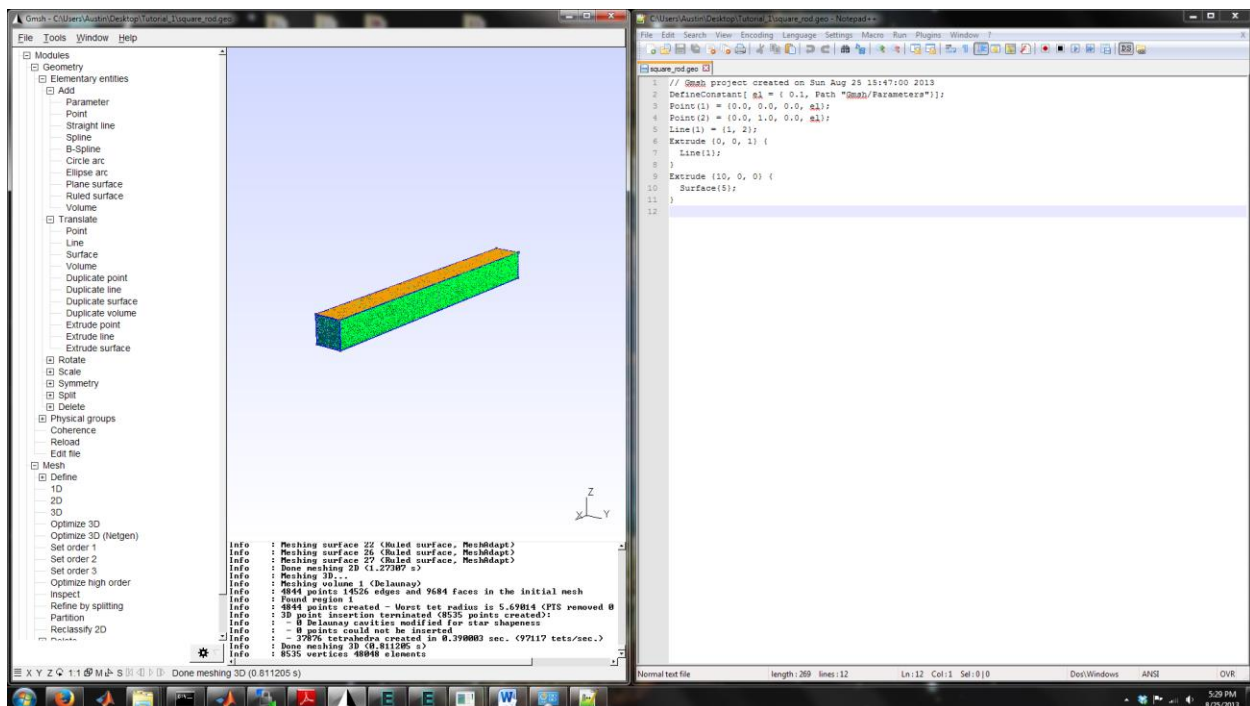
Check the .geo file to see the new code (click in the notepad++ window and accept the update)



See how much more elegant this method is?

Now we mesh the part as before ('Mesh' -> '3D')

With nearly identical results to the previous mesh (~48000 elements):



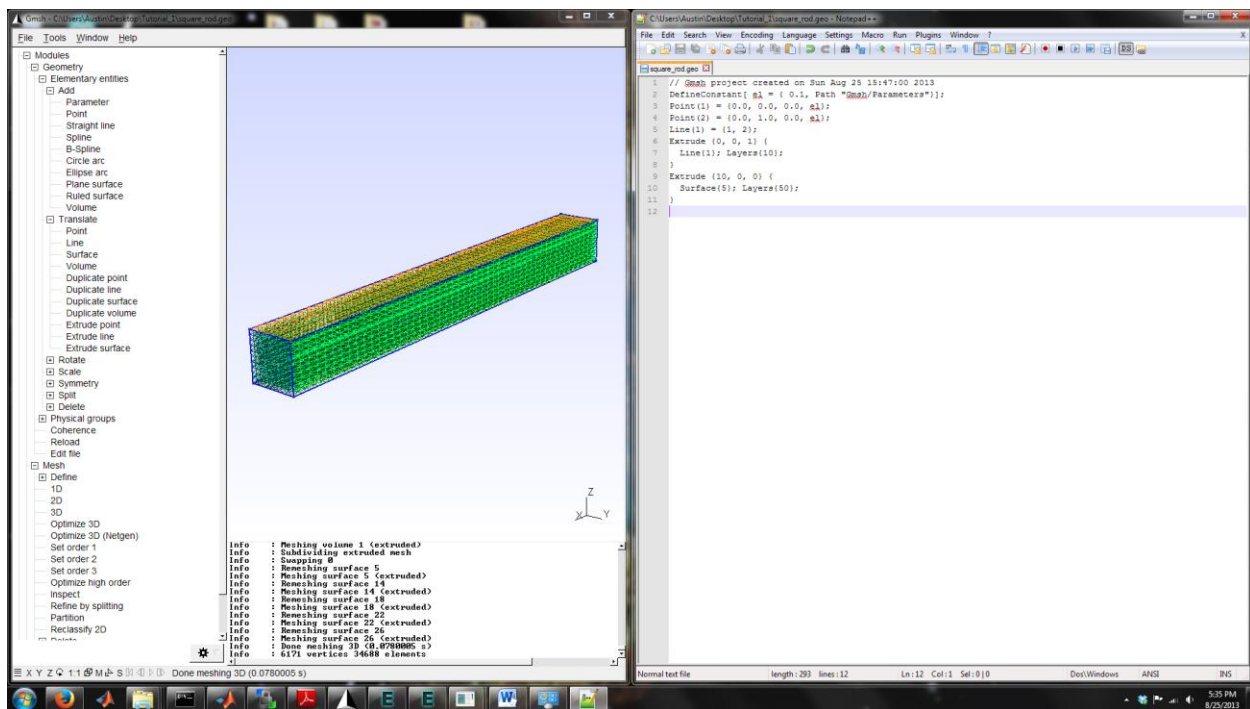
The .geo script is now amended. We are going to define 'Layers' in each extrusion. this means that the mesh will be layered along the extrusion and not random as it is currently. The value in 'Layers' determines the *number* of layers along the length (not the *size* of the layers).

```
Extrude {0, 0, 1} {
  Line{1};
}
Extrude {10, 0, 0} {
  Surface{5};
}
```

Is modified to:

```
Extrude {0, 0, 1} {
  Line{1}; Layers{10};
}
Extrude {10, 0, 0} {
  Surface{5}; Layers{50};
}
```

This will create a structured mesh along each of the extrusions. Save and reload the .geo file. Mesh the part and review the difference.



The number of layers can be modified as desired for the analysis. Here, the values given above will be kept.

Reload the geometry to clear the screen before continuing.

One difference between Commercial FEA and these methods is that both the geometry and mesh are understood by the solver. Here, that is not the case as only the mesh will be sent to the Solver. In order to apply boundaries and otherwise ‘click on’ the part after it is imported to the solver, ‘Physical Groups’ of nodes must be created. Typically, these groups will include the volume (to apply material properties later) and the surfaces to which Boundary Conditions (BC’s) will be attached. In the case of Elmer, it is visually more useful to assign All surfaces to some group, even if no BC will be applied there.

We will first add a Physical Volume (the entire part):

Select ‘Geometry’->‘Physical groups’->‘Add’->‘Volume’

Select the Spherical volume indicator in the center of the part. It will turn red to indicate it has been selected.

Press e to complete selection

Next, we choose the two ends individually (they will have BC’s)

Select ‘Geometry’->‘Physical groups’->‘Add’->‘Surface’

Select the Center of the Leftmost end surface (Ruled Surface 5 in the Hover-Text)

Press e to complete selection

Select the Center of the Rightmost end surface (Ruled Surface 27 in the Hover-Text)

Press e to complete selection

Select the other four surfaces around the perimeter of the part. These will be in the same group.

Press e to complete selection.

Open and view the additions to the .geo file:

```
Physical Volume(28) = {1};  
Physical Surface(29) = {5};  
Physical Surface(30) = {27};  
Physical Surface(31) = {26, 22, 18, 14};
```

These are the selectable groups once the mesh is exported.

We now re-mesh and save the mesh.

Choose 'Mesh'-'>'3D' to revive the mesh.

In the 'File' Menu on the upper toolbar, select 'Save Mesh'. No further input is required, the mesh has been saved in ...*\Tutorial\_1\square\_rod.msh*

The GMSH portion of this tutorial is complete, congratulations!