

# Homework 8 - CS 6210

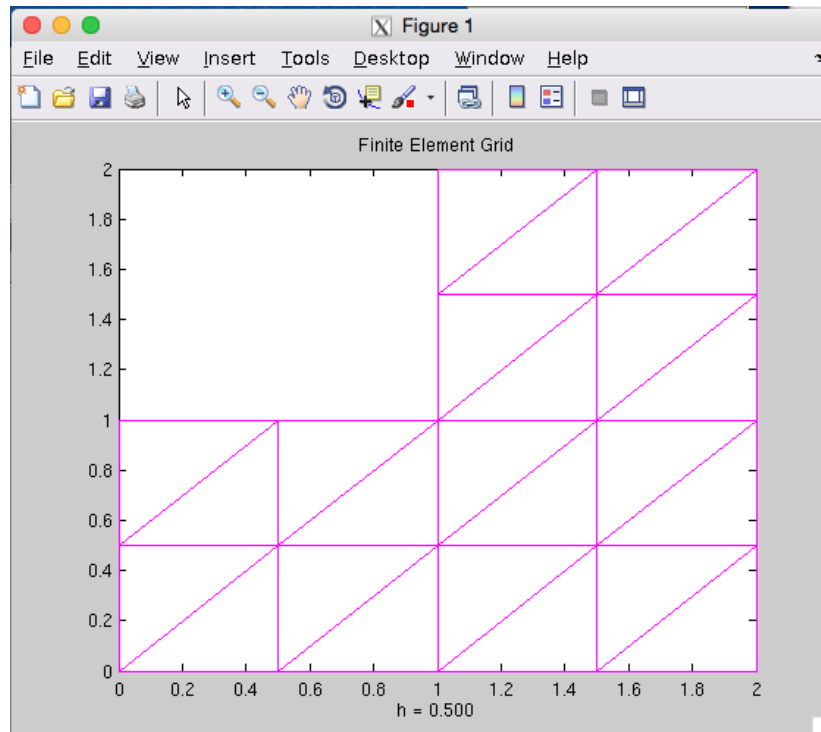
Ally Warner

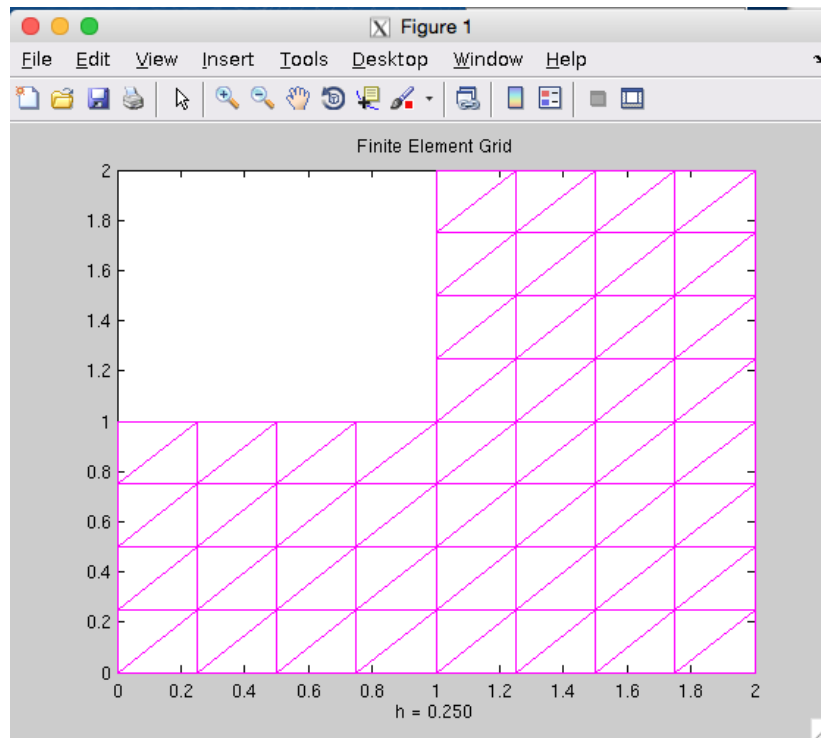
December 19, 2014

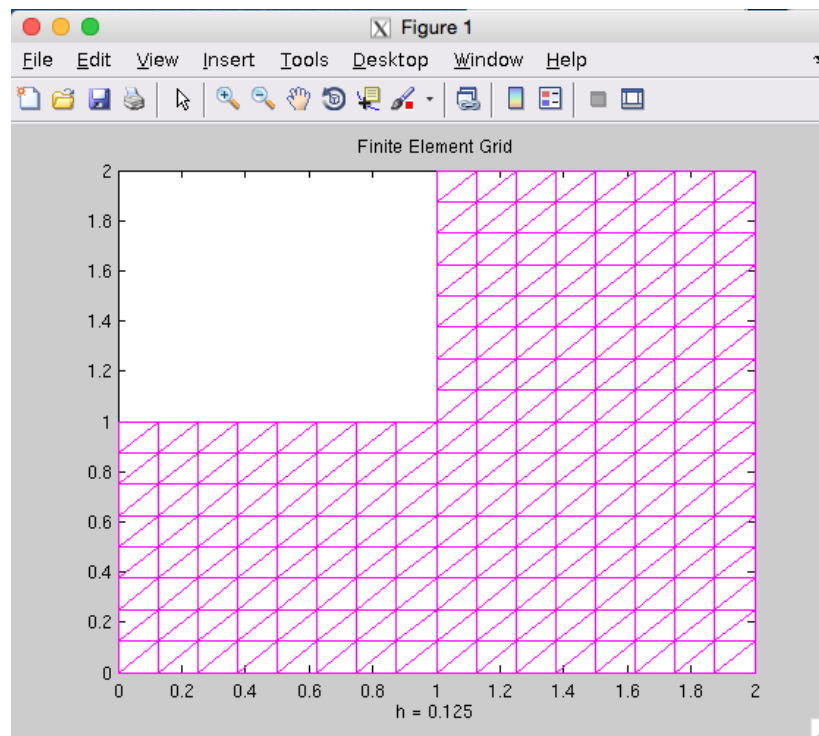
## 1 Grid

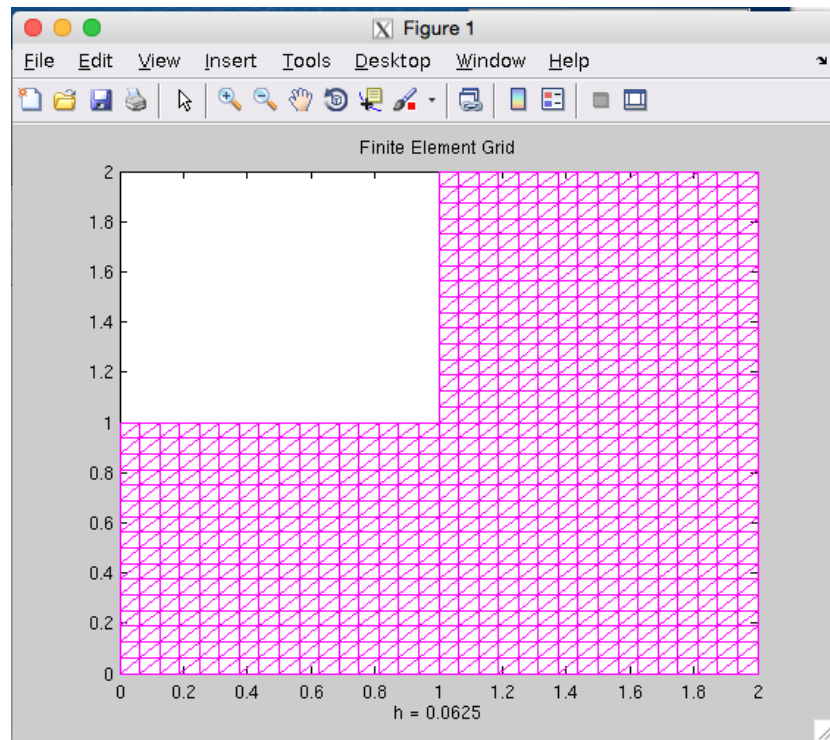
Create triangular finite element grids of different resolution by dividing the finite difference grids squares you created in Homework 7 into two triangles.

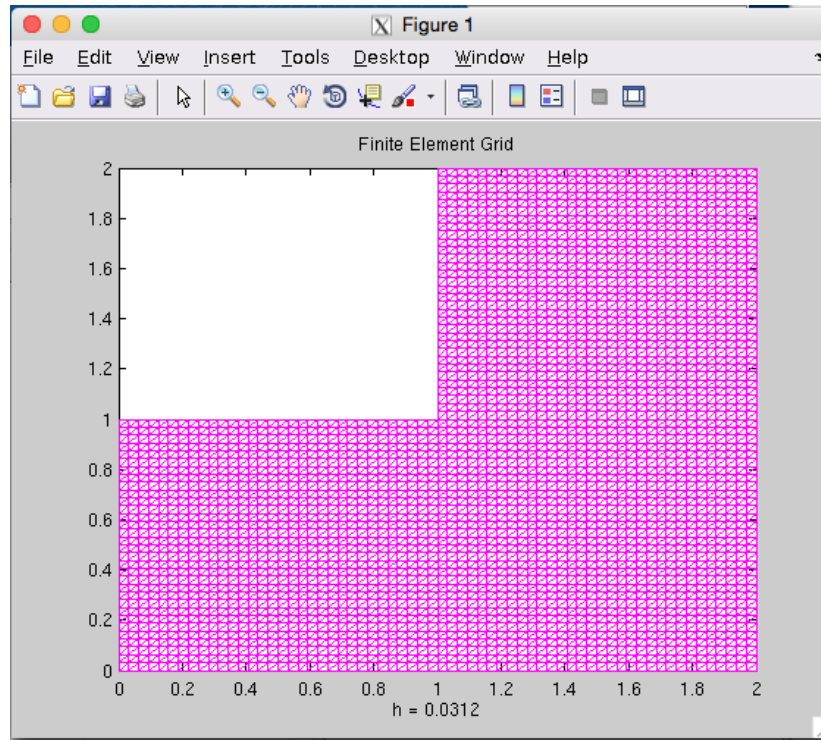
To crease my triangular finite element grid, I calculated the size of the grid based on the input of  $h$ , which is the distance between nodes on our grid. I used my method from homework 7 to find the physical positions of the nodes in relation to the grid. I then made a mesh by saving the nodes that made up each element in a matrix. I used triplot to visualize my grid. The following figures show my triangular mesh with different values of  $h$ .





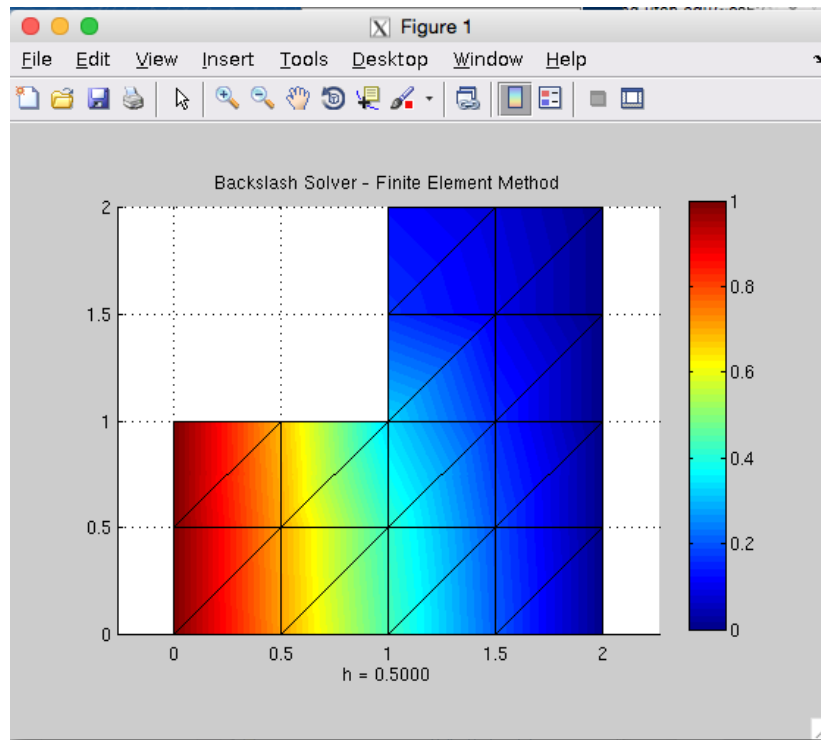


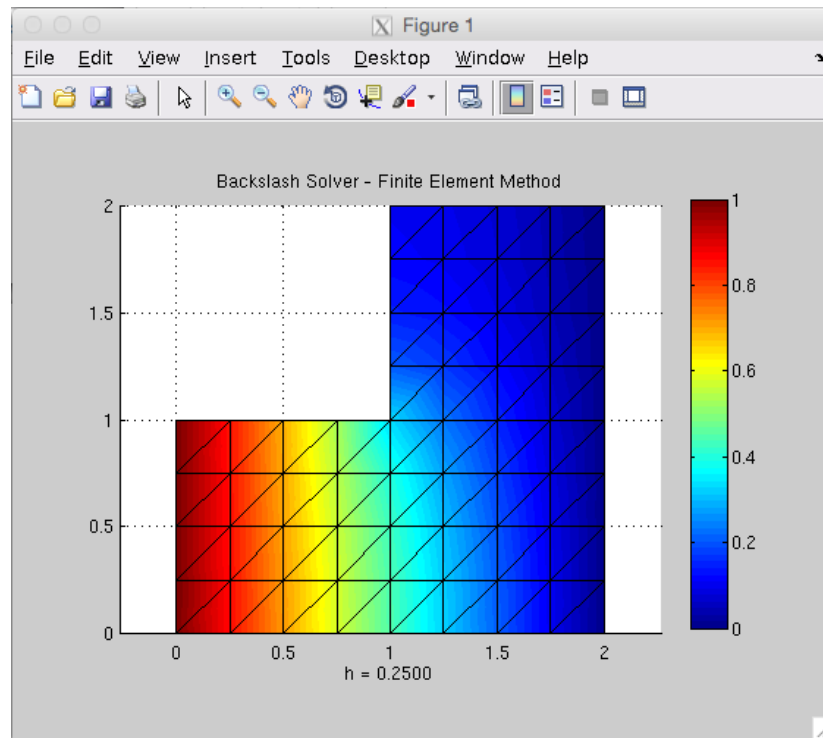


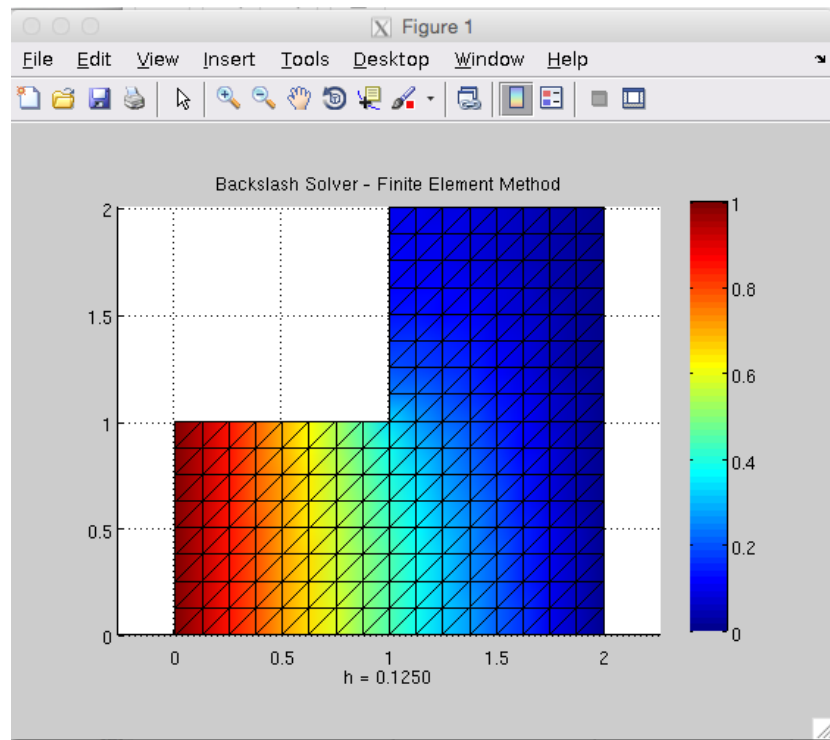


## 2 Finite Element Method

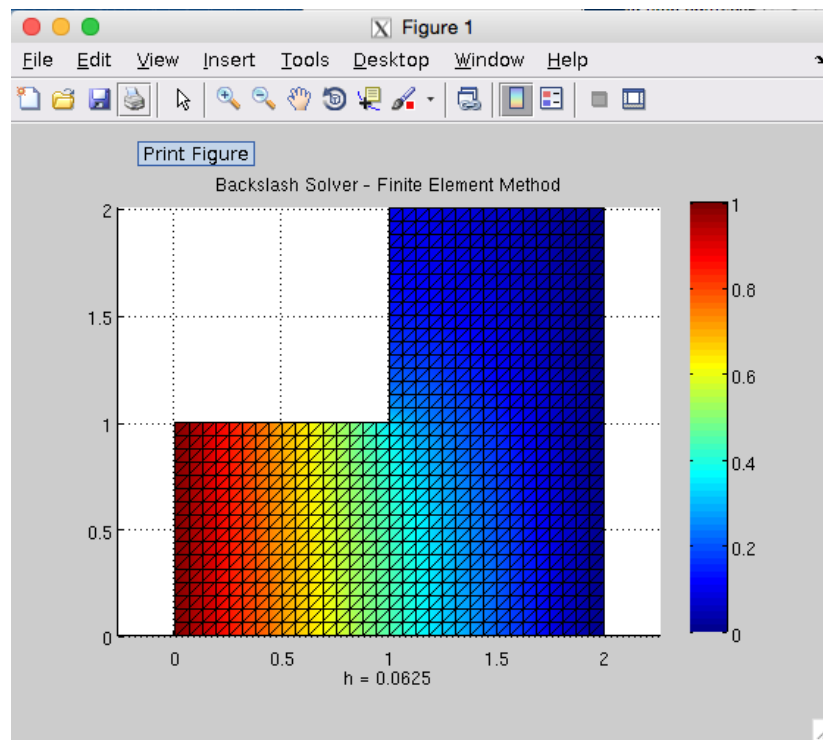
Writing a finite element method was a bit more complex than writing the finite difference method code. First, we take our positions of nodes and the mesh information and we find the physical position of the vertices of each element. From this we can make our local matrices which need first order triangular elements. After we have made these, I save their row information into three separate matrices. I then make the global matrix by looping throw these row matrices. After the global matrix has been made, the boundary conditions are applied and there is a linear system to solve! The solved linear system is shown in the following figures at different values of  $h$ .

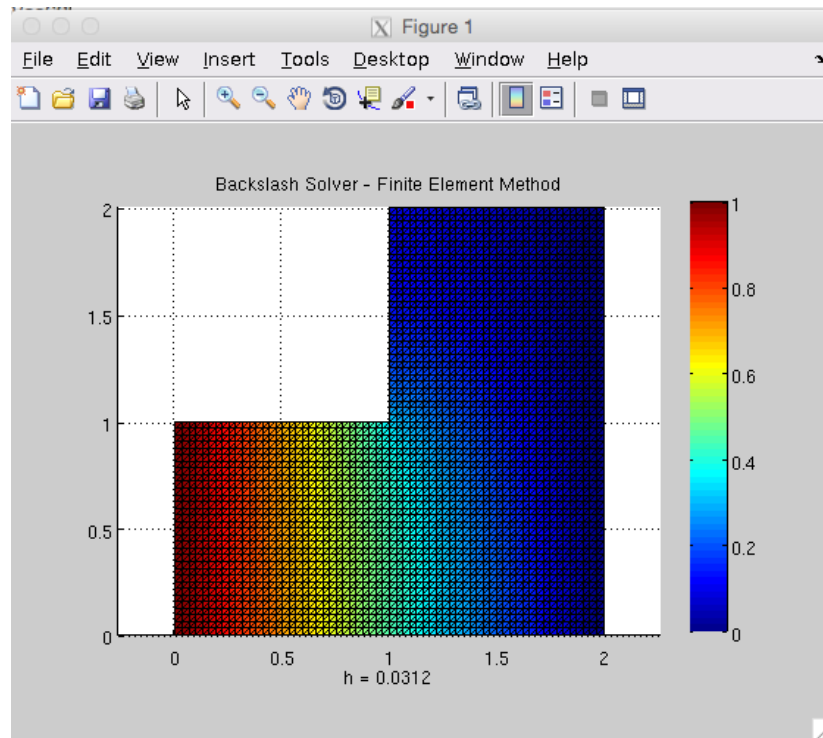










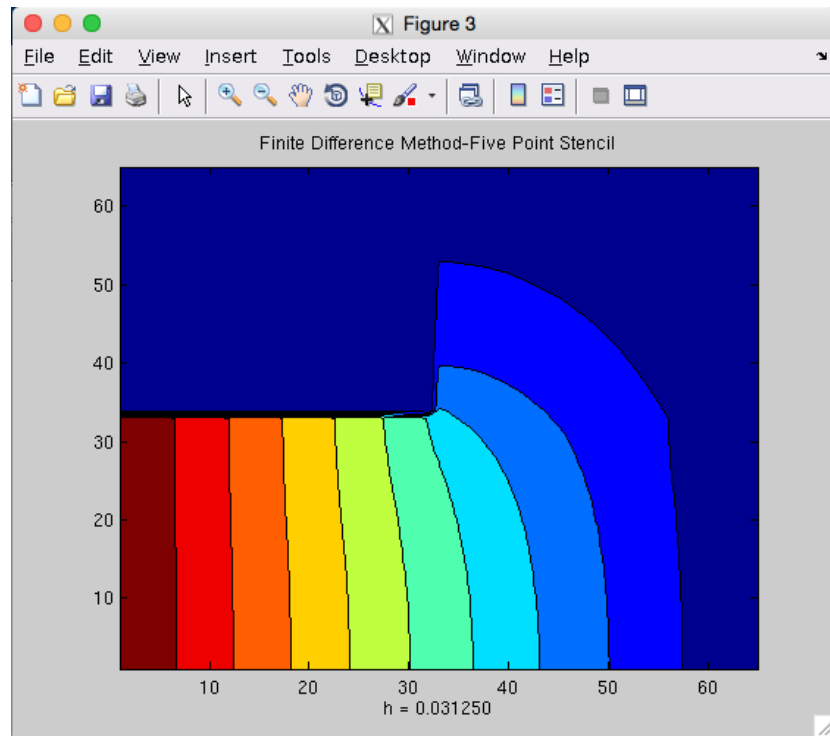


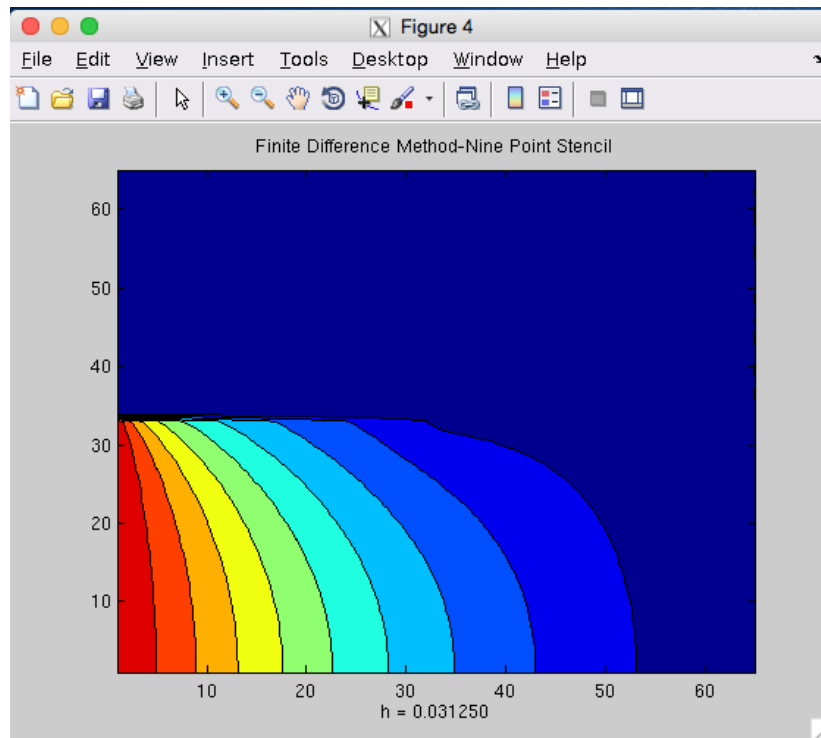
### 3 FEM and FDM Comparison

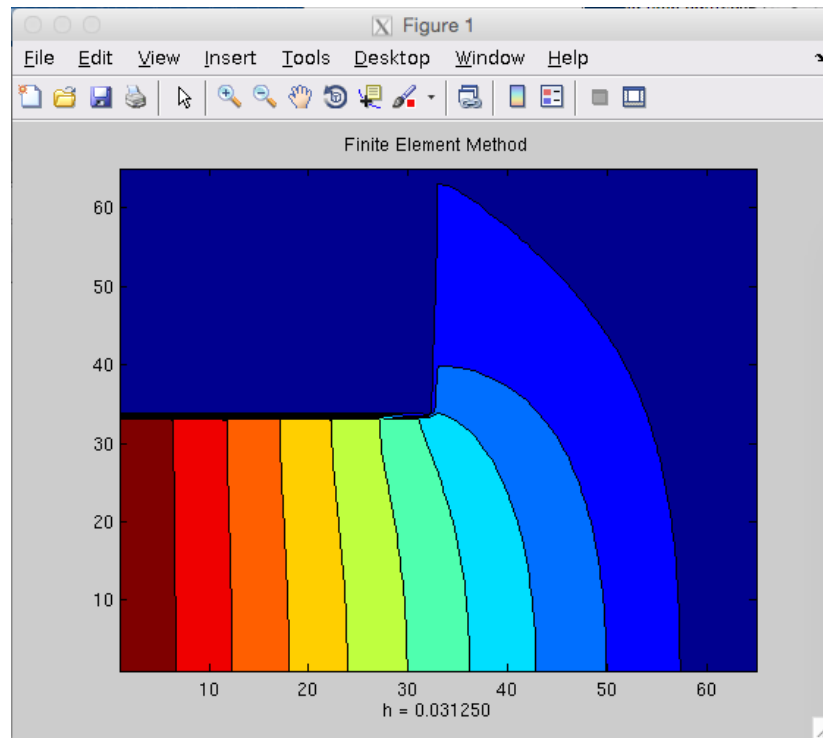
I solved the steady state heat problem using the finite element method and the finite difference method and compared their run time, using the backslash solver, and their solutions via contour plots. The cpu time comparison is shown in the table below.

<b>Backslash Solver h = 0.3125</b>	<b>FDM Five Point</b>	<b>FDM Nine Point</b>	<b>FEM</b>
<b>1</b>	3.86	3.55	9.36
<b>2</b>	3.85	3.57	9.04
<b>3</b>	3.75	3.54	9.0
<b>4</b>	3.82	3.63	8.98
<b>5</b>	3.74	3.56	8.99
<b>6</b>	3.8	3.57	9.0
<b>7</b>	3.81	3.55	8.98
<b>8</b>	3.86	3.63	9.01
<b>9</b>	3.83	3.55	8.9
<b>10</b>	3.77	3.54	9.01
<b>Average</b>	3.809	3.569	9.027

As you can see, the finite element method is about three times slower than finite difference method. Although this method is slower, it is more accurate. This can be seen from the contour plots in the following figures.

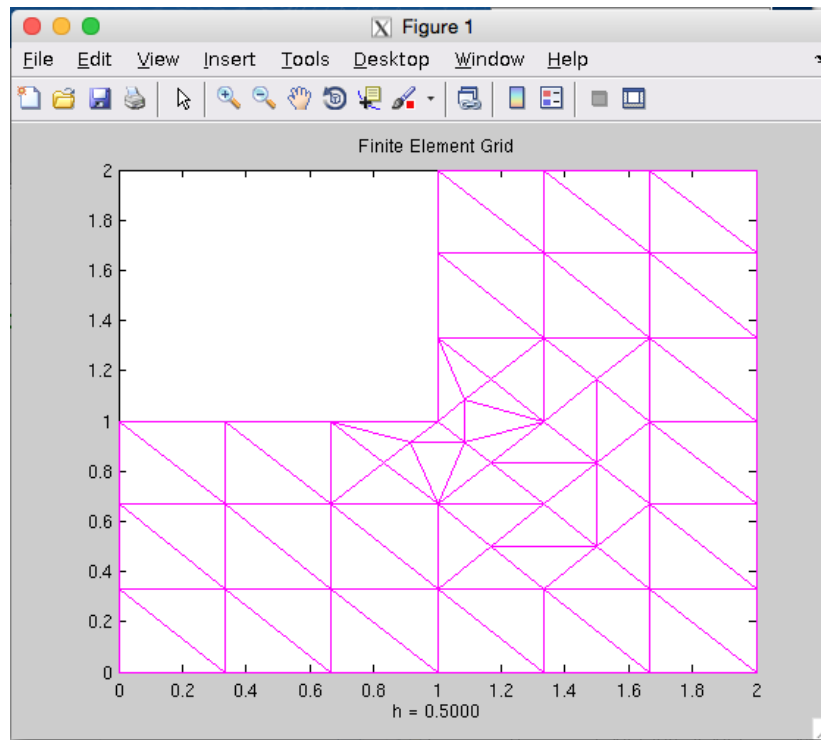


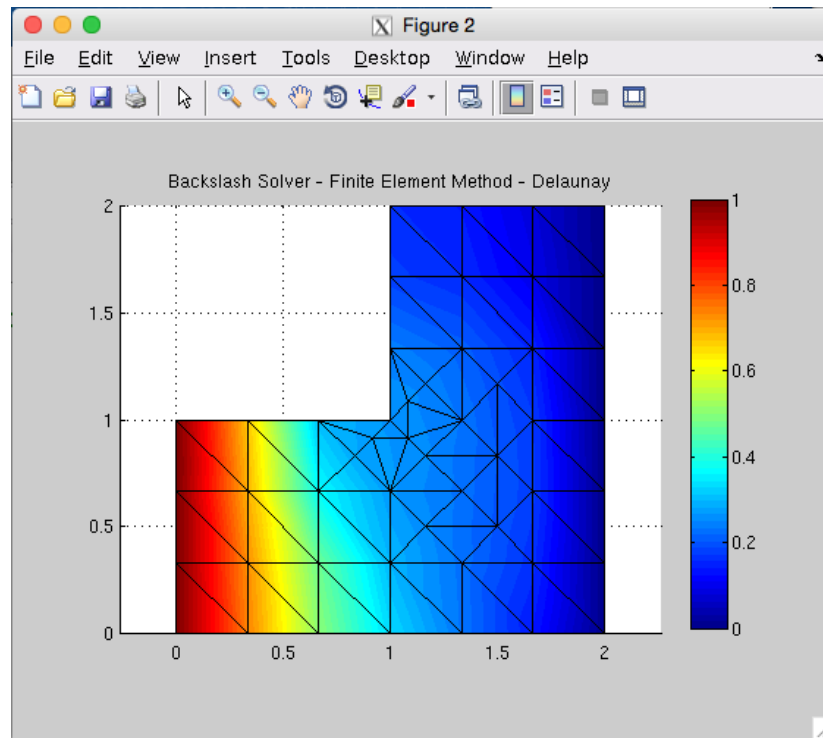




## 4 Delaunay

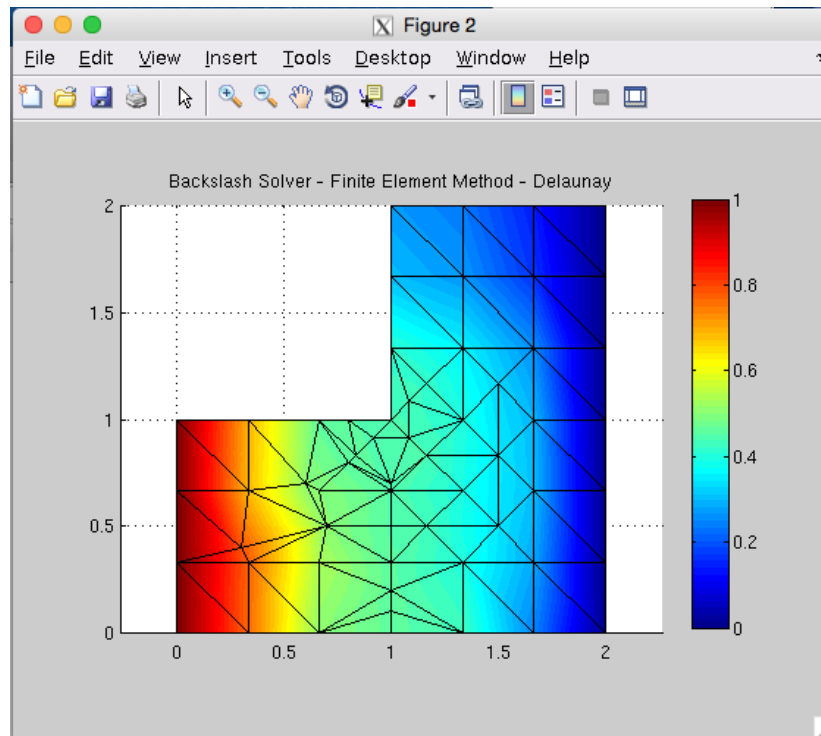
Last, we get to work with an unstructured grid. After loading in the delaunay file and removing the extra points, I ran it through my original code and had a grid and a solution shown in the following figures.



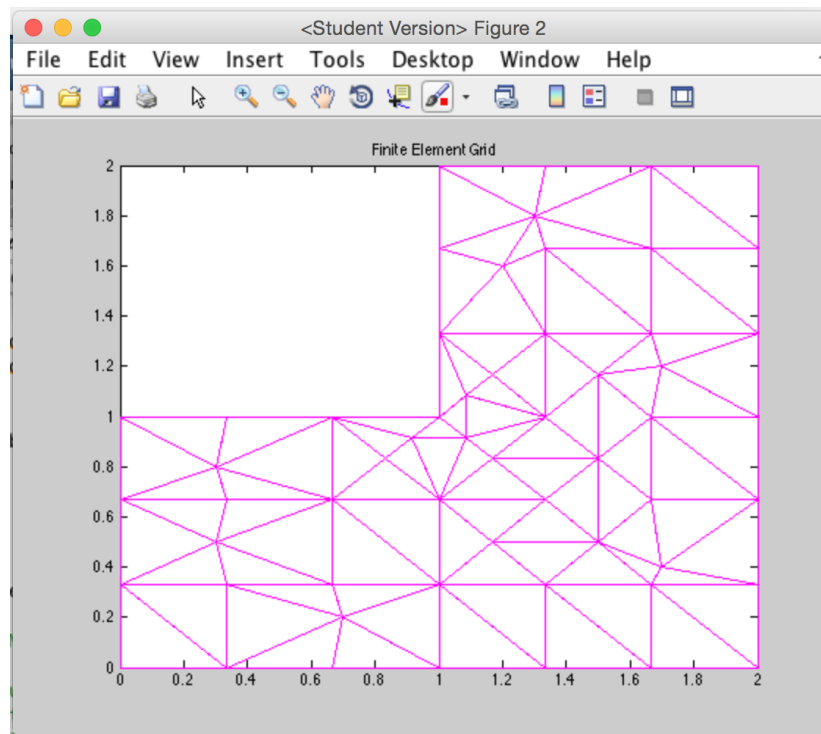


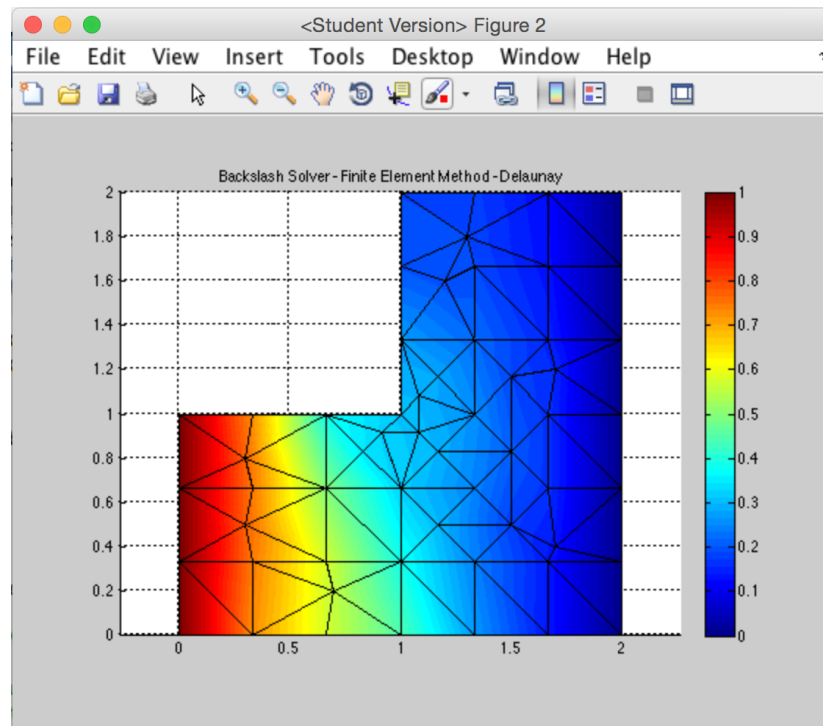
In comparison to our structured grid, we know that this is quite the right answer. So adding nodes to the grid can help the solution to become more accurate. Figure out how adding nodes in specific places of the grid affects the solution was a bit tricky at first. The figure shown below shows my first pass at making the solution more accurate.





We can see that this isn't a very accurate solution at all. So instead of adding many points at once, I just added one point at a time to see how to "push" the solution to be more accurate. My second attempt at making the solution is much better and shown below with the corresponding grid.





\*All of the code for this assignment is submitted with the assignment.