ELEC 4700 Modelling of Integrated Devices

Assignment 2: Finite Difference Method

Due: February 23, 2020

Professor: Tom Smy

Student Name: Rudi Hidvary

Student Number: 101037815

Introduction

The problem that is being solved in this report is how voltage will naturally be distributed over a 2-D plate having fixed voltages applied at the boundaries. When the voltages are initially applied there is a transient state where the voltage is being distributed which then settles to a steady state. Using Laplace’s equation and boundary conditions, a network of resistors will be set up and the equation will be solved under differing boundary conditions, resistor sizes, and resistor values. These will determine the nature of the problem so that the simulation can handle a plate with changing resistance values and changing boundary values.

Using the forward difference method, and not an iterative method, the solution is achieved faster and more accurately than the iterative method. The electrostatic problem of solving Laplace’s equation is difficult to realize in computer simulations using continuous solutions. Using a powerful tool such as MATLAB, a numerical solution can be achieved that will very closely resemble the same continuous value problem. To check to ensure that the simulation is giving the correct response, an analytical solution to one of the cases will be solved. Even so, this analytical solution is merely an infinite sum of a Fourier series which is not perfectly realizable with a computer, so it is flawed but still gives reasonable results.

This report outlines the way in which the solution to this numerical problem was formulated using MATLAB and how electric fields and currents are distributed given varying conductance. Examples of each simulation will be shown with explanation on what is occurring and how the simulation was coded to achieve this result.

Part 1: Uniform Resistance Laplace Solution Compared to Analytical Solution

Before any simulations are run, the basic setup of the simulation needs to be explained. Using the forward difference method, matrix equations for each node of the plate will need to be established so that the voltage values can be calculated. Using the formula GV = B, where G is the conductance matrix, B is the boundary values, and V is the voltage of that node, the rearranged equation V = G\B can be used to determine the voltages. Since each node of the plate needs an equation, these matrices can become very large so sparse matrices are used to speed up the calculations in MATLAB.

The initial case that will be solved is essentially a 1-D problem that can be solved using a plate for simplicity later on in the code. This case is for when the applied voltage V0 forces X = 0 to this voltage and holds X = L at ground for the boundary conditions and then solves the Laplace equations. This numerical solution is an example of a plate with constant resistance over the surface and how the voltage is distributed along the length of the resistor. Figure 1 shows an example of this problem in a 2-D plot using a 5-volt supply as V0.

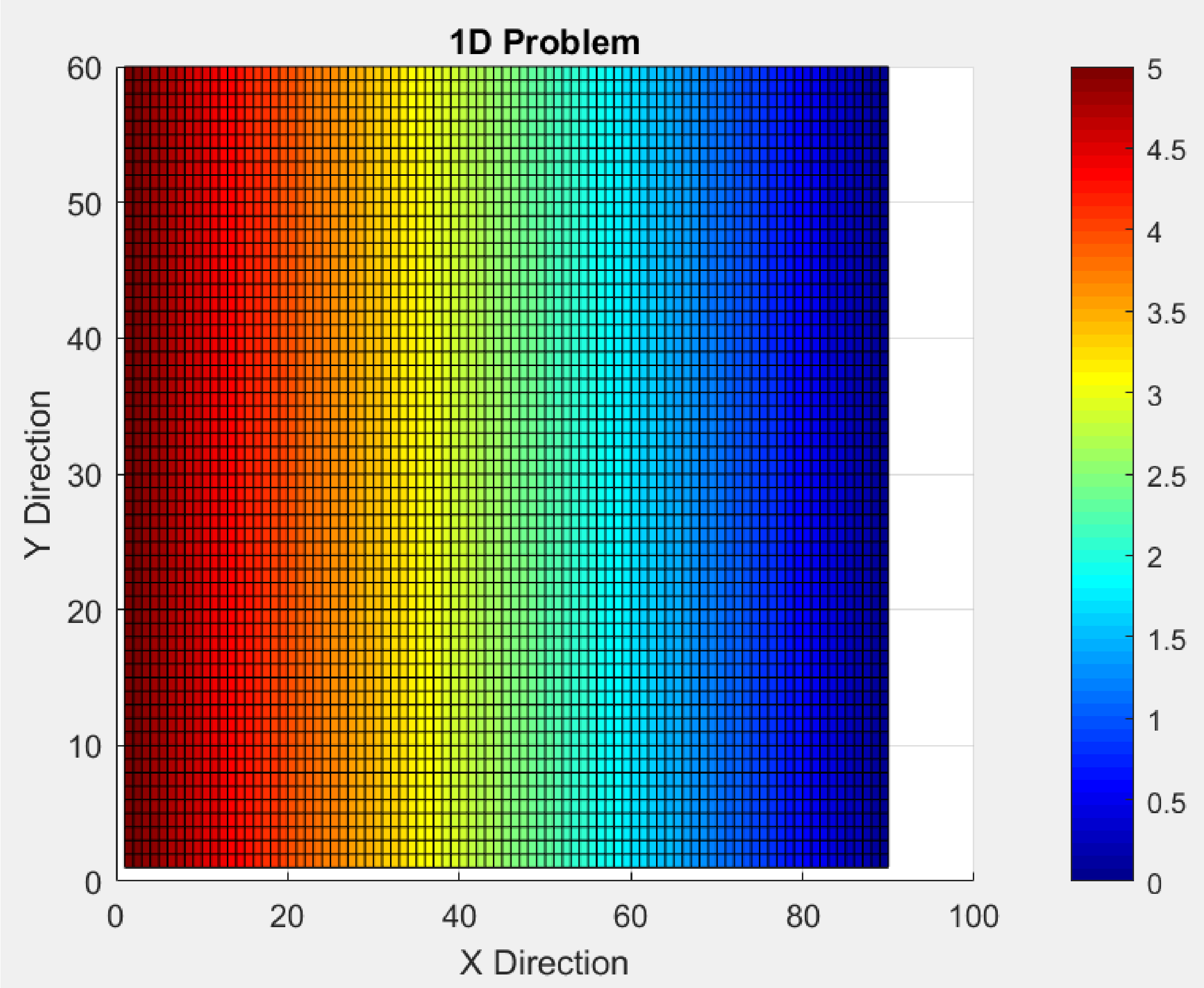

Figure 1: Laplace Solution to 1-D Problem

This solution can essentially be viewed as a 1-D problem because the way the voltage is distributed over the plate is the same for every value of Y.

The next case that was solved was a plate where two ends are held at V0 and the other two ends are held at ground to give the effect of a voltage distribution that looks like a saddle. The same code was used for this case with the only changes being that the B vector different values for the boundaries and the G matrix changes accordingly by placing a value of 1 at the (n,n) space in the matrix. Figure 2 shows the 2-D plot of the voltages which are represented by colour and Figure 3 shows the graph in 3-D where the applied voltage is again 5 volts.

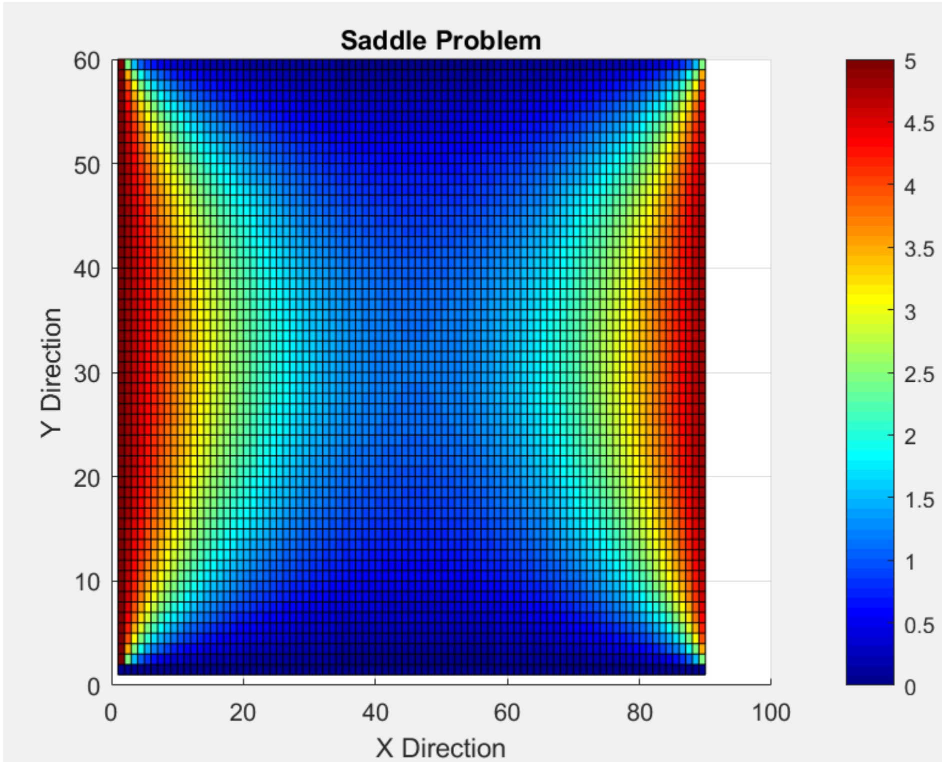

Figure 2: 2-D Saddle Problem Solution

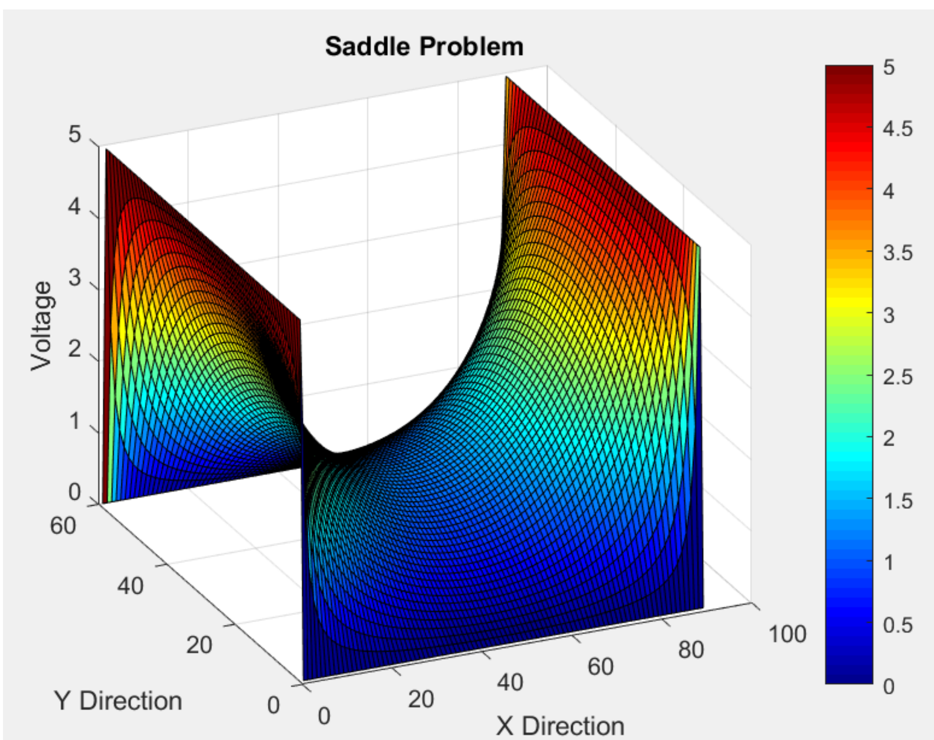

Figure 3: 3-D Saddle Problem Solution

This numerical solution can be compared to an approximated analytical solution which is provided by Griffiths “Intro to Electrodynamics 3e” which is a solution calculated using an infinite sum of sinusoidal terms. This solution is still only an approximation of the true analytical solution because MATLAB can only calculate a finite sum, but it can be done very well so that it is extremely close to the true analytical solution. Two factors decide the resolution of the analytical solution. The first being the number of terms summed up, more terms means a more accurate solution up until a point where the number becomes so small that MATLAB can no longer handle it. The second factor is the size of the meshes used in the analytical solution with a higher number giving a more accurate result. To show the effect these have, 4 plots will be made using a combination of small and large mesh sizes, and a small and large number of summed n terms. Figure 4 shows the different combinations for the analytical solution.

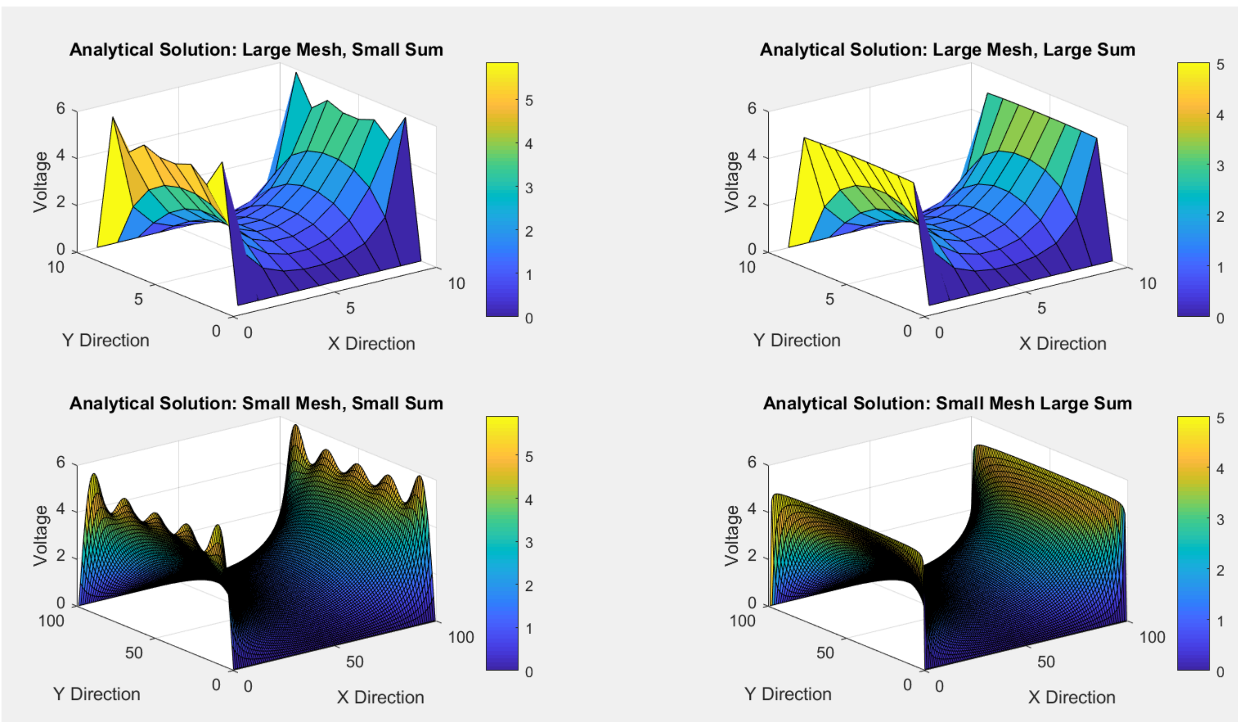

Figure 4: Analytical Solution and Variants

From these results, it can be seen that the more meshing that is used, or the smaller each mesh becomes, the analytical result becomes more and more accurate. As well as the more iterations of sums that are done, the closer the answer comes to be the true analytical solution. As described before, there is a limit due to the computational power of MATLAB and the device it is run on, so to avoid an unnecessarily long simulation, a large number of sums and meshes needs to be used such as in the bottom right hand corner of Figure 4.

The differences between the numerical and analytical solution are minimized as the mesh size of the numerical solution is increased and as the mesh size and summing number are increased in the analytical solution. Because the analytical solution is an infinite sum, which MATLAB must truncate to solve, the numerical solution can be used as a reasonable approximation to the correct answer if the resolution of the mesh size is sufficient.

Part 2: Non-Uniform Resistance Laplace Solution, Electric Field, and Current Density

Now that the groundwork is laid for a numerical solution to Laplace equation with uniform resistance, a simulation can be run where the plate that is being solved can have varying resistance across it. To solve this problem, the code can be modified so that the conduction matrix is included when solving for the voltage at a node. To do this, the G matrix is modified to have the values of the conduction matrix calculated for each node and all adjacent nodes, then this is used to solve for the voltage. Once this is accomplished, the electric field and the current can be found easily, and then plotted to see the effect of the bottleneck.

For the purpose of this report, the bottleneck region will be half as wide as the simulation space and it will also be half as long as the simulation space. Figure 5 shows the conduction map of the simulation so that it is easy to tell what is occurring in the bottleneck and normal regions where the conduction has a value of 1 everywhere but the bottleneck which has a conduction value of 0.01.

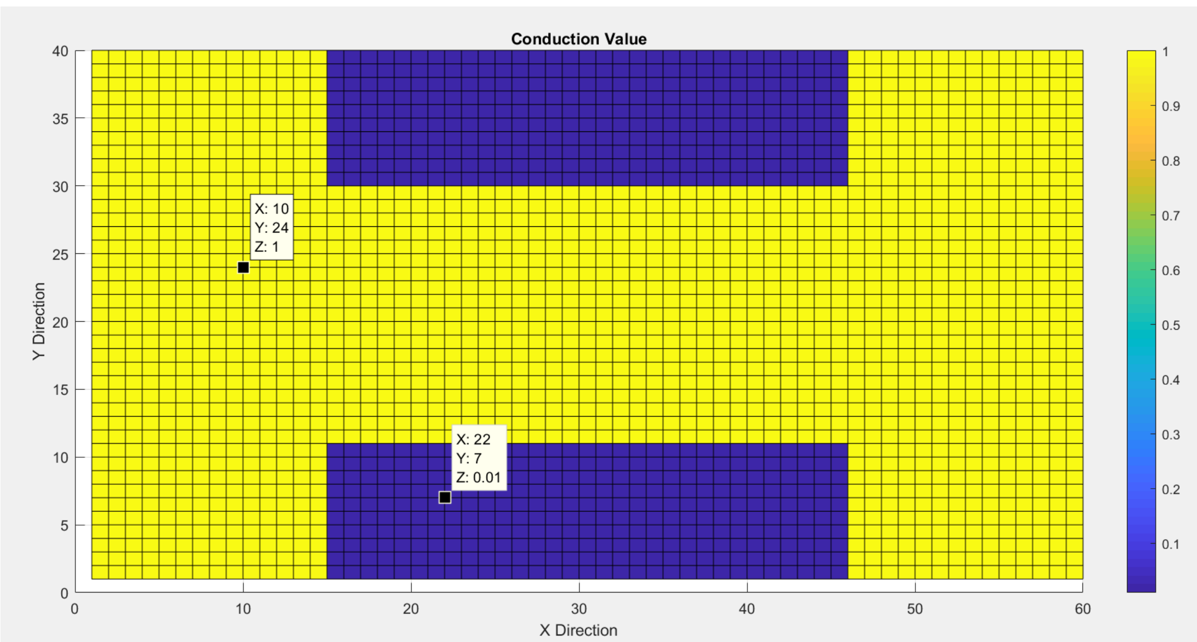

Figure 5: Conduction Map

Using this conduction map, the value of the voltage over the plate can be found. This is done as described above and can be seen in Figure 6.

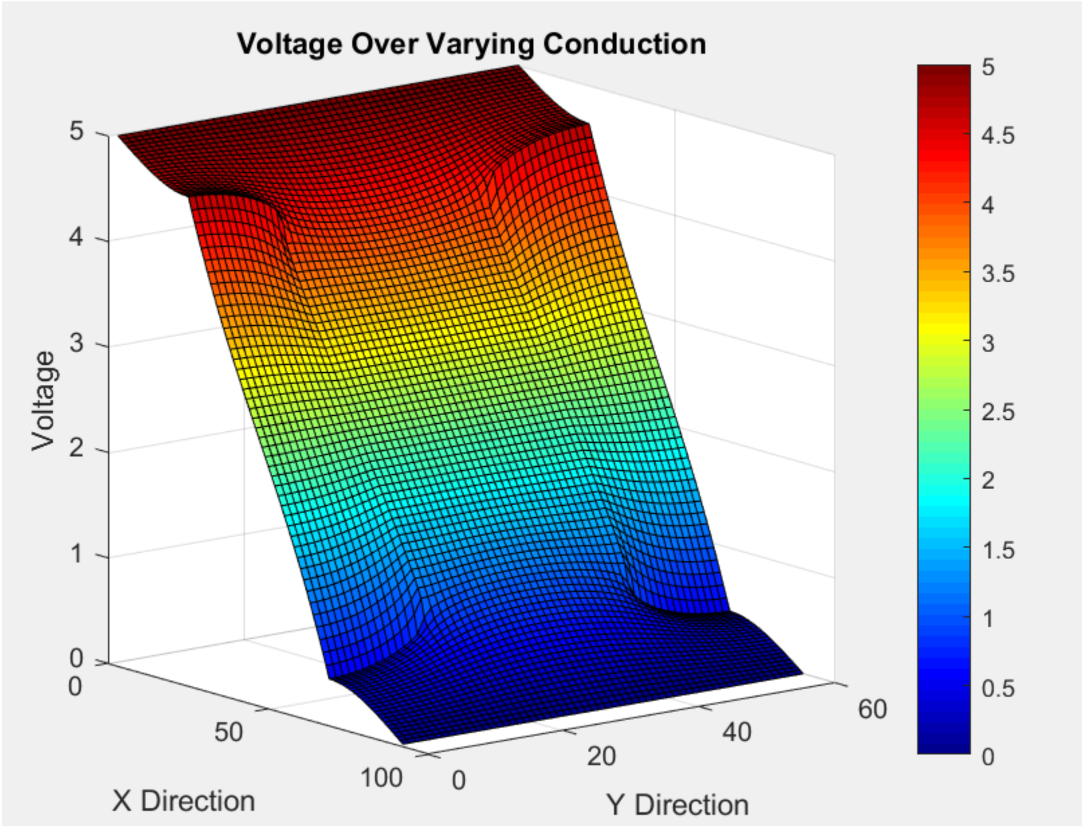

Figure 6: Conduction Varying Voltage

Using this plot, the electric field can easily be found by taking the gradient of the voltage then multiplying that by negative one. This can be seen in Figure 7 where the arrows represent the direction of the electric field.

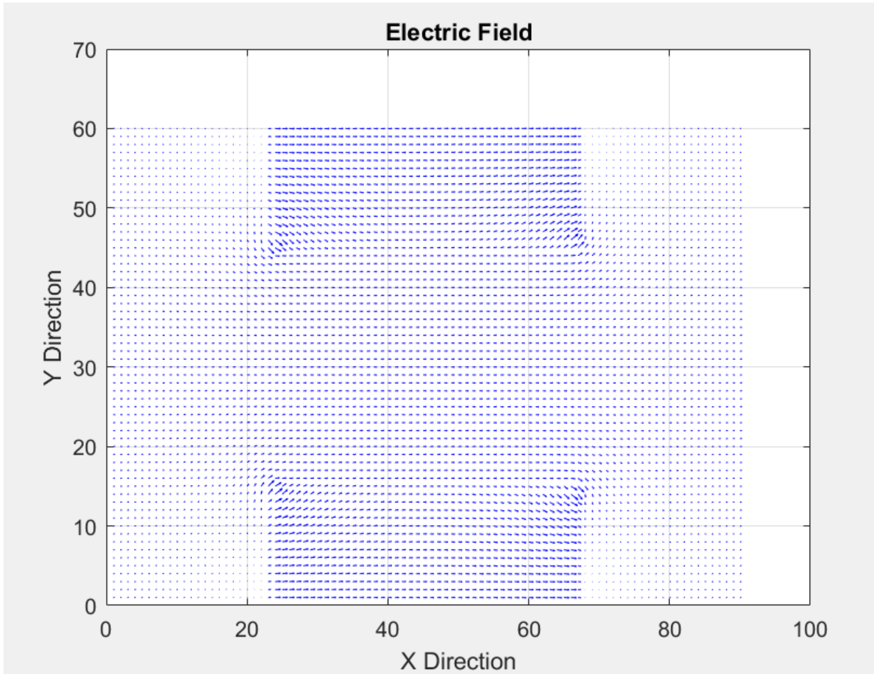

Figure 7: Electric Field

Using the conduction mapping, Ohm's Law can be solved to find the current density in the plate. Figure 8 shows the current density plot in red where the arrow direction and size indicates how large and in what direction the current is flowing.

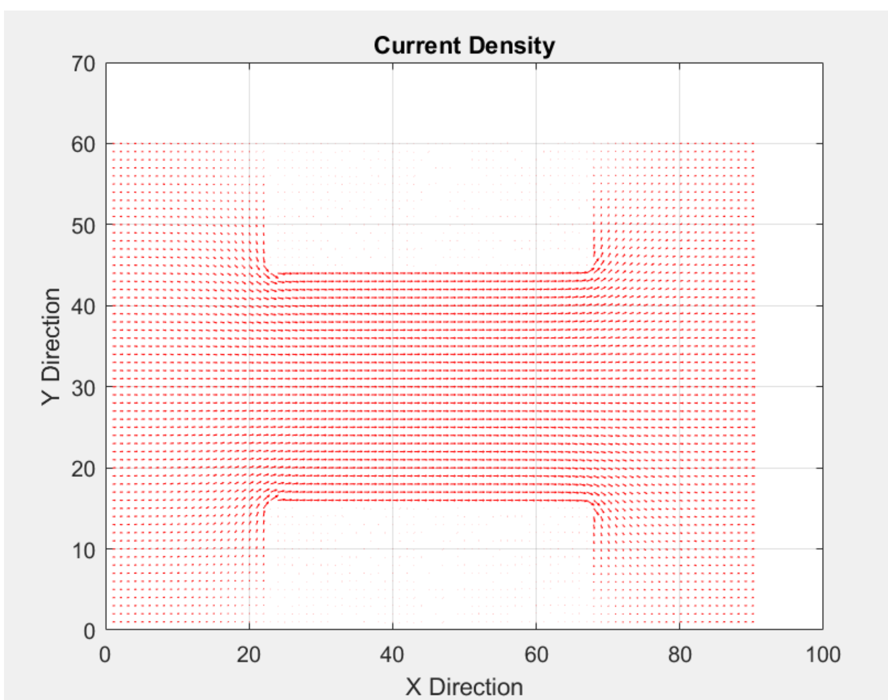

Figure 8: Current Density

From the plot there is almost no current flow in the low conductance regions. All the current goes through the bottleneck and it increases in the thin region. To investigate this more, various bottleneck sizes were simulated to determine the effect this had on the current density. Using the same mesh size of 90 x 60, a bottleneck of 20 percent can be seen in Figure 9.

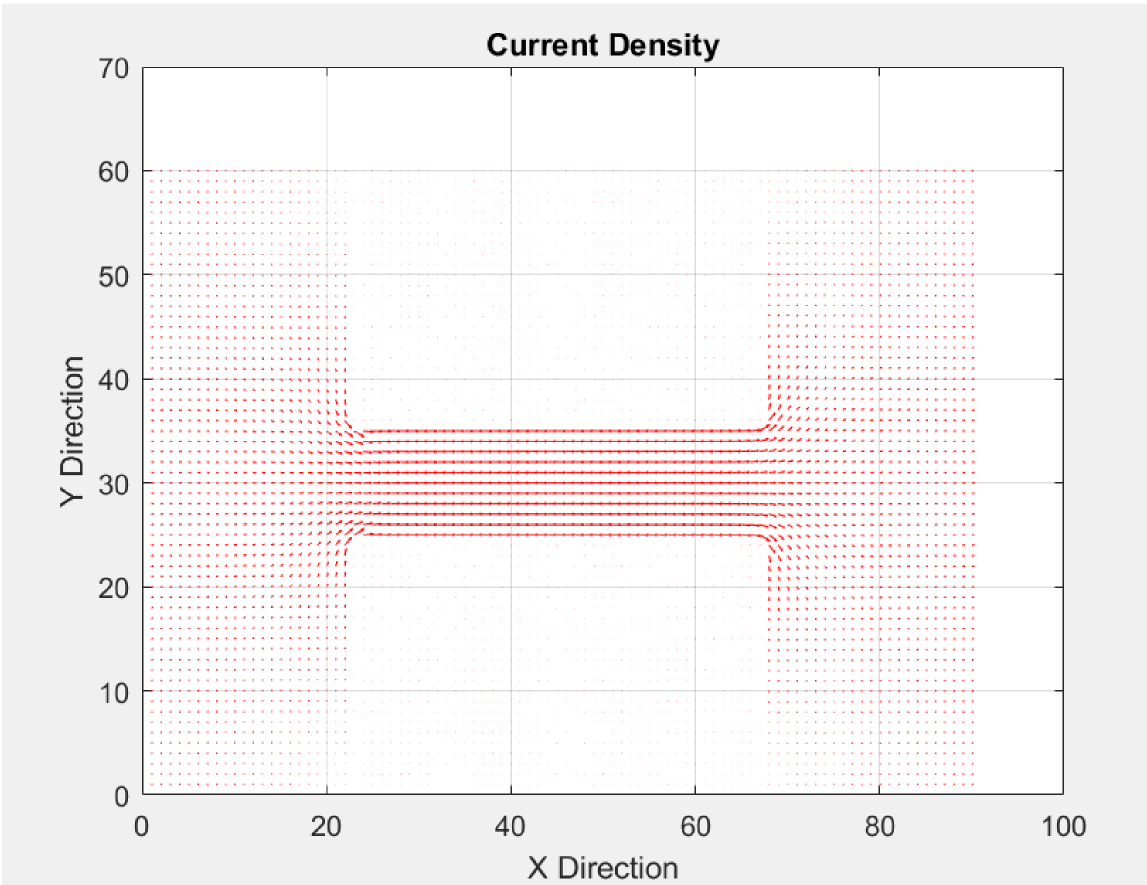

Figure 9: Current Density for Small Bottleneck

Just showing various current densities is not sufficient to show how changing various aspects of the simulation can change the way the current is in the plate. The mesh size, the bottleneck width, and the conductance can all be varied to see the effects they have on the current density in the simulation. For each simulation the default settings are a mesh size of 90 x 60, a bottleneck that has a width of 30 and a length of 45, and a conductance of 1 in the active region, and a 0.01 in the restricted region. Over three simulation, each will be varied one by one to see their effects while the other two will remain in the default position. First the mesh sizes effect on the current density is explored by iterating the simulation over different mesh sizes and calculating the current density as the mesh changes. This is done by looping over a variable N that multiplies the length and width. The aspect ratio of 3:2 is maintained where the simulation starts with 3 x 2 and goes up until 150 x 100. Figure 10 shows the current density change over these mesh sizes.

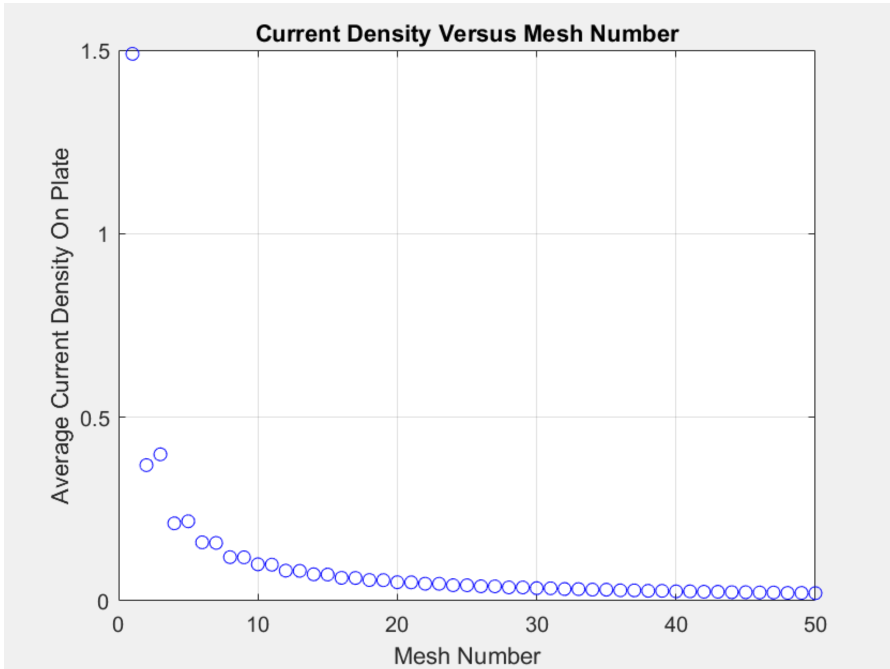

Figure 10: Current Density Versus Mesh Number

From the plot it can be seen that on average, the smaller the mesh size, the lower the current density. While the simulation mesh is small, the current density increases slightly due to how small the mesh is and how MATLAB is handling the calculation. Once the size increases enough, it can be seen that the current density decreases as the mesh number increases. This observation makes sense as the difference between the nodes will decrease which will lower the voltage difference, thus lowering the electric field, thus lowering the current density.

The next parameter to be varied is the size of the bottleneck. The length of the bottleneck is held constant where the middle 50% is the bottleneck region, and the width of the bottleneck is varied from being non-existent to covering the region entirely with low conductance. Figure 11 shows the current density as the bottleneck width shrinks.

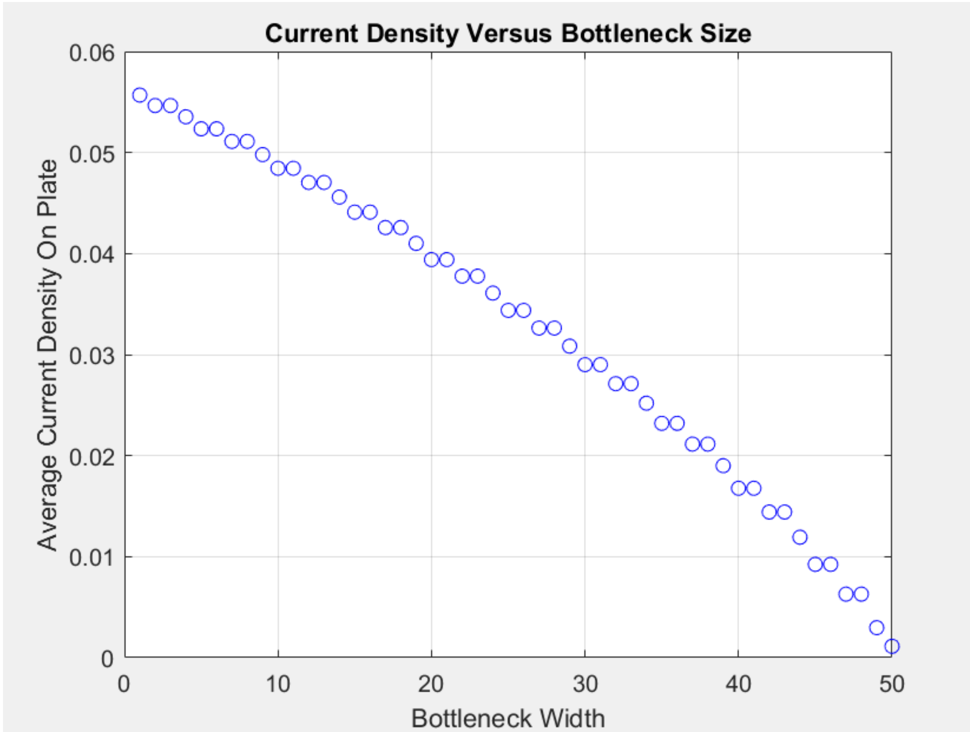

Figure 11: Current Density Versus Bottleneck Width

As the bottleneck region grows to encompass the entire middle of the simulation, the average current density seems to decrease. This makes reasonable sense because as the bottleneck shrinks, there is less of a path for the current to travel so there will be a lower current density.

The last parameter that will be investigated in this simulation is the value of the conductance in the restricted region of the simulation. The conductance is varied from 1 to 0.02 in steps of 0.02 as the simulation is run 50 times. This results in a varying current density that can be seen in Figure 12.

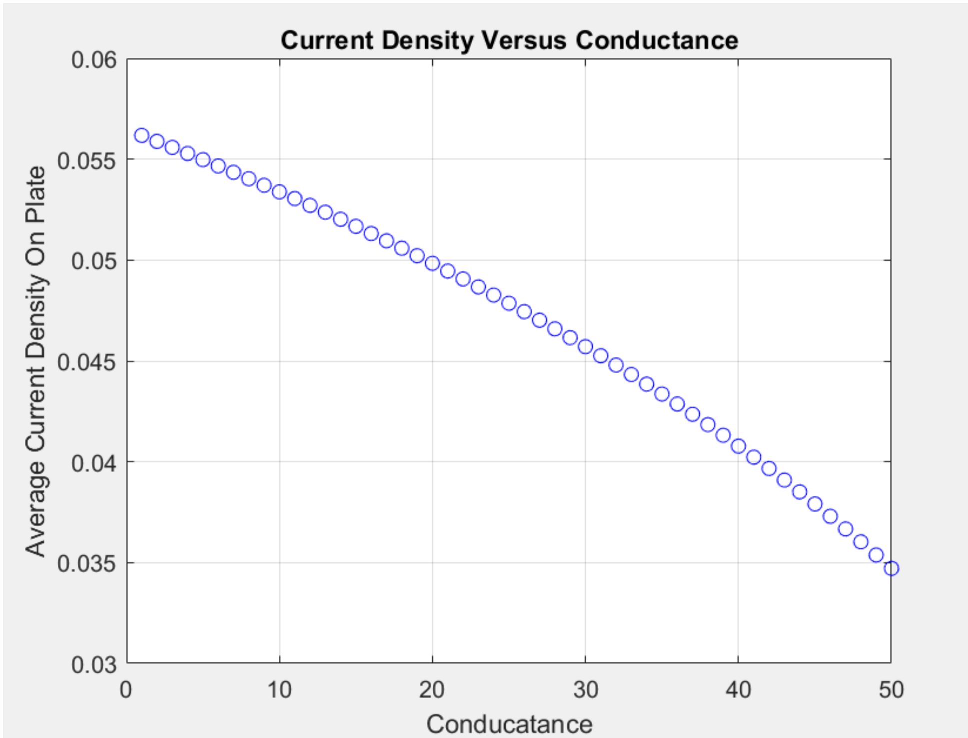

Figure 12: Current Density Versus Conductance

The result of varying the conductance is that as the conductance decreases in the restricted zones, the average current density also decreases. This makes sense, as the simulation starts, its as if there is no restricted zone and the whole simulation has the same conductance. As the simulation continues, the conductance decreases in the restricted zone and the current density also decreases.

Conclusion

Solving the Laplace equation using the forward difference method on MATLAB is a fast and efficient way to get a numerical solution that under certain conditions, reasonably approximate the analytical solution. Various boundary conditions were simulated to ensure that the code worked for many cases. The addition of a bottleneck and different conductance regions were varied to show that the current changes under these conditions. In the future this code can be slightly modified to fit to most boundary case conditions.