Ryan Lindsay 101038101 2/28/2021

1 Part 1

In part 1, Laplace's equation was solved using the finite difference method in a rectangular region with isolated conducting sides. A 2D plot of the voltage within the region was created and a comparison between different numerical and analytical solutions was done.

The first goal of this assignment was to create a voltage plot within the region. Using the constraints given, the following block of code initialized the work space, creating both a G and F matrix that could be iterated through. The matrices were stepped through and then a voltage matrix could be created using the equation V = G/F. This code can be seen below.

```
for i = 1:nx
    for j = 1:ny
        n = ny * (i - 1) + j;
        if i == 1
            G(n,n) = 1;
            F(n) = 1;

        elseif i == nx
            G(n,n) = 1;

        elseif j == 1
            G(n,n) = -3;
            G(n, n+1) = 1;
            G(n, n+ny) = 1;
            G(n, n-ny) = 1;

        elseif j == ny

            G(n,n) = -3;
            G(n, n-1) = 1;
            G(n, n+ny) = 1;
            G(n, n-ny) = 1;

        else
            G(n,n) = -4;
            G(n, n+1) = 1;
            G(n, n-1) = 1;
            G(n, n+ny) = 1;
            G(n, n-ny) = 1;
        end
    end
end
V = G\F;
Vmat = zeros(nx,ny,1);

for i = 1:nx
    for j =1:ny
        n = j+(i-1)*ny;
        Vmat(i,j) = V(n);

    end
end
```

Using this block of code, the following plot was created.

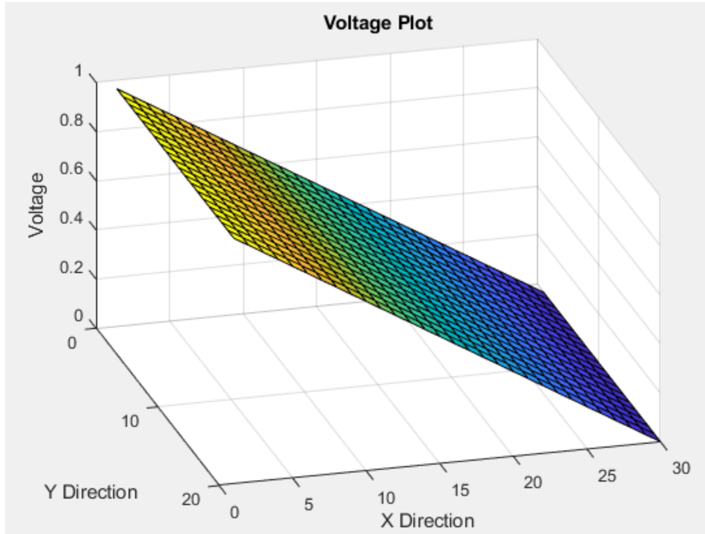

Figure 1: Voltage Plot

After creating the voltage plot, the next step was to solve both the numerical and analytical solutions to the problem. Using the following block of code the numerical solution to Laplace's equation was found, the plot generated by the code follows.

```
for i = 1:nx
    for j = 1:ny

        n = i + (j-1)*nx;
        nxp = (i+1) + (j-1)*nx;
        nxm = (i-1) + (j-1)*nx;
        nym = i + (j-2)*nx;
        nyp = i + (j*nx);

        if i ==1 || i ==nx

            G(n,n) =1;
            F(n) = 1;

        elseif j == 1 || j ==ny

            G(n,n) =1;

        else

            G(n,n) = -4;
            G(n,nxm) = 1;
            G(n,nxp) = 1;
            G(n,nym) = 1;
            G(n, nyp) = 1;
            F(n) = 0;

        end
    end
end

Vmat2 = zeros(nx,ny);
V = G\F;

for i =1:nx
    for j = 1:ny
        n = i + (j-1)*nx;
        Vmat2(i,j) = V(n);

    end
end
```

N

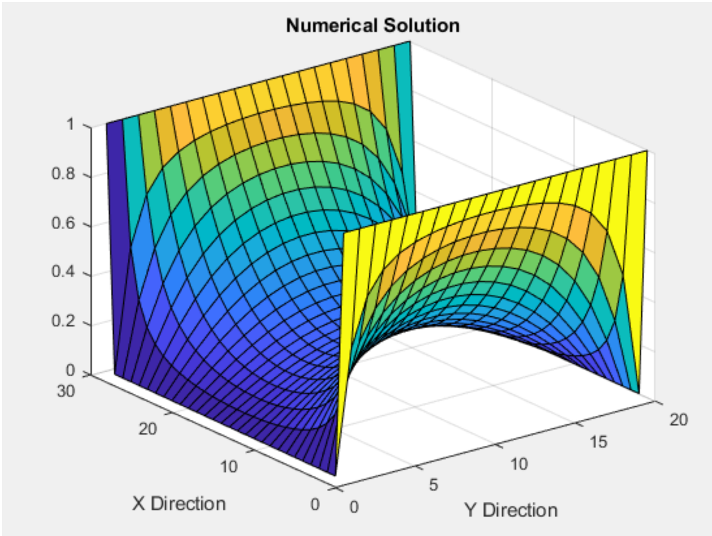

Figure 2: Numerical Solution

Now that the numerical solution has been found, the next step is to use an analytical approach. The next block of code was used to determine this solution. The corresponding plot follows.

```
sum = 0;

for i = 1:nx
    for j = 1:ny
        for n = 1:2:(nx*ny)

            sum = sum + ((1/n) * cosh(n*pi*i/dx) * sin(n*pi*j/dx)) / cosh(n*pi*dy/dx);

        end

    end

    Vmat2 (i,j) = 4*sum/pi;

end


figure(3)
surf(Vmat2)
title('Analytical Solution')
xlabel('Y Direction')
ylabel('X Direction')
```

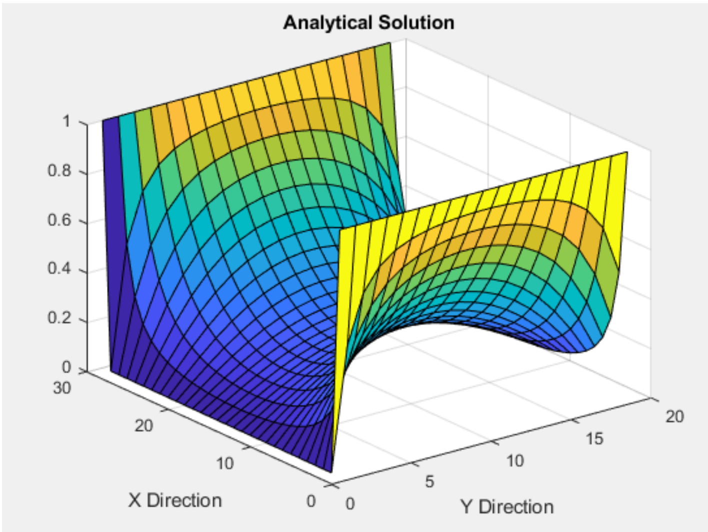

Figure 3: Analytical Solution

Both solutions are quite similar to one another, both generating roughly the same plot. There seems to be a slight error in the analytical solution as it does not perfectly meets the bounds. This could be an programming error on my part or the simulation was simply not run for long enough. Plotting analytically deals with the issue of simulation run time, while it can give a marginally accurate result quite quickly, it takes a very long time for the simulation to be "perfect". However this is contrasted with the simplicity of how it can implemented. For the numerical solution, instead of running through an infinite series we are solving Laplace's equation direction, while it may be more difficult to implement it is more accurate in the long run.

2 Part 2

In part 2, the finite difference method was used to solve for current flow in the rectangular region. In this section several different variables will be measured and then plotted, this include sigma, the E field in both the X and Y direction and the current density. The code that completes this problem can be seen below.

```
for i = 1:nx
    for j = 1:ny

        n = i + (j-1)*nx;
        nxp = (i+1) + (j-1)*nx;
        nxm = (i-1) + (j-1)*nx;
        nym = i + (j-2)*nx;
        nyp = i + (j*nx);

        if i ==1 || i ==nx
            G(n,n) =1;
            F(n) = 1;

        elseif j == 1 || j ==ny
            G(n,n) =1;

        else
            G(n,n) = -4;
            G(n,nxm) = 1;
            G(n,nxp) = 1;
            G(n,nym) = 1;
            G(n, nyp) = 1;
        end
    end
end

Vmat = zeros(nx,ny);
Ex = zeros(nx,ny);
Ey = zeros(nx,ny);
V = G\F;

for i = 1:nx
    for j = 1:ny

        n = i + (j-1)*nx;

        Vmat(i,j) = V(n);


    end
end
```

```
for i = 1:nx
    for j = 1:ny
        if i ==1
            Ex(i,j) = Vmat(i+1,j) - Vmat(i,j);
        elseif i ==nx
            Ex(i,j) = Vmat(i,j) - Vmat(i-1,j);
        else
            Ex(i,j) = (Vmat(i+1,j) - Vmat(i-1,j))*0.5;
        end
        if j ==1
            Ey(i,j) = Vmat(i, j+1) - Vmat(i,j);
        elseif j == ny
            Ey(i,j) = Vmat(i,j) - Vmat(i,j-1);
        else
            Ey(i,j) = (Vmat(i, j+1) - Vmat(i, j-1))*0.5;
        end
    end
end
```

Using this code, four different plot were created, they can all be seen the following figures.

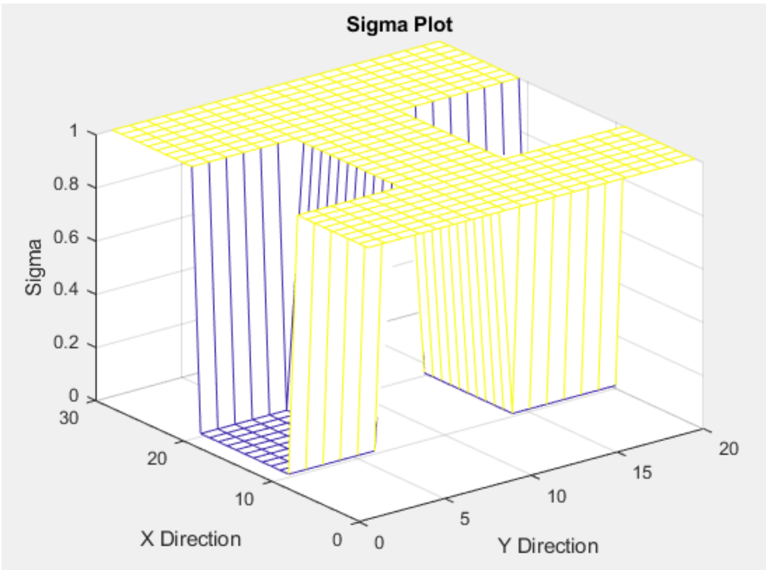

Figure 4: Sigma Plot

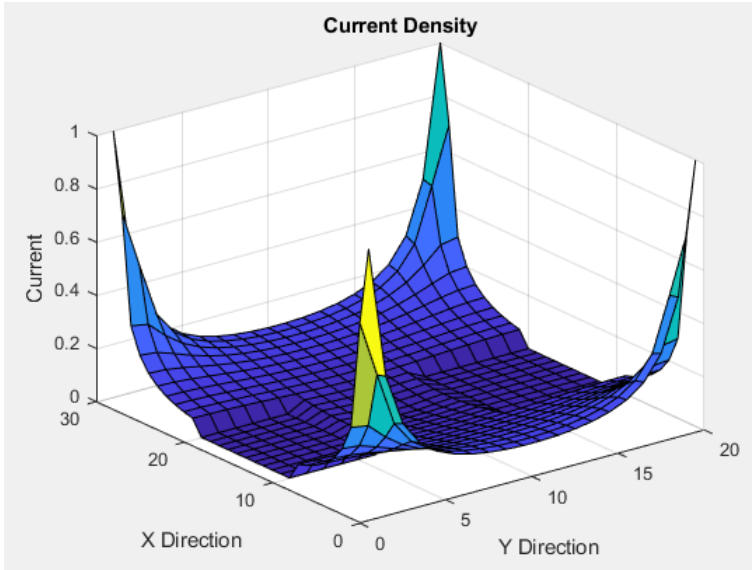

Figure 5: Current Density

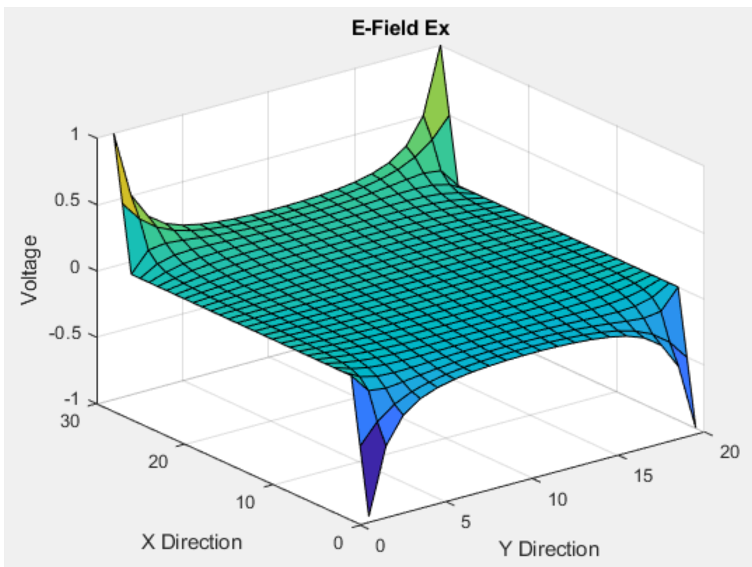

Figure 6: Ex Field

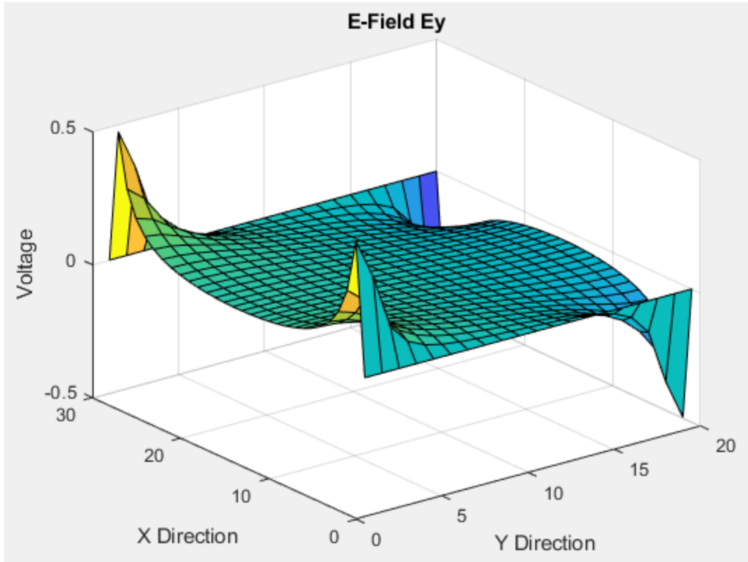

Figure 7: Ey Field

The next step of this assignment is to determine how the current reacts to a changing mesh size of the region. The following block of code that completes this task is quite large and repeats a large amount of code so it is impractical to include it. Essentially, a large loop that adjusted the size of the mesh ran over top of the previously written code, the current was recorded and then compared against the various sizing of the mesh. The plot of the current vs the mesh size can be seen in the figure below.

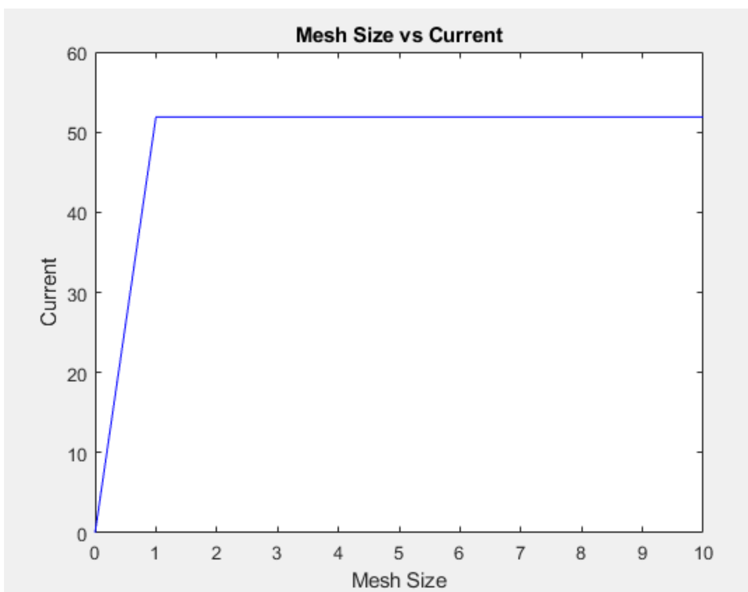

Figure 8: Ey Field

From this plot, we can gather that the mesh was already sufficiently large enough, and that increasing the mesh further did not have an affect on the system. This could be due to the fact that the system is already simplistic, so we should not expect any significant changes. For a larger and more complex system, this should have more of an effect.

The next section of this report deals with adjusting the bottle neck area of the region. Four new regions were created and then compared with the original region that was given. The following block of code shows how each bottleneck region was set up.

```
% Create multiple instances of sigma
sigma1 = ones(nx,ny);
sigma2 = ones(nx,ny);
sigma3 = ones(nx,ny);
sigma4 = ones(nx,ny);
sigma5 = ones(nx,ny);
```

```
% Adjust the boundaries of the various sigmas
for i=1:nx

    % Original Sigma
    for j =1:ny
        if j <= ny/3 || j>= 2*ny/3
            if i >= nx/3 && i<= 2*nx/3
                sigma1(i,j) = 10^-2;
            end
        end

     % Sigma 2 Lengthen X barrier
     if j <= ny/3 || j>= 2*ny/3
         if i >= nx/8 && i<= 7*nx/8
             sigma2(i,j) = 10^-2;
         end
     end

     % Sigma 3 Shorten X barrier

     if j<= ny/3 || j>= 2*ny/3
         if i > 7*nx/16 && i <= 9*nx/16
             sigma3(i,j) = 10^-2;
         end
     end

     % Sigma 4 Lengthen Y barrier
     if j<=ny/8 || j>= 7*ny/8
         if i>= nx/3  && i <= 2*nx/3
             sigma4(i,j) = 10^-2;
         end
     end

     % Sigma 5 Shorten Y Barrier
     if j<=7*ny/16 || j>= 9*ny/16
         if i>= nx/3  && i <= 2*nx/3
             sigma5(i,j) = 10^-2;
         end
     end
    end
end
```

As mentioned previously, this code generates 4 new bottle necks, as well as the original bottleneck used previously. The current for each sigma map was also recorded and used to create a plot of current vs each of the corresponding sigmas. The following sigma plots for each bottleneck can be seen below.

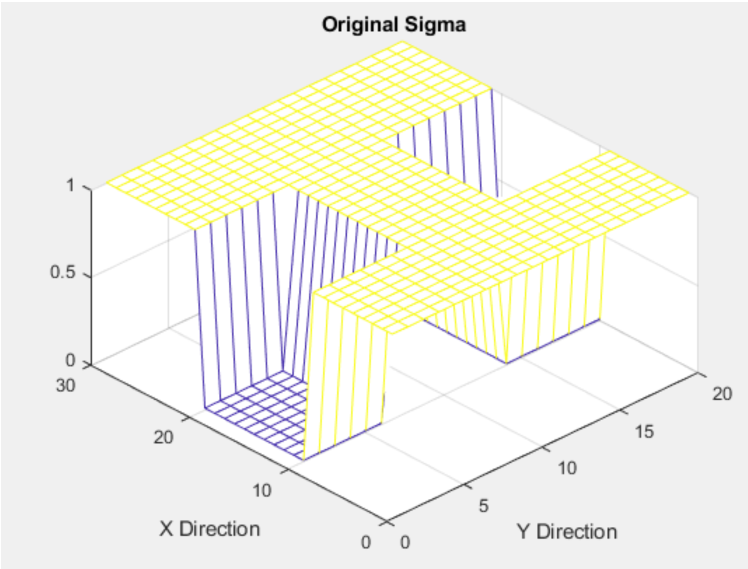

Figure 9: Original Sigma

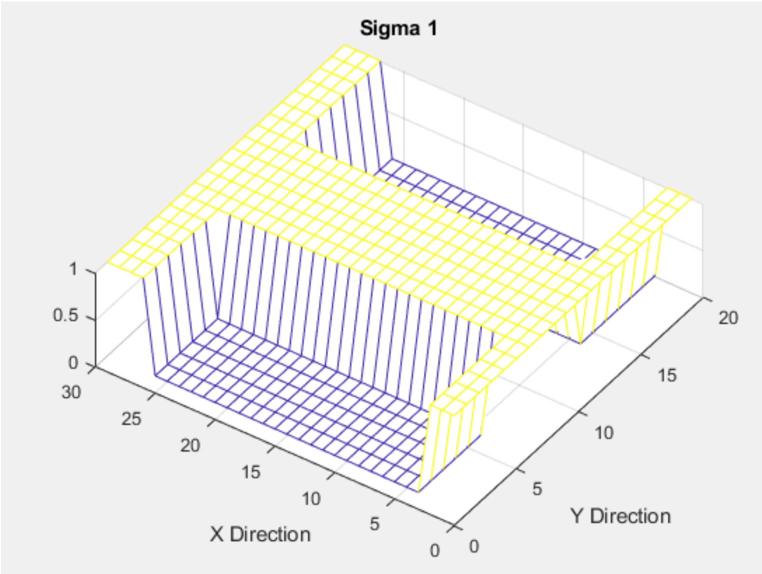

Figure 10: Sigma 1

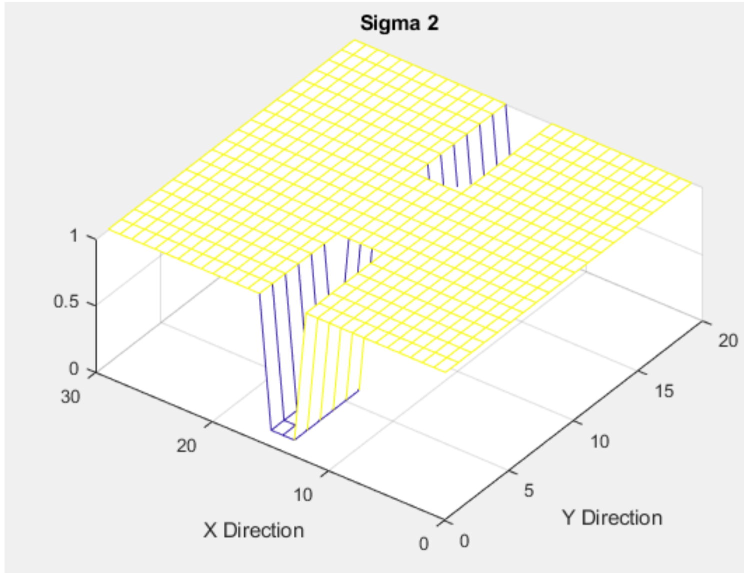

Figure 11: Sigma 2

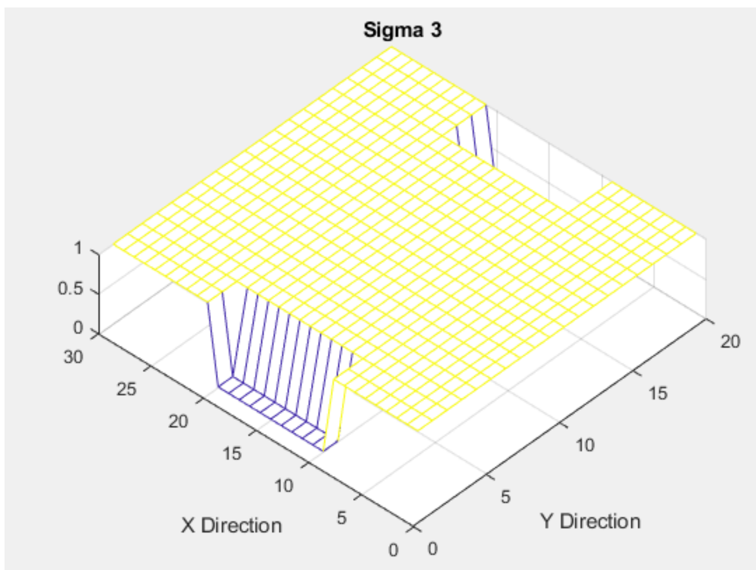

Figure 12: Sigma 3

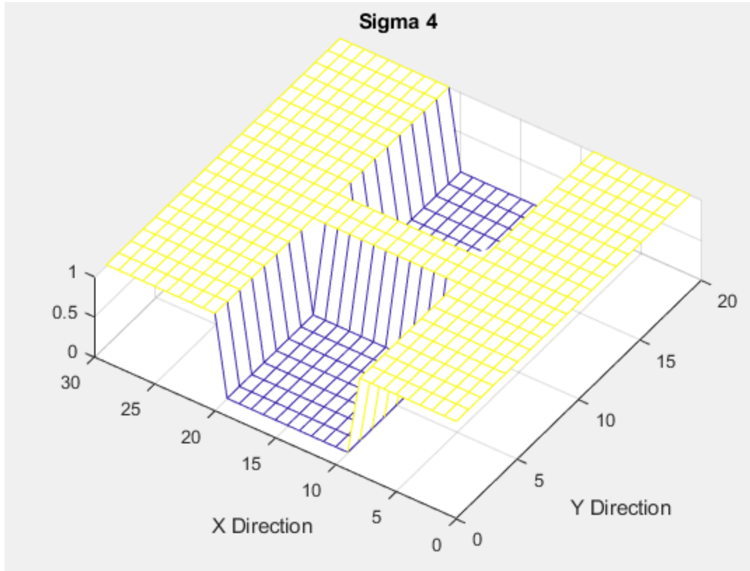

Figure 13: Sigma 4

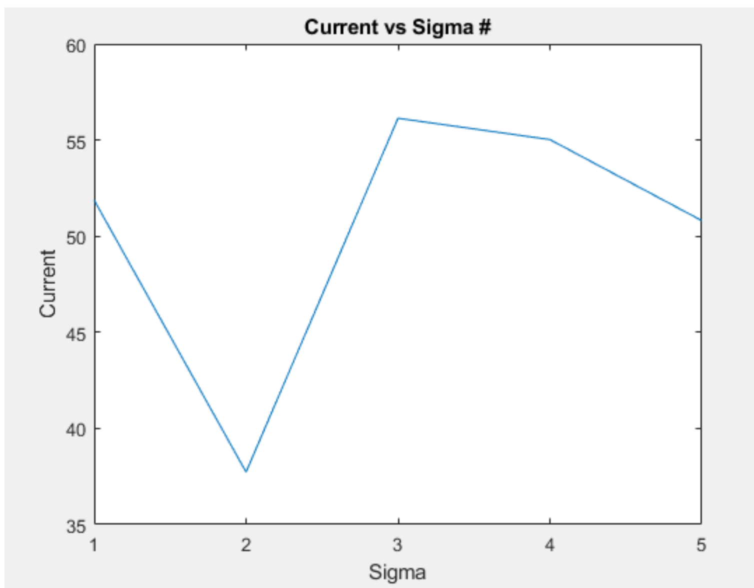

Figure 14: Sigma 4

The final portion of the assignment is to determine how the current is affected when sigma is increased. Similar to the mesh size problem, a large loop was used which loop through various sizes of sigma and then compared the current against it. The following figure shows that when sigma is increased, the current increases linearly along with it.

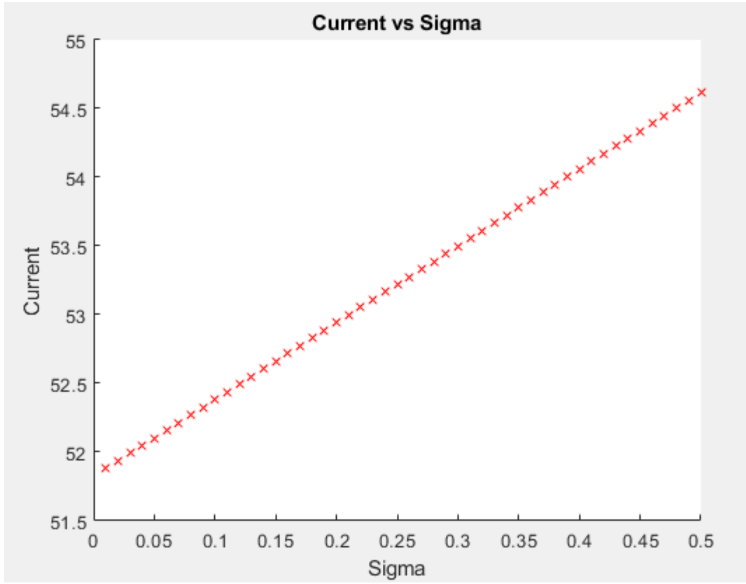

Figure 15: Sigma 4

In conclusion, the simulations were successful in producing accurate representations of the Laplace equation using two different methods and electrostatic potential. Several bottlenecks were also investigated and compared against one other in terms of current and sigma.