Part 1 - Finite Difference Method

a) The simple 2D case plots are shown below. Figure 1 is solving by iteration and figure 2 is solved with a G matrix.

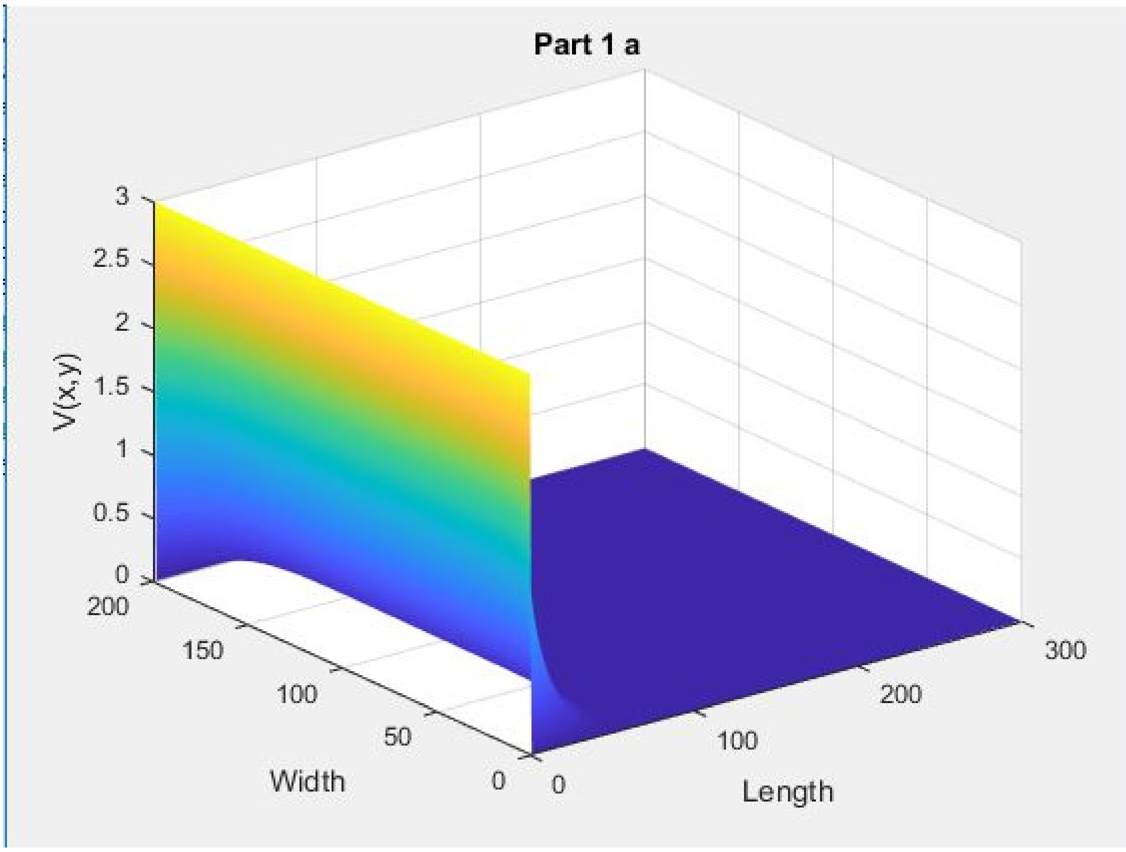

Figure 1: Simple 2-D case solved by iteration

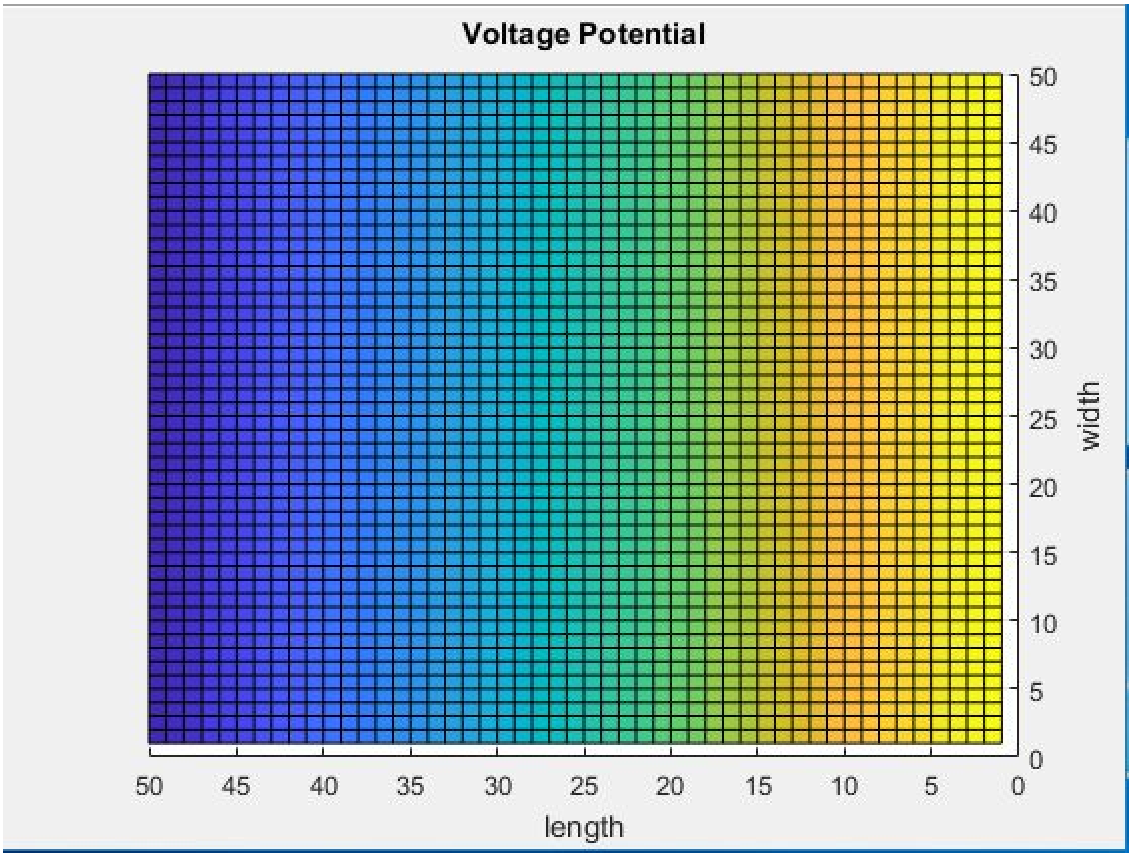

Figure 2: Simple 2-D case solved with a G matrix

b) Part B was to solve the case with boundary conditions on either side of the box. The solution solved by iteration is shown in figure 3 below.

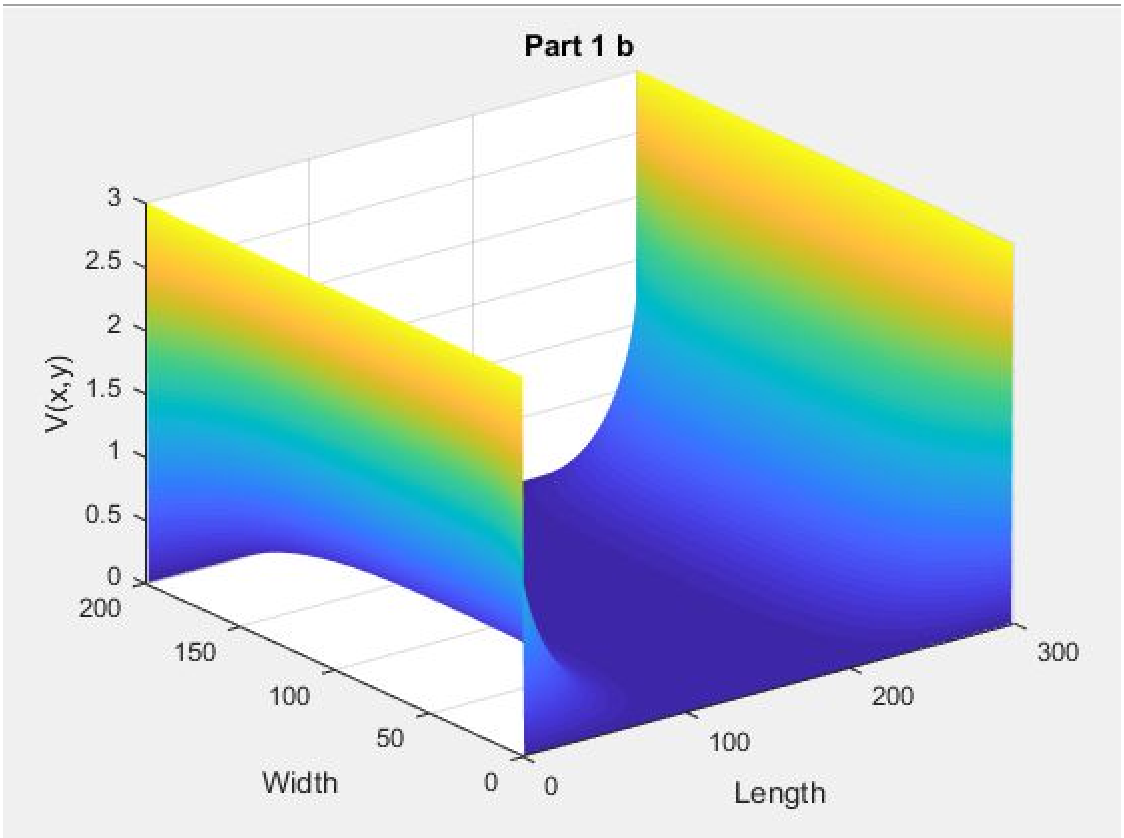

Figure 3: 2 B.C solved by iteration.

Part 2 - Finite Difference Method with “Bottle-neck”

a) Below are the plots for sigma(x,y) V(x,y), E(x,y), and J(x,y).

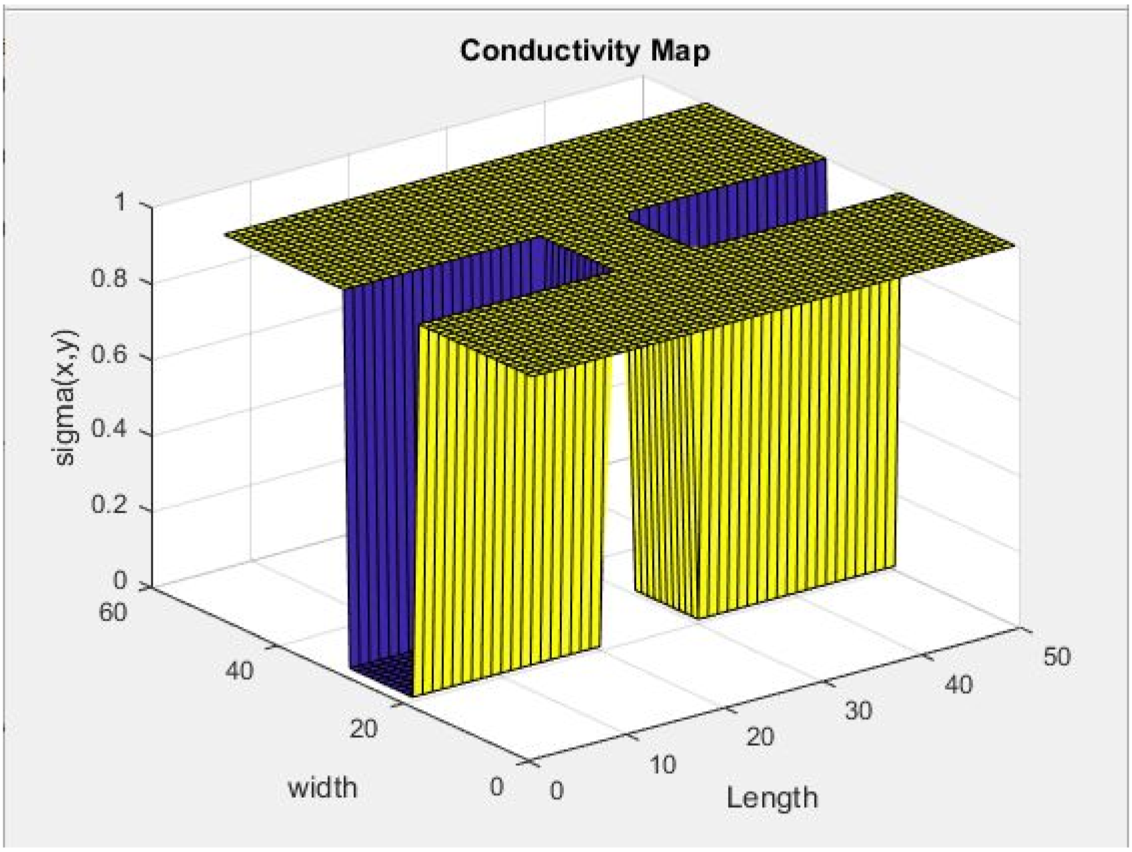

Figure 4: Conductivity Map

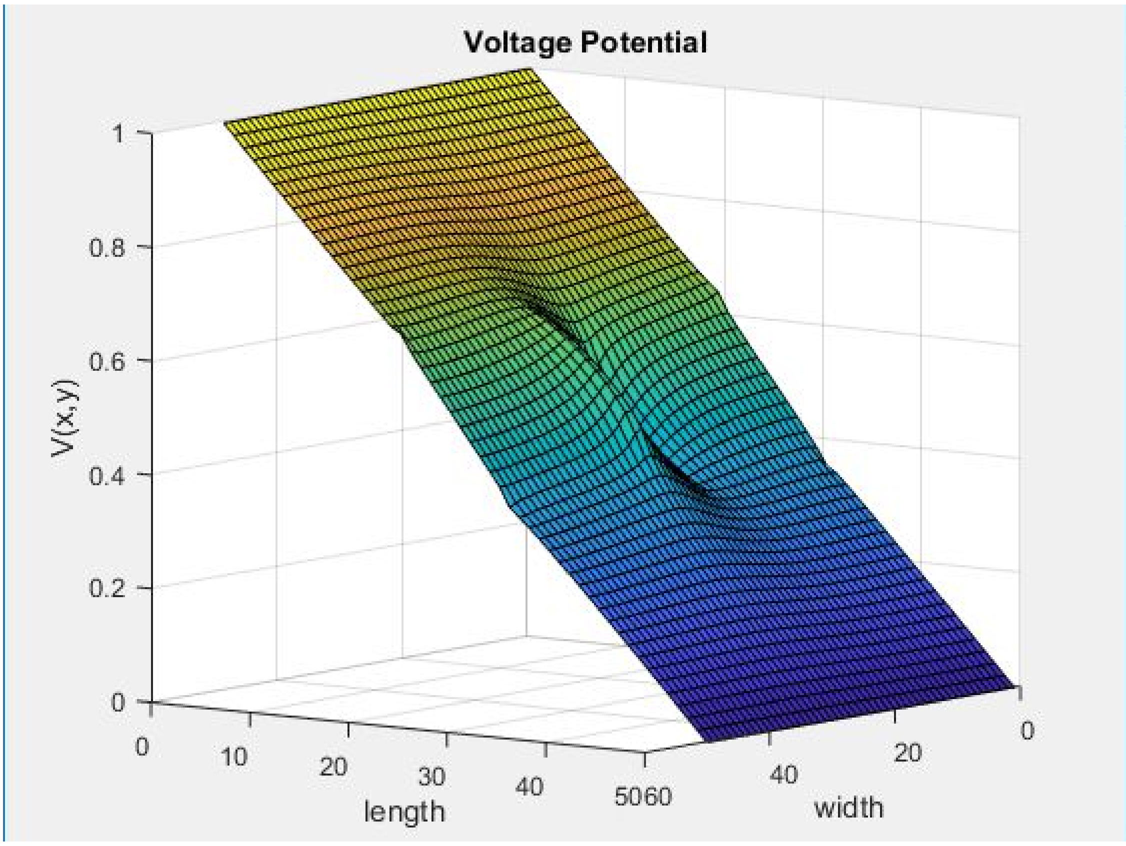

Figure 5: Voltage Potential

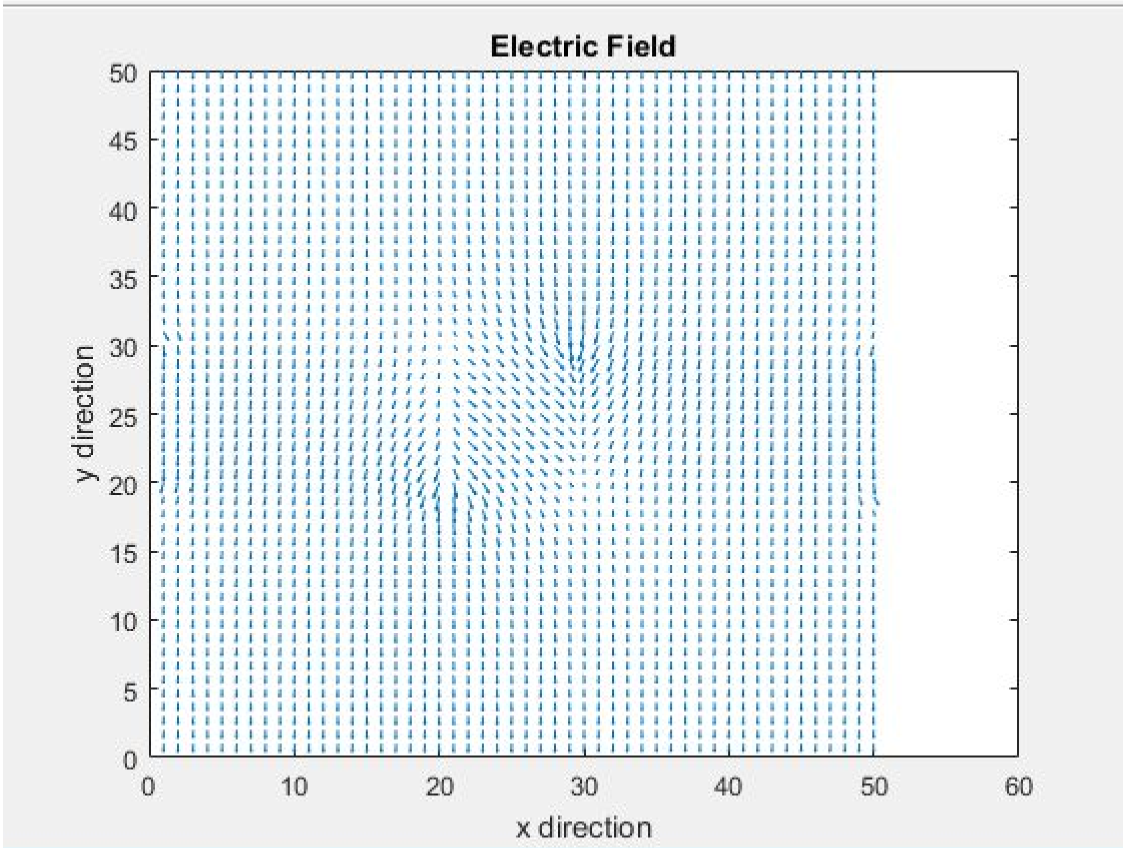

Figure 5: Electric Field

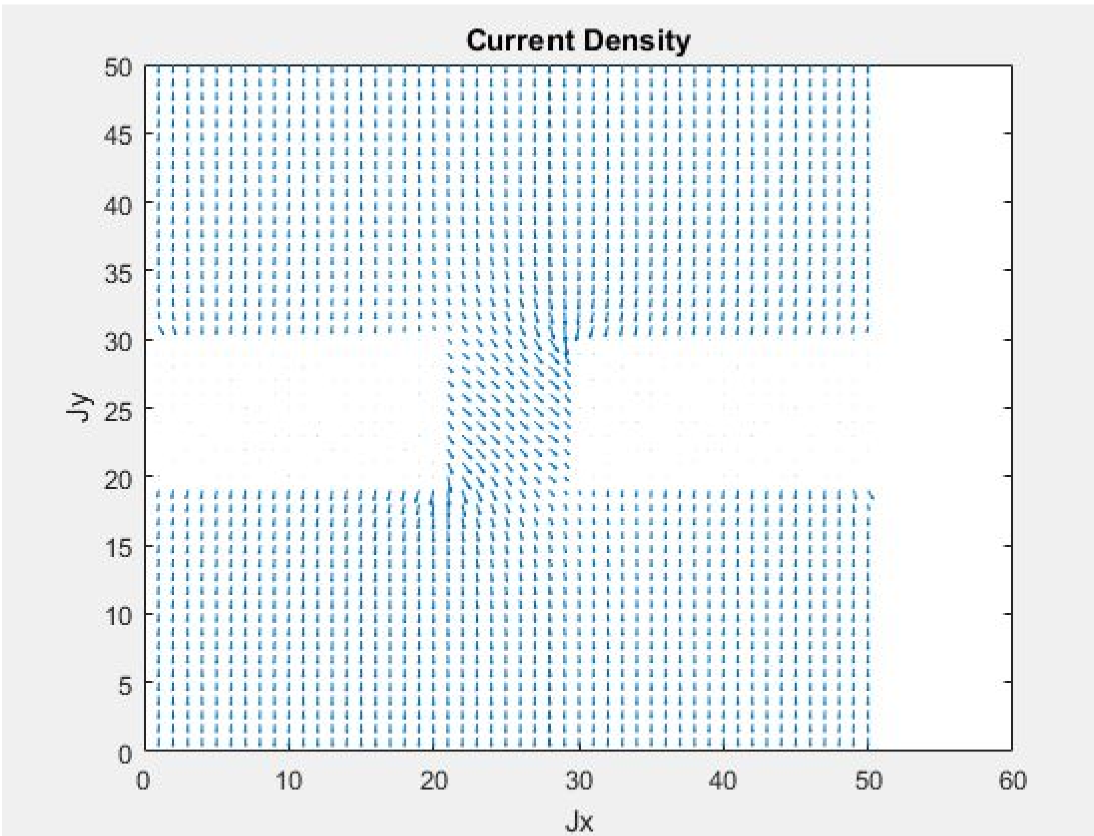

Figure 6: Current Density

Conclusions

For part 2 of the assignment, the affect of bottle necking was investigated. The conductivity was mapped to a matrix and plotted in 3 dimensions. The “Cmap” was then mapped onto a G-matrix and the equivalent resistance was found using the following code.

```
for i=1:1:ny
   for j=1:1:nx
      n = j +(i-1)*nx;
      if i==1
         G(n,:) = 0;
         G(n,n) = 1;
         F(n,1) = 1;
      elseif i==ny
         G(n,:) = 0;
         G(n,n) = 1;
         F(n,1) = 0;
      elseif j== 1
         G(n, n+1) = 1/((1/(2*Cmap(i+1,j))) + (1/(2*Cmap(i,j))));
         G(n , n-1)= 1/((1/(2*Cmap(i-1,j))) + (1/(2*Cmap(i,j))));
         G(n,n+nx) = 1/((1/(2*Cmap(i,j+1))) + (1/(2*Cmap(i,j))));

         G(n,n) = -(G(n,n+1) + G(n,n-1) + G(n,n+nx));

      elseif j == nx
         G(n, n+1) = 1/((1/(2*Cmap(i+1,j))) + (1/(2*Cmap(i,j))));
         G(n , n-1)= 1/((1/(2*Cmap(i-1,j))) + (1/(2*Cmap(i,j))));
          G(n,n-nx) = 1/((1/(2*Cmap(i,j-1))) + (1/(2*Cmap(i,j))));
         G(n,n) = -(G(n,n+1) + G(n,n-1) + G(n,n-nx));

      else

         if Cmap(i,j-1) ~= Cmap(i,j)

         end

         G(n, n+1) = 1/((1/(2*Cmap(i+1,j))) + (1/(2*Cmap(i,j))));
         G(n , n-1)= 1/((1/(2*Cmap(i-1,j))) + (1/(2*Cmap(i,j))));
         G(n,n-nx) = 1/((1/(2*Cmap(i,j-1))) + (1/(2*Cmap(i,j))));
         G(n,n+nx) = 1/((1/(2*Cmap(i,j+1))) + (1/(2*Cmap(i,j))));
         G(n,n) = -(G(n,n+1) + G(n,n-1) + G(n,n-nx) + G(n,n+nx));
      end
   end
end
```

For some reason, the G matrix was not mapped correctly, as seen by the voltage potential plot, there should not be a peak and should be a rectangular dip only. The way I chose to fill in the G matrix was to create 2 ressistors for each node and add them together. What I think happened was that the R on the left node did not match the R on the right node, this is what caused a fault in my program. I tried to fix the problem but in the end I could not solve the issue.

The voltage was then reshaped onto a new vector to surf using the following code.

```
V = G\F;
X = zeros(ny,nx,1);
for i=1:1:ny
   for j=1:1:nx
      n = j +(i-1)*nx;
      X(i,j) = V(n);
   end
end

  figure(2);
  surf(X(:,:,1));
  title 'Voltage Potential';
  xlabel 'width';
  ylabel 'length';
  zlabel 'V(x,y)';
```

This plotted my figure and we can clearly see that it plotted as expected. The only problem in the code was discussed earlier.