SYSC 4405: Modelling of Integrated Devices

Assignment 2:

Finite Difference Method

Neale Skinner 100965404

Due: 2020-02-23

Introduction:

In this assignment we are exploring Laplace's equation by finite difference method. We will be looking at electrostatic potential as well as current flow in bounded areas. For part one we will be comparing our analytical solution to the experimental solution to see how they converge and if they are similar. In part two we modify our bounds inside the mesh box as well as see what happens when we change constant values over time.

Electrostatic Potential:

For part one we are exploring electrostatic potential in a bounded, rectangular region. Given our constants and bound widths, we design code to solve this, shown below is the code for part 1.

```
x= 3;            % x dimension
y=2;             % y dimension
%%%%% ratio for L/W is 3/2

dx = .1;
dy = .1;
nx = x/dx;
ny = y/dy;
%%create matrices for the F and G values.
G = sparse(nx*ny,nx*ny);        %large matrix of mostly 0's
F = zeros(nx*ny,1);
%step through the each point and change the values of the matrix that
are
%equivalent to n or the boundaries
for i = 1:nx
for j = 1:ny
n = j + (i - 1) * ny;
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
%%%%rearrange the equation GV=F to find V,
```

```
Volt = G\F;
%%%create a matrix/ map that holds the voltages in each cell
Voltmap = zeros(nx,ny,1);
%for loop that changes cells of voltmap to the given voltage
for i = 1:nx
for j = 1:ny
n = j+(i-1)*ny;
Voltmap(i,j) = Volt(n);
end
end
%%use surf to 3d map the linear voltage
figure(1)
surf(Voltmap);
title('Voltage Plot from FD');
xlabel('x');
ylabel('y');
zlabel('Voltage');
view(-100,10)
```

the given code above gives us the result of figure 1 below.

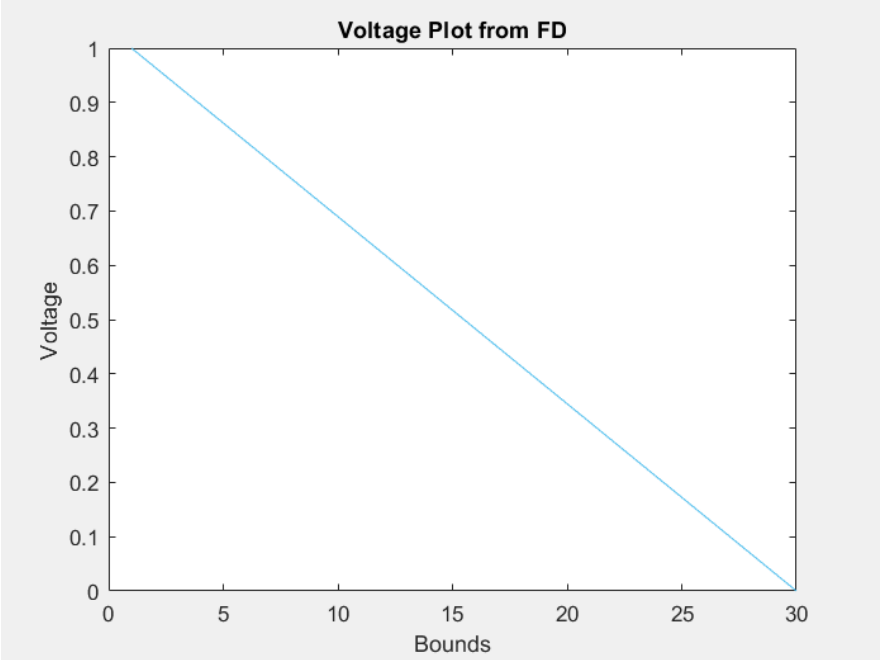

Figure 1: Voltage Plot from Finite Difference Method

From the above plot we can see that the voltage rises when Y is at 0 and is evenly distributed across X. the voltage then drops as Y is lowered.

In part 1. B we are comparing our analytical solution to the numerical solution. The code for part B is shown below.

```
%%%%%%%%%%%%%%%%%%%%%%part b%%%%%
G = sparse(nx*ny,nx*ny);
F = zeros(nx*ny,1);
for i = 1:nx
    for j = 1:ny
```

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

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

Voltmap2 = zeros(nx, ny);
V = G\F;

for i = 1:nx
    for j = 1:ny
        n = i + (j - 1)*nx;
        Voltmap2(i, j) = V(n);
    end
end

figure(2)
mesh(Voltmap2)
title('Electrostatic Potential: Numerical')
xlabel('x')
ylabel('y')

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
    Voltmap2(i, j) = (4 * sum ) / pi ;
end

figure(3)
mesh(Voltmap2)
```

```
title('Electrostatic Potential: Analytical')
xlabel('x')
ylabel('y')
```

The code outputs two plots, the first being the numerical solution and the second being the analytical solution.

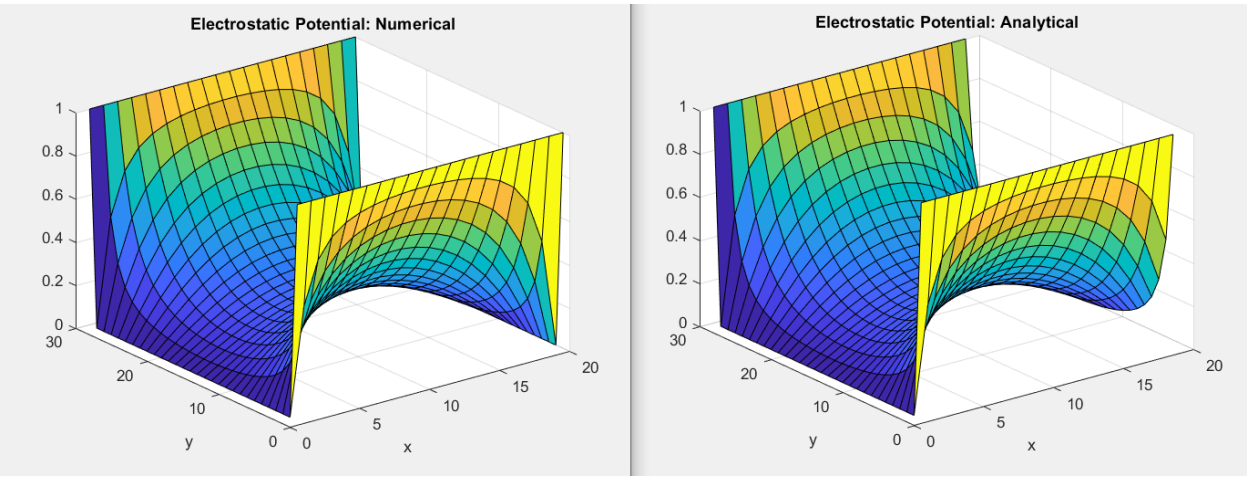

Figure 2: Electrostatic Potential from Numerical and Analytical

The above plots are quite similar, the only difference is what happens when the X variable is closing towards the outer bound and the Y value is reaching 0. The analytical value rounds out instead of hitting zero as the numerical solution does.

Current Flow:

In part 2 we are exploring the current flow in the rectangular bounds. In this section we will be looking at phi, voltage, density on each axis as well as the total density. We will also be experimenting with changing the "bottle neck" of the bounds as well as the phi.

Part 2. A is the calculating of each variable with the normal bounds and bottle neck given. The code is shown below.

```
x= 3;            % x dimension
y=2;             % y dimension
%%%%% ratio for L/W is 3/2

dx = .1;
dy = .1;
nx = x/dx;
ny = y/dy;
%%create matrices for the F and G values.
G = sparse(nx*ny,nx*ny);        %large matrix with mostly 0's
F = zeros(nx*ny,1);
% iterate through each point and give it the proper value.
for i = 1:nx
    for j = 1:ny

        n = i + (j - 1) * nx;
        nym = i + (j - 2) * nx;
```

```
        nyp = i + j * nx;
        nxm = i - 1 + (j - 1) * nx;
        nxp = i + 1 + (j - 1) * nx;

        if i == 1
            G(n, n) = 1;
            F(n) = 1;
        elseif i == nx
            G(n, n) = 1;
            F(n) = 1;
        elseif j == 1
            G(n, n) = 1;
        elseif 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
%create the voltmap as well the matrix for V, density on X and Y.
Voltmap2 = zeros(nx, ny);
Ex=zeros (nx, ny);
Ey=zeros (nx, ny);
V = G\F;
%numerical way to answer voltage map
for i = 1:nx
    for j = 1:ny
        n = i + (j - 1)*nx;
        Voltmap2(i, j) = V(n);
    end
end
%iterate through the density matrices and find the corresponding
density.
 for i = 1:nx
    for j = 1:ny
        if i == 1
            Ex(i, j) = Voltmap2(i+1, j) - Voltmap2(i, j);
        elseif i == nx
            Ex(i, j) = Voltmap2(i, j) - Voltmap2(i-1, j);
        else
            Ex(i, j) = (Voltmap2(i+1, j) - Voltmap2(i-1, j))*0.5;
        end

        if j == 1
            Ey(i, j) = Voltmap2(i, j+1) - Voltmap2(i, j);
        elseif j == ny
            Ey(i, j) = Voltmap2(i, j) - Voltmap2(i, j-1);
        else
            Ey(i, j) = (Voltmap2(i, j+1) - Voltmap2(i, j-1))*0.5;
        end
    end
 end
%plot the electric field for X and Y.
```

```
figure(1)
surf(Ex)
title('Electric Field Ex')
xlabel('x')
ylabel('y')

figure(2)
surf(Ey)
title('Electric Field Ey')
xlabel('x')
ylabel('y')
%create the sigma matrix of ones. could have chosen 0, or 1.
sigma = ones(nx,ny);
%iterate through the points and change where the bottle neck is.
for i = 1:nx
    for j = 1:ny
        if j <= (ny/3) || j >= (ny*2/3)
            if i >= (nx/3) && i <= (nx*2/3)
                sigma(i,j) = 10^-12;
            end

        end
    end
end
%find the total electric field
TotalE= sqrt(Ex.^2+ Ey.^2);
%J is the total electric field where sigma is.
J=sigma .* TotalE;
% plot sigma map
figure(3);
mesh(sigma);
xlabel('X');
ylabel('Y');
title('Sigma Map');
%plot current density map
figure(4)
mesh(J)
title('Current Density Map')
xlabel('x')
ylabel('y')
```

the above code will output 4 figures. It will give us our sigma map, our current density and our electric fields.

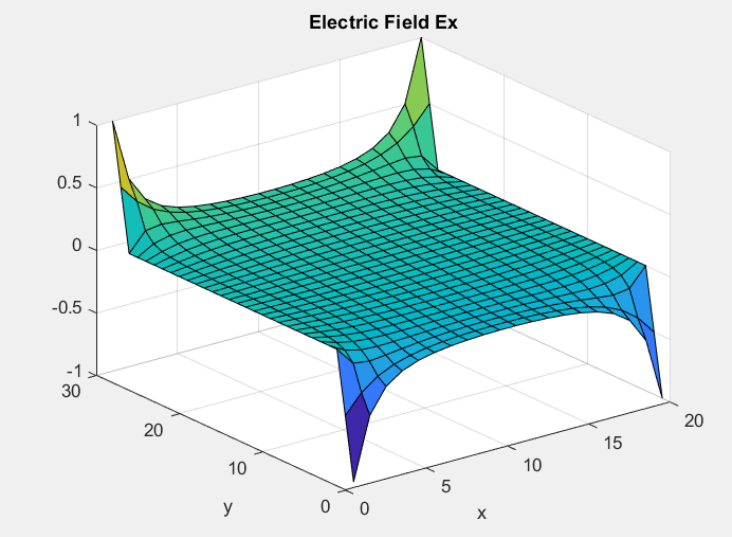

Figure 3: Electric Field Ex

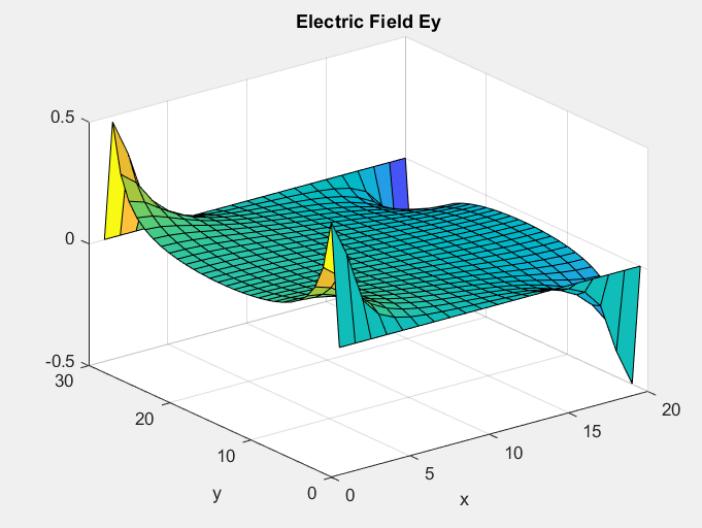

Figure 4: Electric Field Ey

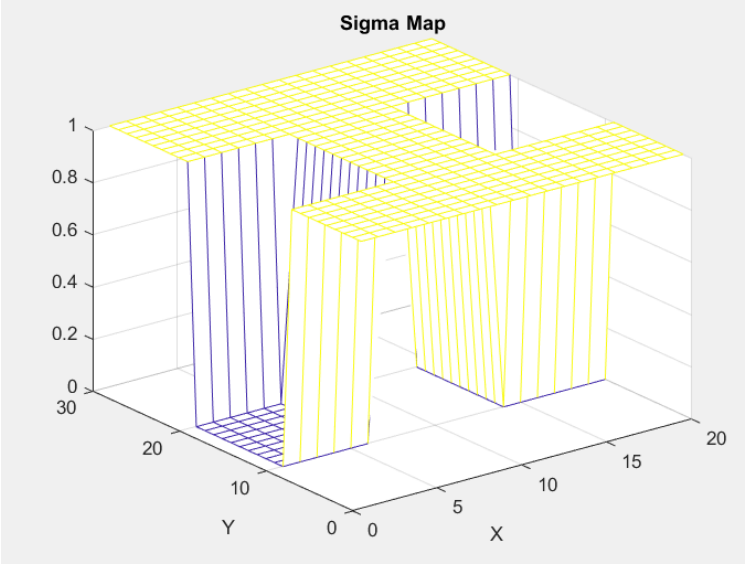

Figure 5: Sigma Map

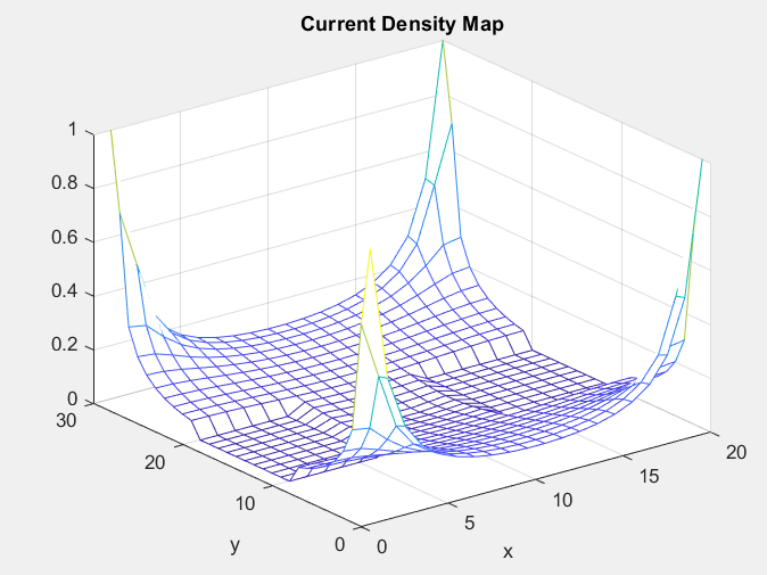

Figure 6: Current Density Map

In part B of this assignment we are exploring what will occur to the current to as the mesh gets more dense. The code is shown below.

```
current =0;
OldCurrent=0;
TotalCurrent=0;
TotalE=0 ;
G = sparse(nx*ny,nx*ny);
F = zeros(nx*ny,1);
J=0;
TotalE=0;
sigma=0;
for meshsize = 1:5
```

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

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

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

Voltmap2 = 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;
        Voltmap2(i, j) = V(n);
    end
end

sigma = ones(nx,ny);
    for i = 1:nx
    for j = 1:ny
        if j <= (ny/3) || j >= (ny*2/3)
            if i >= (nx/3) && i <= (nx*2/3)
                sigma(i,j) = 10^-12;
            end

        end
    end
end
 for i = 1:nx
    for j = 1:ny
        if i == 1
            Ex(i, j) = Voltmap2(i+1, j) - Voltmap2(i, j);
        elseif i == nx
```

```
                Ex(i, j) = Voltmap2(i, j) - Voltmap2(i-1, j);
            else
                Ex(i, j) = (Voltmap2(i+1, j) - Voltmap2(i-1, j))*0.5;
            end

            if j == 1
                Ey(i, j) = Voltmap2(i, j+1) - Voltmap2(i, j);
            elseif j == ny
                Ey(i, j) = Voltmap2(i, j) - Voltmap2(i, j-1);
            else
                Ey(i, j) = (Voltmap2(i, j+1) - Voltmap2(i, j-1))*0.5;
            end
        end
    end
      TotalE= sqrt(Ex.^2 + Ey.^2);
      J=sigma .* TotalE;
      OldCurrent=current;
      current =sum(J,'All');
      figure(5)
      plot([meshsize-1 meshsize], [OldCurrent current])
      hold on
      title ('Current vs Mesh Size')
end
   xlabel('Mesh Size')
   ylabel('Current')
```

The output of this code gives us figure 7. When ran in Matlab is updated over each iteration.

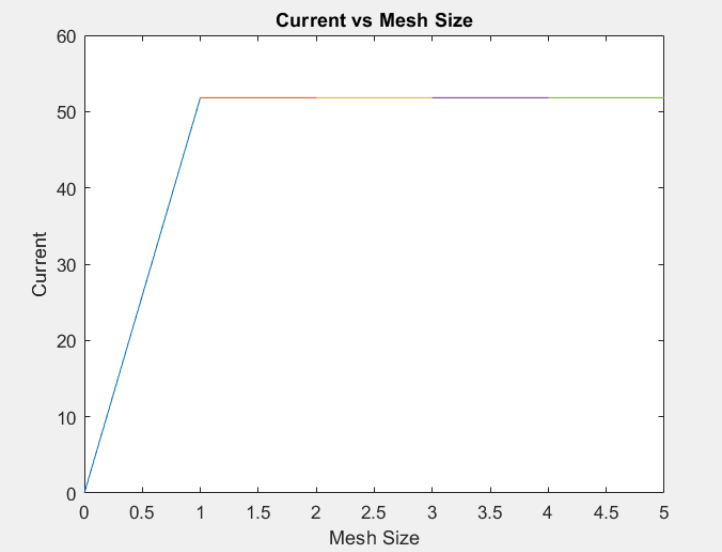

Figure 7: Current over Increased Density

From the plot we can see that the current spikes quite quickly with increased mesh size, but quickly evens out as the mesh size continuously increases. Part C is the experimental part with changing the "Bottle- neck" area to see what will happen with sigma. The code is shown below.

```
%%%%%%%%%%%%%%%%%%%%Part C%%%%%%%%%%%%%%
% change the size of our boxes inside dimensions and see effect on
current
% density.
```

```
sigma1 = ones(nx,ny);
sigma2 = ones(nx,ny);
sigma3 = ones(nx,ny);
sigma4 = ones(nx,ny);

for i = 1:nx %Changing the lengths and widths
        for j = 1:ny
            if j <= (ny/3) || j >= (ny*2/5)
                if i >= (nx/5) && i <= (nx*2/3)
                    sigma1(i,j) = 10^-12;
                end
            end
        if j <= (ny/4) || j >= (ny*3/4)
            if i >= (nx/2) && i <= (nx*2/3)
                sigma2(i,j) = 10^-2;
            end
        end
        if j <= (ny/3) || j >= (ny*2/3)
            if i >= (nx/3) && i <= (nx*2/3)
                sigma3(i,j) = 10^-2;
            end
        end
        if j <= (ny/4) || j >= (ny/2)
            if i >= (nx/4) && i <= (nx/2)
                sigma4(i,j) = 10^-2;
            end
        end
        end
end

figure(6);
subplot(2,2,1)
mesh(sigma1);
xlabel('X');
ylabel('Y');
title('Sigma1 Map');
subplot(2,2,2)
mesh(sigma2);
xlabel('X');
ylabel('Y');
title('Sigma2 Map');
subplot(2,2,3)
mesh(sigma3);
xlabel('X');
ylabel('Y');
title('Sigma3 Map');
subplot(2,2,4)
mesh(sigma4);
xlabel('X');
ylabel('Y');
title('Sigma4 Map');
```

this gives us 4 plots all on the same figure for easy comparison.

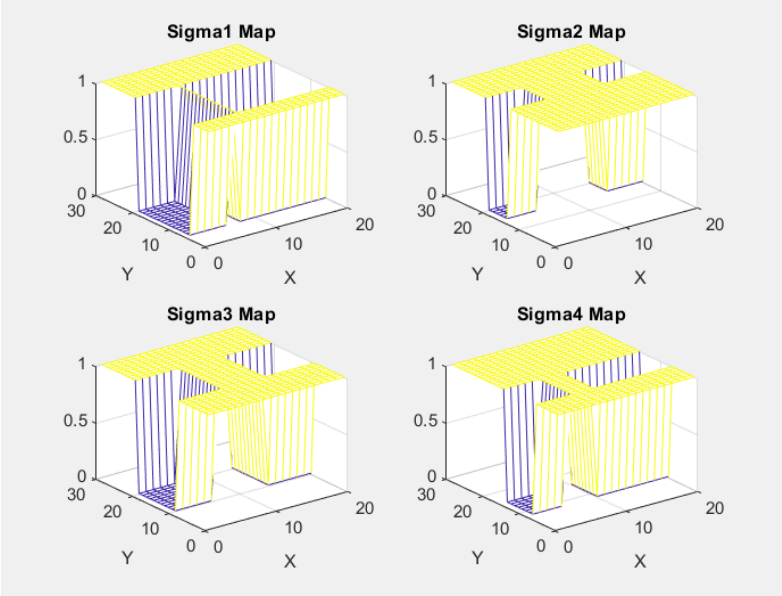

Figure 8: Sigma Maps for Changing "Bottle- Necks"

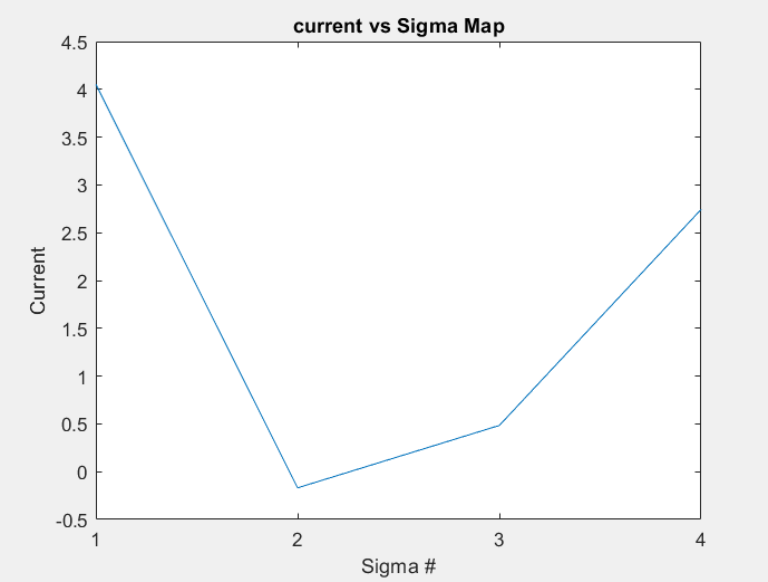

Figure 9:Total Current for Corresponding Sigma Map

The above plots show the “Bottle- necks” changing and the areas affected. The plot below that is the corresponding current for each sigma map. We can see that the smaller the bottle neck area the lower the current. The last part for the assignment was investigating the current while sigma changes. The code for this is shown below.

```
sumJ=0;
oldJ=0;
for s= 1e-12:0.01:0.9
x= 3;            % x dimension
y=2;             % y dimension
%%%%% ratio for L/W is 3/2
```

```
dx = .1;
dy = .1;
nx = x/dx;
ny = y/dy;
%%create matrices for the F and G values.
G = sparse(nx*ny,nx*ny);
F = zeros(nx*ny,1);

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

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

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

Voltmap2 = 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;
        Voltmap2(i, j) = V(n);
    end
end
 for i = 1:nx
    for j = 1:ny
        if i == 1
            Ex(i, j) = Voltmap2(i+1, j) - Voltmap2(i, j);
        elseif i == nx
            Ex(i, j) = Voltmap2(i, j) - Voltmap2(i-1, j);
        else
            Ex(i, j) = (Voltmap2(i+1, j) - Voltmap2(i-1, j))*0.5;
        end
```

```
            if j == 1
                Ey(i, j) = Voltmap2(i, j+1) - Voltmap2(i, j);
            elseif j == ny
                Ey(i, j) = Voltmap2(i, j) - Voltmap2(i, j-1);
            else
                Ey(i, j) = (Voltmap2(i, j+1) - Voltmap2(i, j-1))*0.5;
            end
        end
    end
    Sigma = ones(nx,ny);

    for i = 1:nx
        for j = 1:ny
            if j <= (ny/3) || j >= (ny*2/3)
                if i >= (nx/3) && i <= (nx*2/3)
                    Sigma(i,j) = s;
                end

            end
        end
    end
    oldJ=sumJ;
    TotalE= Ex + Ey;
    J=Sigma .* TotalE;
    sumJ=sum(J,'All');
    figure(7);
    hold on
    plot([s s], [oldJ sumJ])
    %mesh(Sigma);
    xlabel('Sigma');
    ylabel('Current');
    title('Current vs Sigma');
    end
```

this outputs the following plot.

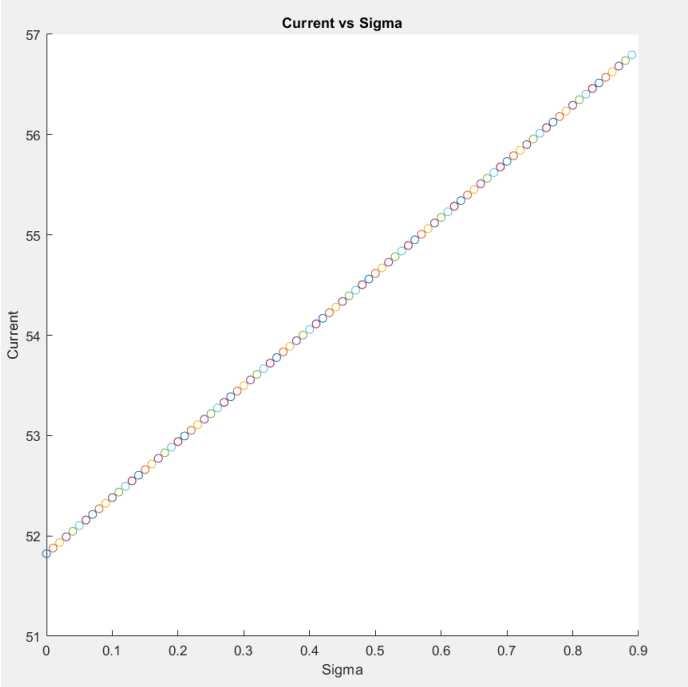

Figure 10: Current vs Sigma

We can see from the plot that as sigma is increased our current decreases.

Conclusion:

In this assignment we are exploring the variables and maps for the finite difference method. Initially we are exploring electrostatic potential over bounds. The second part we are experimenting with the current flow over a bottle- neck area. In both parts we experiment with changing values and finding what the results are. We also compare two different methods of solving a problem. The assignment was completed in Matlab.