MODELLING OF INTEGRATED DEVICES

ELEC 4700

Finite Difference Method

Author: Tony Lam
Student number: 101077253
Date submitted: March 1, 2021

1 Finite difference Method

1.1 Part 1 Ramp

A ramp 1D simulation is created with the boundary condition of 1V at X=0 and 0V at X=L.

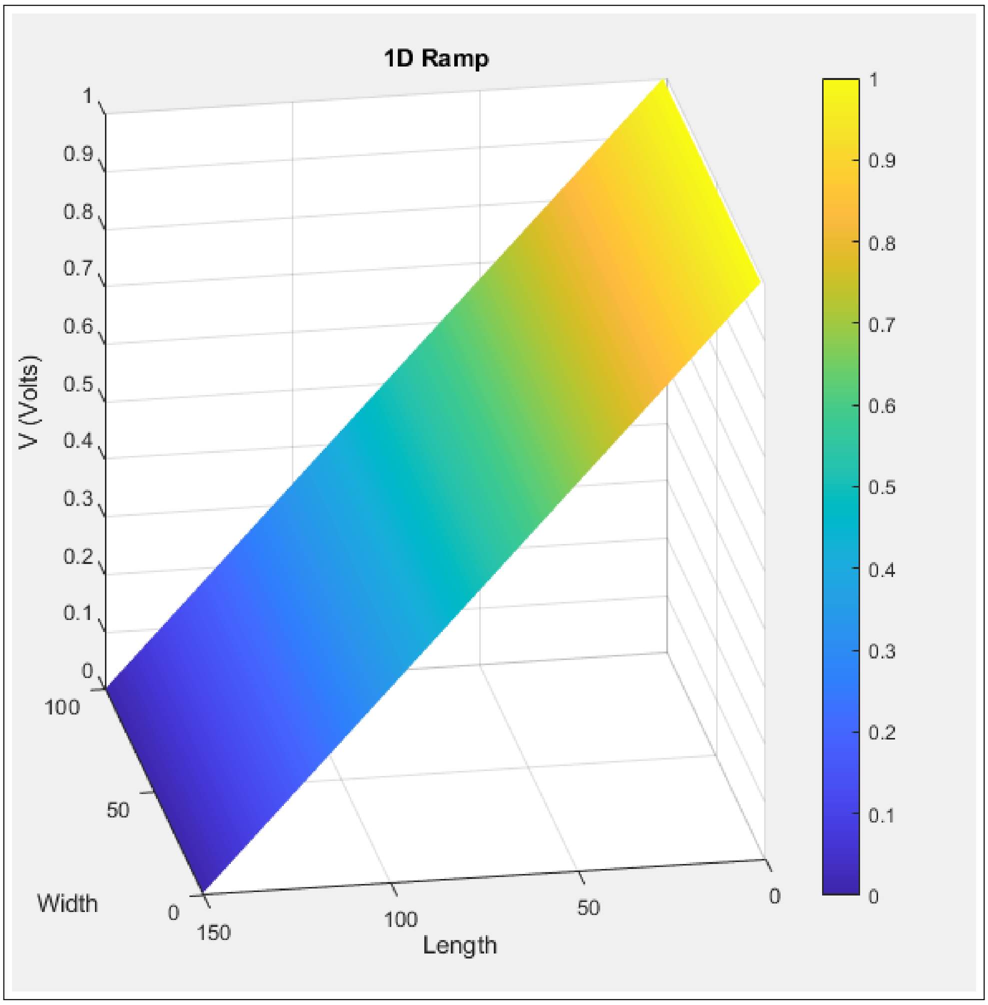

Figure 1: 1-D Ramp Simulation

1.2 Part 1 Saddle

A saddle 2D simulation is created with the boundary X condition of 1V at X=0 and 0V at X=L. The Y boundary condition is 0V at y=0 and Y=W.

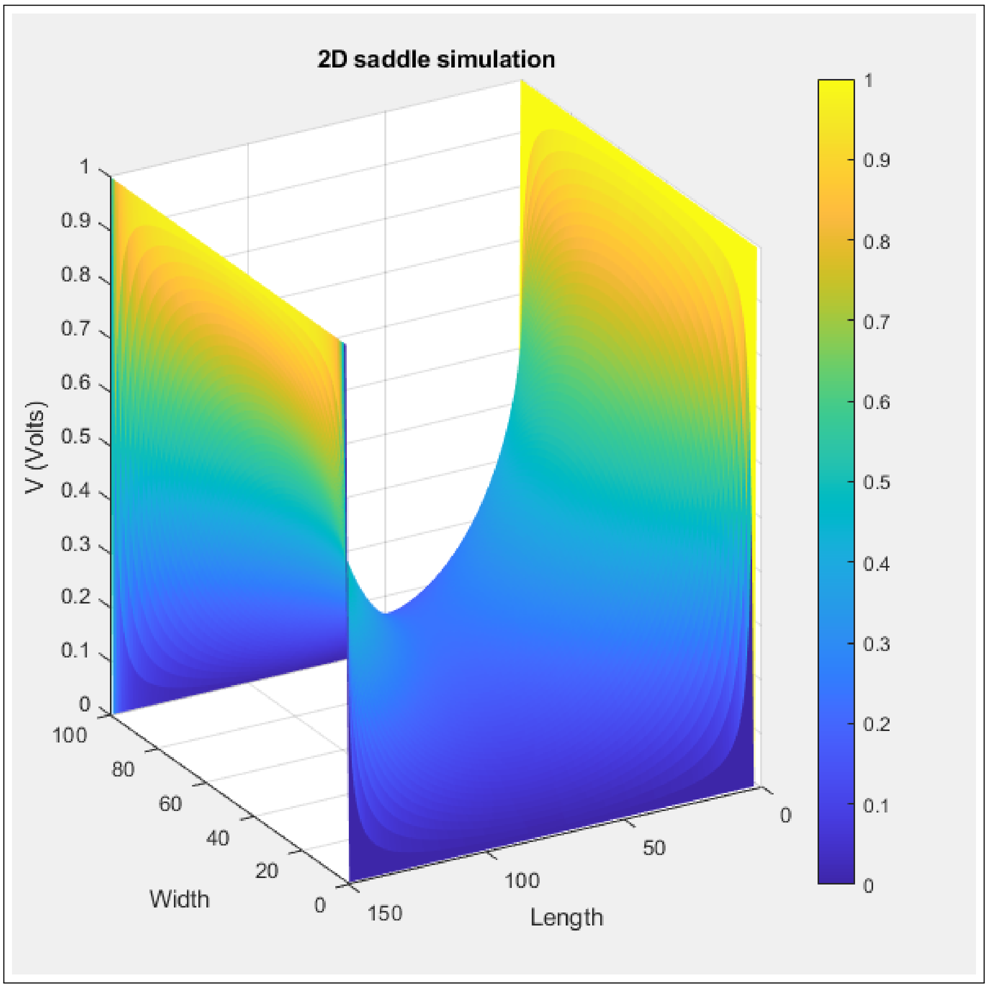

Figure 2: 2-D Saddle Simulation

1.3 Analytical Solution of Saddle

Using the Equation(1) below:

$$V(x,y) = \frac{4V_0}{\pi} \sum_{n=1,3,5..}^{\infty} \frac{1}{n} \frac{Cosh(\frac{n\pi x}{a})}{Cosh(\frac{n\pi b}{a})} sin\left(\frac{n\pi y}{a}\right) \tag{1}$$

The analytical iteration result is shown in the Figure (3) below

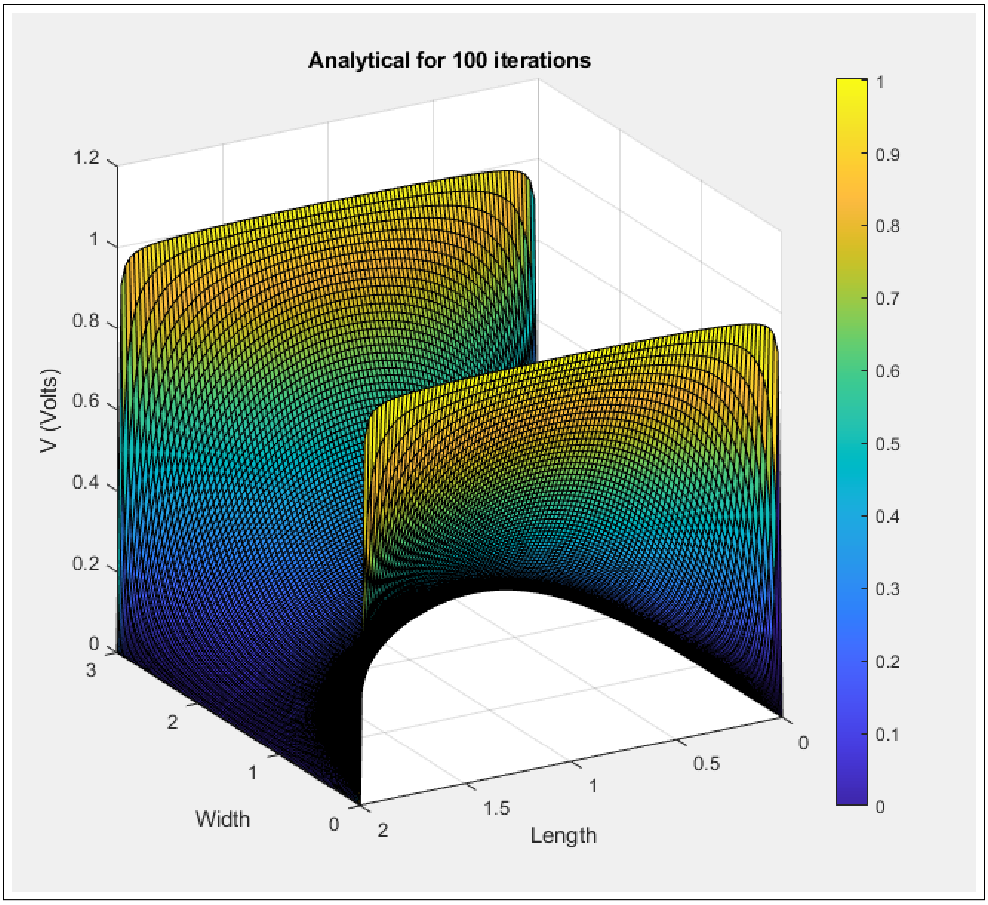

Figure 3: 2-D Saddle Simulation

Meshing can be a trade off between accurate and time. A denser meshing can provide better results but will take longer to produce and a less dense mesh can be inaccurate but much faster for more complex simulations.

Both method is fairly accurate, however the analytical solution took longer to produce and the 1V in the X=0 and X=L is slightly rounded at Y=0 and Y=W.

2 Bottle-neck and current flow

A bottle neck is added to the region where σ is 1 outside the box and $\sigma = 10^{-2}$ inside the box.

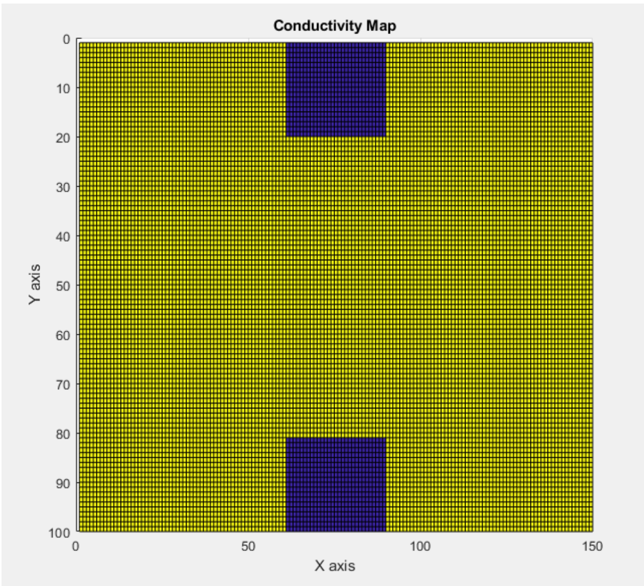

Figure 4: Conduction Map

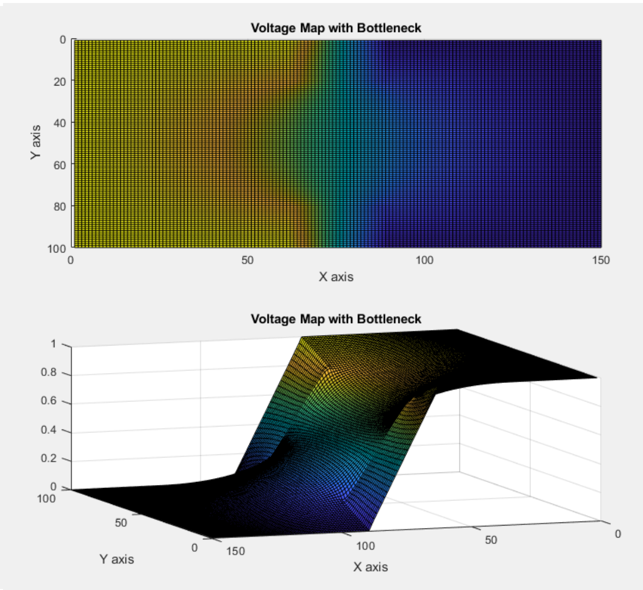

Figure 5: Voltage Map

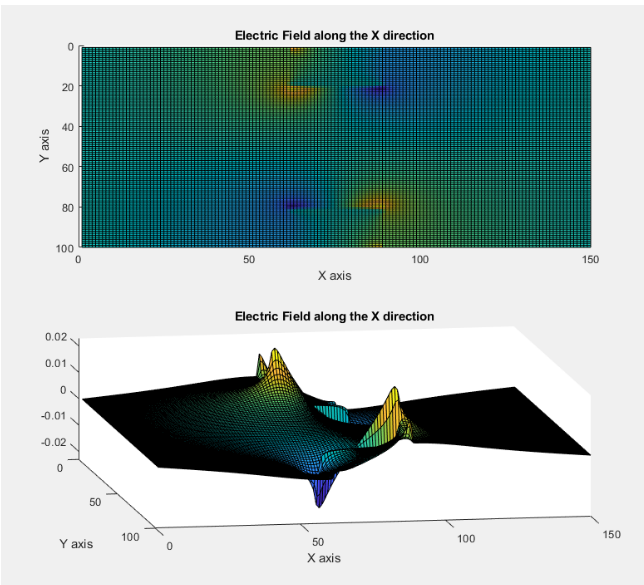

Figure 6: Electric Field along X

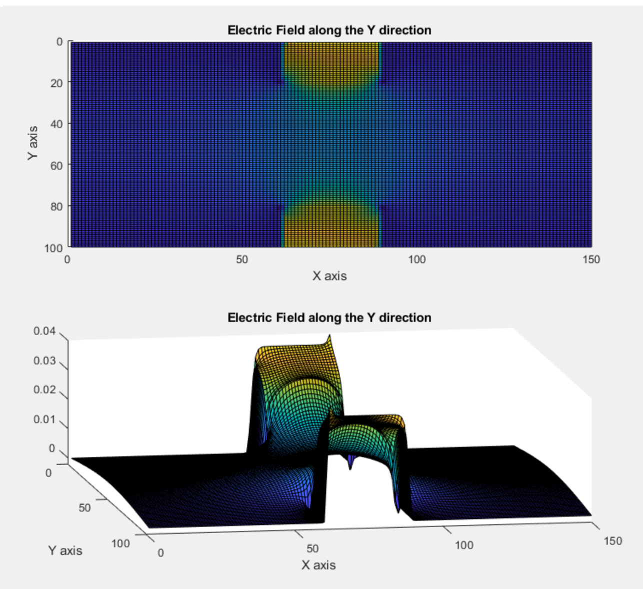

Figure 7: Electric Field along Y

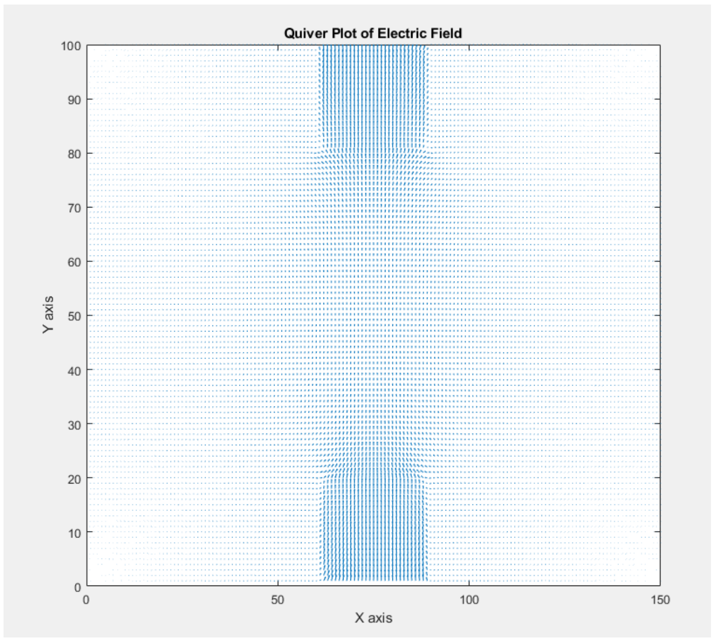

Figure 8: Electric Field

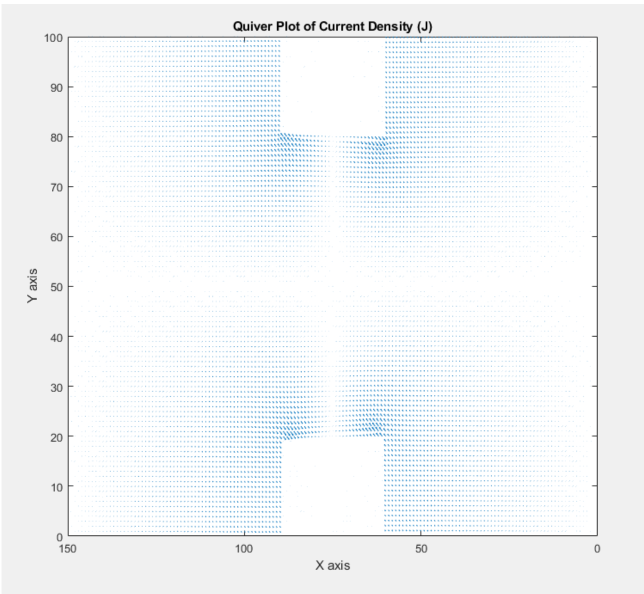

Figure 9: Current Density

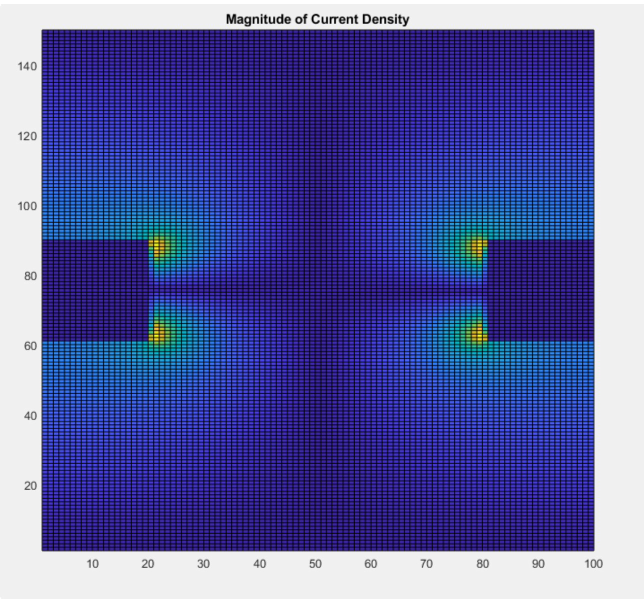

Figure 10: Current Density Magnitude

2.1 Narrowing of the Bottle-neck

The bottle neck is narrowed.

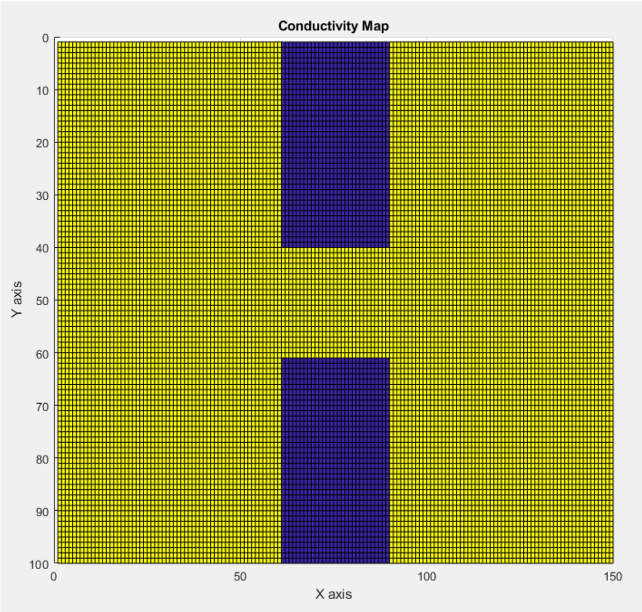

Figure 11: Conduction Map

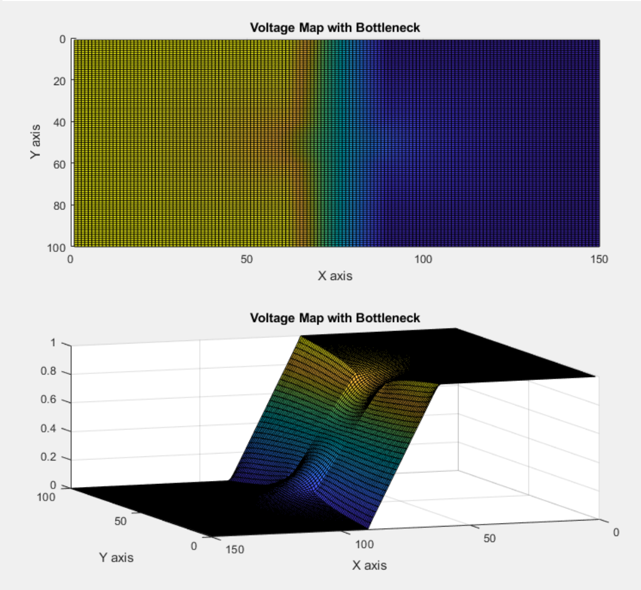

Figure 12: Voltage Map

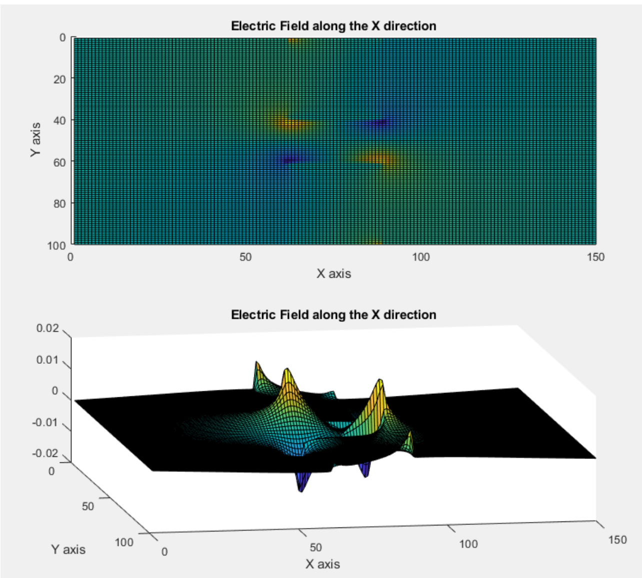

Figure 13: Electric Field along X

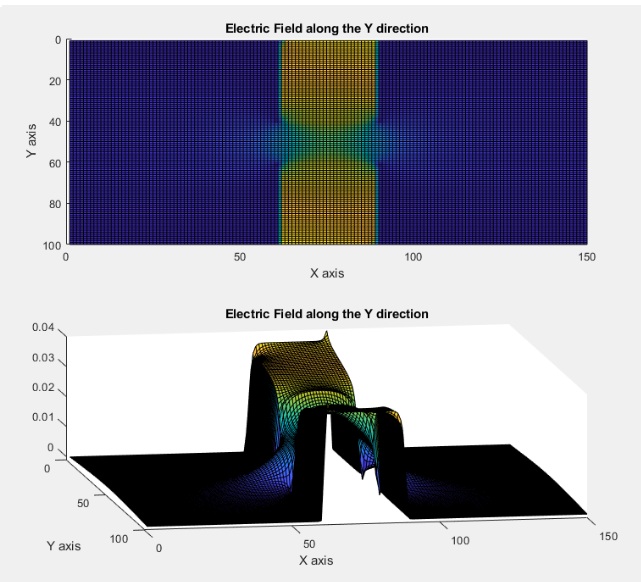

Figure 14: Electric Field along Y

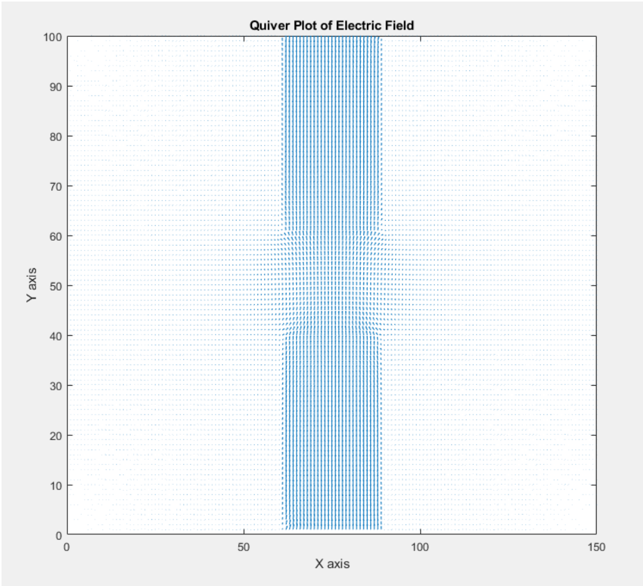

Figure 15: Electric Field

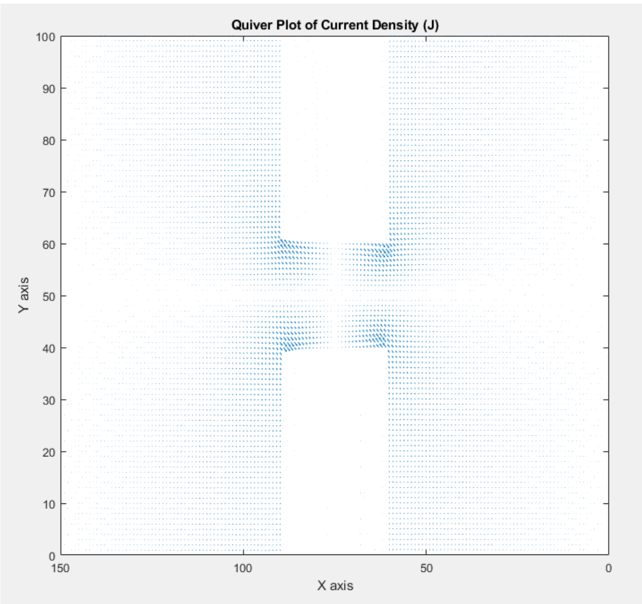

Figure 16: Current Density

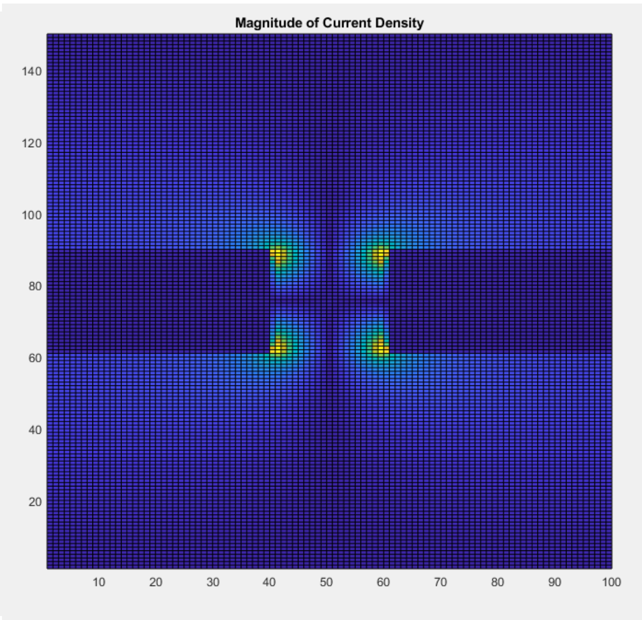

Figure 17: Current Density Magnitude

2.2 Different Delta value

The σ value is kept at 1 outside the box but is change to $\sigma = 10^{+2}$ inside the box.

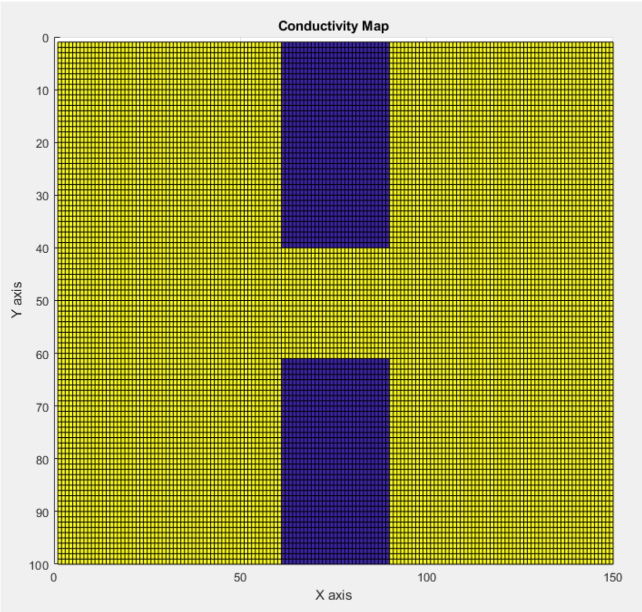

Figure 18: Conduction Map

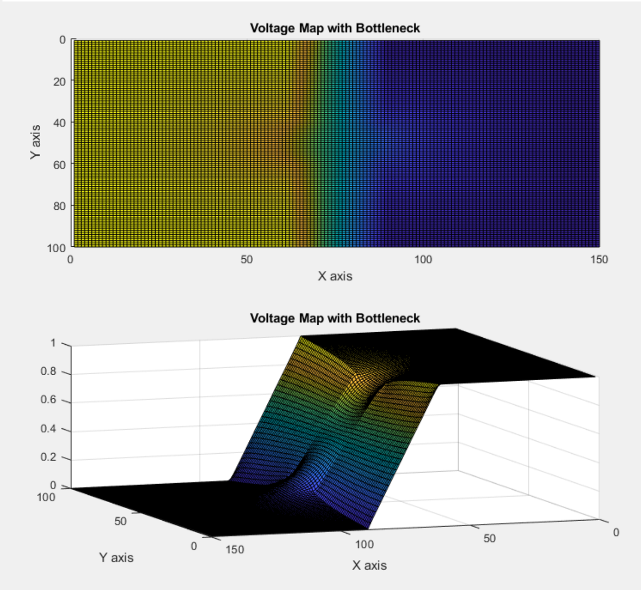

Figure 19: Voltage Map

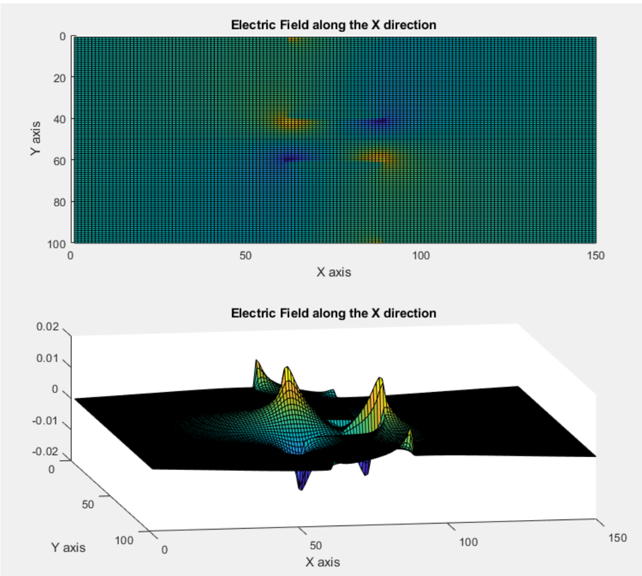

Figure 20: Electric Field along X

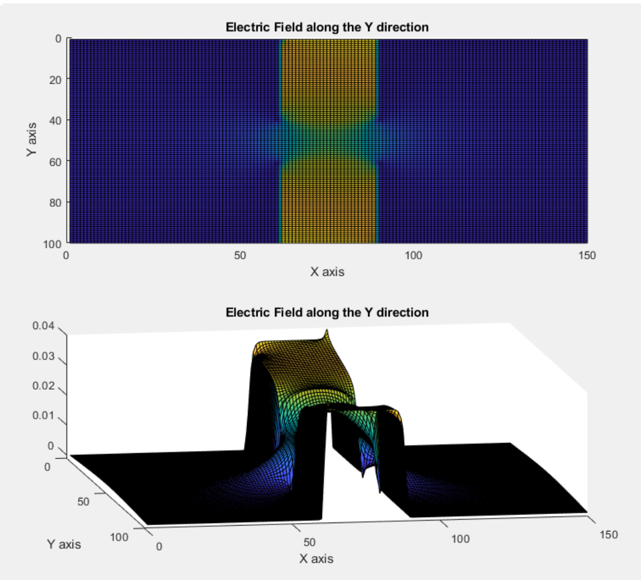

Figure 21: Electric Field along Y

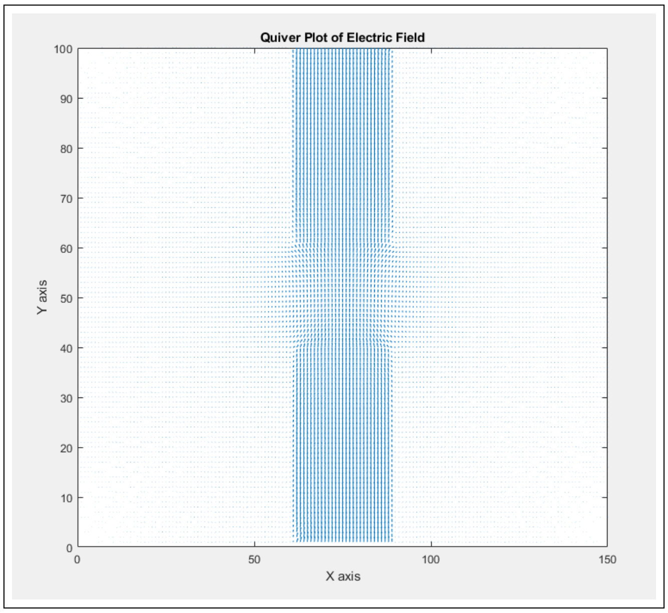

Figure 22: Electric Field

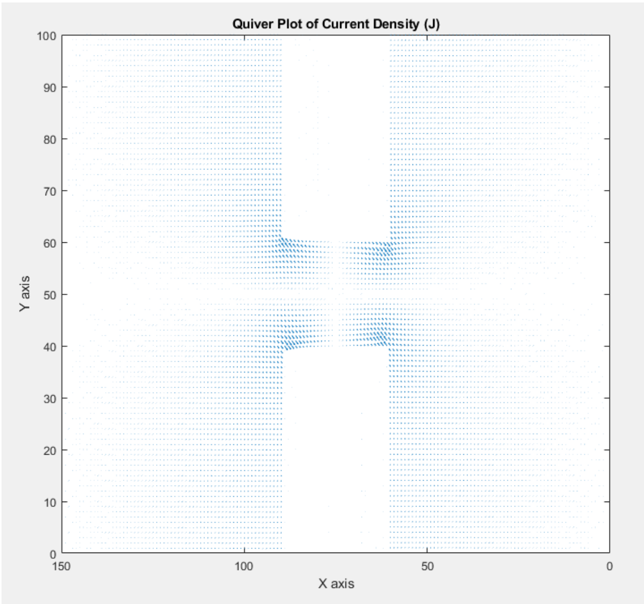

Figure 23: Current Density

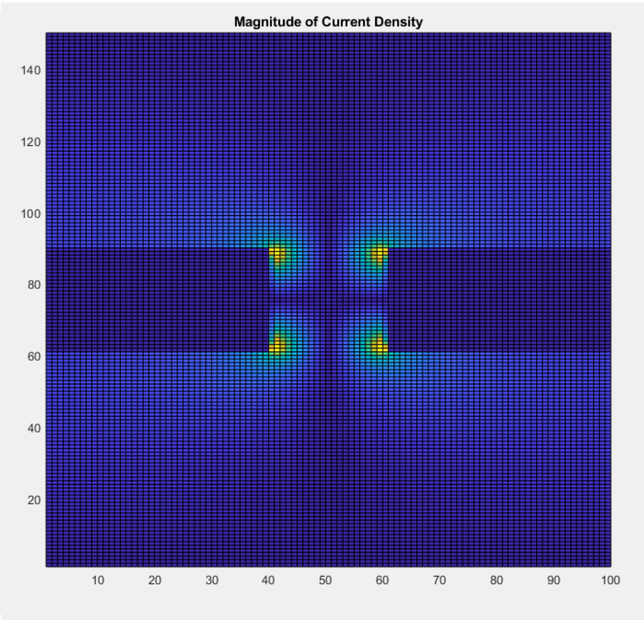

Figure 24: Current Density Magnitude

3 Appendix

3.1 Main Script

```
% clear all
close all
clear
clearvars -GLOBAL
clc
format shorte

set(0, 'DefaultFigureWindowStyle', 'docked')

global L W V0

L = 3;
W = 2;
V0 = 1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Variables to be changed below %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

nx = 150;                    % Number of X elements
ny = 100;                    % Number of Y elements
Iteration = 100;

%===================================%
%========= Part 1A  Ramp ===========%
%===================================%
Part1a_Ramp(nx,ny);

%===================================%
%========= Part 1B  Saddle =========%
%===================================%
Part1b_Saddle(nx,ny,Iteration);


%===================================%
%============= Part 2A =============%
%===================================%

Part2a(nx,ny);

Part2c_Narrowing(nx,ny);

Part2d_DiffSigma(nx,ny);
```

3.2 Function for Part 1 Ramp

```
function [outputArg1,outputArg2] = Part1a_Ramp(nx,ny)

solutMat = zeros(nx,ny);
solutVect = zeros(nx*ny,1);
G = sparse(nx*ny, nx*ny);

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

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

        if i == 1               %Left side
            G(:,n) = 0;
            G(n,n) = 1;
            solutVect(n) = 1;

        elseif i == nx          % Right size
            G(:,n) = 0;
            G(n,n) = 1;

        elseif j == 1           %Bottom y-axis
            G(n,n) = -3;
            G(n,n+ny) = 1;
            G(n,n-ny) = 1;
            G(n,n+1) = 1;

        elseif j == ny          %Top y-axis
            G(n,n) = -3;
            G(n,n+ny) = 1;
            G(n,n-ny) = 1;
            G(n,n-1) = 1;

        else %Handle standard middle cases

            G(n,n) = -4;
            G(n,n+ny) = 1;
            G(n,n-ny) = 1;
            G(n,n+1)= 1;
            G(n,n-1) = 1;
        end
    end
end

solutVect = G\solutVect;

for i = 1:nx
        for j = 1:ny
                n= j+(i-1)*ny;
                solutMat(i,j) = solutVect(n);
        end
```

```
end

figure(1)
surf(solutMat, 'edgecolor', 'none')
colorbar;
title('1D Ramp');
ylabel('Length');
xlabel('Width');
zlabel('V (Volts)');
view(-100,20)

end
```

3.3 Function for Part 1 Saddle

```
function [outputArg1,outputArg2] = Part1b_Saddle(nx,ny,Iteration)
global L W V0

save = zeros(nx,ny);
solution = zeros(nx*ny,1);
G = sparse(nx*ny, nx*ny);

for x = 1:nx
    for y = 1:ny
        n = y + (x-1)*ny;

        if x == 1
            G(n, : ) = 0;
            G(n, n) = 1;
            solution(n) = 1;
        elseif x == nx
            G(n, : ) = 0;
            G(n, n) = 1;
            solution(n) = 1;
        elseif y == 1
            G(n, : ) = 0;
            G(n, n) = 1;
        elseif y == ny
            G(n, : ) = 0;
            G(n, n) = 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

solution = G\solution;

for i = 1:nx
        for j = 1:ny
                n = j+(i-1)*ny;
                save(i,j) = solution(n);
        end
end
figure(2)
surf(save, 'edgecolor', 'none')
colorbar;
title('2D saddle simulation');
ylabel('Length');
xlabel('Width');
zlabel('V (Volts)');
```

```
view(-120,20)

% Analytical solution
A_Solve = zeros(ny, nx);
x2 = repmat(linspace(-L/2,L/2,nx),ny,1);
y2 = repmat(linspace(0,W,ny),nx,1)';

solution = reshape(solution,[],ny)';
for i=1:Iteration
    n = 2*i - 1;
    A_Solve = A_Solve + 1./n.*cosh(n.*pi.*x2./W) ...
        ./cosh(n.*pi.*(L./2)./W).*sin(n.*pi.*y2./W);
end

A_Solve = A_Solve.*4.*V0./pi;

figure(3);
surf(linspace(0,L,nx),linspace(0,W,ny),A_Solve);
colorbar;
title(sprintf('Analytical for %d iterations', Iteration));
ylabel('Length');
xlabel('Width');
zlabel('V (Volts)');
view(-120,20)

end
```

3.4 Part 2 Current flow through a bottle neck

```
function [outputArg1,outputArg2] = Part2c_Narrowing(nx,ny)

global L W V0

save = zeros(nx,ny);
solution = zeros(nx*ny,1);
G = sparse(nx*ny, nx*ny);

%Question 2

G = sparse((nx *ny), (nx*ny));
V = zeros(1,(nx *ny));
Vo = 1;

conductivity1 = 1; %outside
conductivity2 = 1e-2; %inside

box1 = [(nx* 2/5), (nx* 3/5), ny , (ny *4/5)]; %lower box
box2 = [(nx* 2/5), (nx* 3/5),0, (ny *1/5)]; %top box

bottleNeckVec = zeros(1,10);
currentVec = zeros(1,10);

% conductivity map
map = ones(nx,ny);

for i = 1:nx
    for j = 1:ny
        if (i < box1(2) && i >box1(1) && ((j < box2(4)) || (j > box1(4))))
            map(i,j) = conductivity2;
        end
    end
end
% plot the conduction map
figure(4)
surf(map);
title('Conductivity Map');
xlabel('Y axis');
ylabel('X axis');
view(90,90)

%Matrix G
for i = 1:nx

    for j = 1: ny
        n = j+ (i-1) *ny;

        if (i == 1)
            G(n,:) = 0;

```

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

            V(1,n) = 1;

        elseif (i == nx)
            G(n,:) = 0;

            G(n,n) = 1;


        elseif ((i > 1) && (i < nx) && (j==1))

            G(n, n+ny) = map(i+1,j);

            G(n,n) = -(map(i,j+1)+map(i-1,j)+map(i+1,j));

            G(n,n-ny) = map(i-1,j);

            G(n,n-1) = map(i,j+1);

        elseif ((j == ny) && (i < nx) && (i >1))
            G(n, n+ny) = map(i+1,j);

            G(n,n) = -(map(i-1,j)+map(i+1,j)+map(i,j-1));

            G(n,n-ny) = map(i-1,j);

            G(n,n+1) = map(i,j-1);
        else
            G(n, n+ny) = map(i+1,j);

            G(n,n) = -(map(i-1,j)+map(i+1,j)+map(i,j+1)+map(i,j-1));

            G(n,n-ny) = map(i-1,j);

            G(n,n-1) = map(i,j+1);

            G(n,n+1) = map(i,j-1);
        end
    end
end

solution1 = G\V';

Volt = zeros(nx,ny);

for i =1:nx
    for j = 1:ny
        n = j+(i-1) * ny;
        Volt(i,j) = solution1(n);
    end
```

```
end

% Plot of Voltage Map
figure(5);
subplot(2,1,1);
surf(Volt);
title('Voltage Map with Bottleneck');
xlabel('Y axis');
ylabel('X axis');
view(90,90)

subplot(2,1,2);
surf(Volt);
title('Voltage Map with Bottleneck');
xlabel('Y axis');
ylabel('X axis');
view(-110,20)

% Electric Field plot
[Efx,Efy] = gradient(Volt);
Efx = -Efx;
Efy = -Efy;

figure(6)
quiver(Efx',Efy');
title('Quiver Plot of Electric Field')
xlabel('X axis');
ylabel('Y axis');
axis([0 nx 0 ny]);

% Electric Field Y
figure(7)
subplot(2,1,1);
title('Electric Field along the Y direction')
surface(Efy)
xlabel('Y axis');
ylabel('X axis');
axis([0 ny 0 nx]);
view(90,90)

subplot(2,1,2);
title('Electric Field along the Y direction')
surface(Efy)
xlabel('Y axis');
ylabel('X axis');
axis([0 ny 0 nx]);
view(-280,30)

figure(8)
subplot(2,1,1);
title('Electric Field along the X direction')
```

```
surface(Efx)
xlabel('Y axis');
ylabel('X axis');
axis([0 ny 0 nx]);
view(90,90)

subplot(2,1,2);
title('Electric Field along the X direction')
surface(Efx)
xlabel('Y axis');
ylabel('X axis');
axis([0 ny 0 nx]);
view(-280,30)


%plotting current density
Jx = Efx .* map;
Jy = Efx .*map;

figure(9)
quiver(Jx,Jy);
title('Quiver Plot of Current Density (J)')
xlabel('Y axis');
ylabel('X axis');
axis([0 ny 0 nx]);
view(270,90)

% Magnitude of current density
magnitude = ((Jx .^2) + (Jy .^2)) .^0.5;

figure(10)
title('Magnitude of Current Density')
surface(magnitude)
axis tight

currentflow = sum(Jx(nx/2,:));

end
```

3.5 Part 2 Narrowing bottle neck

```
function [outputArg1,outputArg2] = Part2c_Narrowing(nx,ny)

global L W V0

save = zeros(nx,ny);
solution = zeros(nx*ny,1);
G = sparse(nx*ny, nx*ny);

%Question 2

G = sparse((nx *ny), (nx*ny));
V = zeros(1,(nx *ny));
Vo = 1;

conductivity1 = 1; %outside
conductivity2 = 1e-2; %inside

box1 = [(nx* 2/5), (nx* 3/5), ny , (ny *3/5)]; %lower box
box2 = [(nx* 2/5), (nx* 3/5),0, (ny *2/5)]; %top box

bottleNeckVec = zeros(1,10);
currentVec = zeros(1,10);

% conductivity map
map = ones(nx,ny);

for i = 1:nx
    for j = 1:ny
        if (i < box1(2) && i >box1(1) && ((j < box2(4)) || (j > box1(4))))
            map(i,j) = conductivity2;
        end
    end
end
% plot the conduction map
figure(11)
surf(map);
title('Conductivity Map');
xlabel('Y axis');
ylabel('X axis');
view(90,90)

%Matrix G
for i = 1:nx

    for j = 1: ny
        n = j+ (i-1) *ny;

        if (i == 1)
            G(n,:) = 0;

```

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

            V(1,n) = 1;

        elseif (i == nx)
            G(n,:) = 0;

            G(n,n) = 1;


        elseif ((i > 1) && (i < nx) && (j==1))

            G(n, n+ny) = map(i+1,j);

            G(n,n) = -(map(i,j+1)+map(i-1,j)+map(i+1,j));

            G(n,n-ny) = map(i-1,j);

            G(n,n-1) = map(i,j+1);

        elseif ((j == ny) && (i < nx) && (i >1))
            G(n, n+ny) = map(i+1,j);

            G(n,n) = -(map(i-1,j)+map(i+1,j)+map(i,j-1));

            G(n,n-ny) = map(i-1,j);

            G(n,n+1) = map(i,j-1);
        else
            G(n, n+ny) = map(i+1,j);

            G(n,n) = -(map(i-1,j)+map(i+1,j)+map(i,j+1)+map(i,j-1));

            G(n,n-ny) = map(i-1,j);

            G(n,n-1) = map(i,j+1);

            G(n,n+1) = map(i,j-1);
        end
    end
end

solution1 = G\V';

Volt = zeros(nx,ny);

for i =1:nx
    for j = 1:ny
        n = j+(i-1) * ny;
        Volt(i,j) = solution1(n);
    end
```

```
end

% Plot of Voltage Map
figure(12)
subplot(2,1,1);
surf(Volt);
title('Voltage Map with Bottleneck');
xlabel('Y axis');
ylabel('X axis');
view(90,90)

subplot(2,1,2);
surf(Volt);
title('Voltage Map with Bottleneck');
xlabel('Y axis');
ylabel('X axis');
view(-110,20)

% Electric Field plot
[Efx,Efy] = gradient(Volt);
Efx = -Efx;
Efy = -Efy;

figure(13)
quiver(Efx',Efy');
title('Quiver Plot of Electric Field')
xlabel('X axis');
ylabel('Y axis');
axis([0 nx 0 ny]);

% Electric Field Y
figure(14)
subplot(2,1,1);
title('Electric Field along the Y direction')
surface(Efy)
xlabel('Y axis');
ylabel('X axis');
axis([0 ny 0 nx]);
view(90,90)

subplot(2,1,2);
title('Electric Field along the Y direction')
surface(Efy)
xlabel('Y axis');
ylabel('X axis');
axis([0 ny 0 nx]);
view(-280,30)

figure(15)
subplot(2,1,1);
title('Electric Field along the X direction')
```

```
surface(Efx)
xlabel('Y axis');
ylabel('X axis');
axis([0 ny 0 nx]);
view(90,90)

subplot(2,1,2);
title('Electric Field along the X direction')
surface(Efx)
xlabel('Y axis');
ylabel('X axis');
axis([0 ny 0 nx]);
view(-280,30)


%plotting current density
Jx = Efx .* map;
Jy = Efx .*map;

figure(16)
quiver(Jx,Jy);
title('Quiver Plot of Current Density (J)')
xlabel('Y axis');
ylabel('X axis');
axis([0 ny 0 nx]);
view(270,90)

% Magnitude of current density
magnitude = ((Jx .^2) + (Jy .^2)) .^0.5;

figure(17)
title('Magnitude of Current Density')
surface(magnitude)
axis tight

currentflow = sum(Jx(nx/2,:));

end
```

3.6 Part 2 Different Delta value

```
function [outputArg1,outputArg2] = Part2d_DiffSigma(nx,ny)

global L W V0

save = zeros(nx,ny);
solution = zeros(nx*ny,1);
G = sparse(nx*ny, nx*ny);

%Question 2

G = sparse((nx *ny), (nx*ny));
V = zeros(1,(nx *ny));
Vo = 1;

conductivity1 = 1; %outside
conductivity2 = 1e2; %inside

box1 = [(nx* 2/5), (nx* 3/5), ny , (ny *3/5)]; %lower box
box2 = [(nx* 2/5), (nx* 3/5),0, (ny *2/5)]; %top box

bottleNeckVec = zeros(1,10);
currentVec = zeros(1,10);

% conductivity map
map = ones(nx,ny);

for i = 1:nx
    for j = 1:ny
        if (i < box1(2) && i >box1(1) && ((j < box2(4)) || (j > box1(4))))
            map(i,j) = conductivity2;
        end
    end
end
% plot the conduction map
figure(18)
surf(map);
title('Conductivity Map');
xlabel('Y axis');
ylabel('X axis');
view(90,90)

%Matrix G
for i = 1:nx

    for j = 1: ny
        n = j+ (i-1) *ny;

        if (i == 1)
            G(n,:) = 0;

```

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

            V(1,n) = 1;

        elseif (i == nx)
            G(n,:) = 0;

            G(n,n) = 1;


        elseif ((i > 1) && (i < nx) && (j==1))

            G(n, n+ny) = map(i+1,j);

            G(n,n) = -(map(i,j+1)+map(i-1,j)+map(i+1,j));

            G(n,n-ny) = map(i-1,j);

            G(n,n-1) = map(i,j+1);

        elseif ((j == ny) && (i < nx) && (i >1))
            G(n, n+ny) = map(i+1,j);

            G(n,n) = -(map(i-1,j)+map(i+1,j)+map(i,j-1));

            G(n,n-ny) = map(i-1,j);

            G(n,n+1) = map(i,j-1);
        else
            G(n, n+ny) = map(i+1,j);

            G(n,n) = -(map(i-1,j)+map(i+1,j)+map(i,j+1)+map(i,j-1));

            G(n,n-ny) = map(i-1,j);

            G(n,n-1) = map(i,j+1);

            G(n,n+1) = map(i,j-1);
        end
    end
end

solution1 = G\V';

Volt = zeros(nx,ny);

for i =1:nx
    for j = 1:ny
        n = j+(i-1) * ny;
        Volt(i,j) = solution1(n);
    end
```

```
end

% Plot of Voltage Map
figure(19)
subplot(2,1,1);
surf(Volt);
title('Voltage Map with Bottleneck');
xlabel('Y axis');
ylabel('X axis');
view(90,90)

subplot(2,1,2);
surf(Volt);
title('Voltage Map with Bottleneck');
xlabel('Y axis');
ylabel('X axis');
view(-110,20)

% Electric Field plot
[Efx,Efy] = gradient(Volt);
Efx = -Efx;
Efy = -Efy;

figure(20)
quiver(Efx',Efy');
title('Quiver Plot of Electric Field')
xlabel('X axis');
ylabel('Y axis');
axis([0 nx 0 ny]);

% Electric Field Y
figure(21)
subplot(2,1,1);
title('Electric Field along the Y direction')
surface(Efy)
xlabel('Y axis');
ylabel('X axis');
axis([0 ny 0 nx]);
view(90,90)

subplot(2,1,2);
title('Electric Field along the Y direction')
surface(Efy)
xlabel('Y axis');
ylabel('X axis');
axis([0 ny 0 nx]);
view(-280,30)

figure(22)
subplot(2,1,1);
title('Electric Field along the X direction')
```

```
surface(Efx)
xlabel('Y axis');
ylabel('X axis');
axis([0 ny 0 nx]);
view(90,90)

subplot(2,1,2);
title('Electric Field along the X direction')
surface(Efx)
xlabel('Y axis');
ylabel('X axis');
axis([0 ny 0 nx]);
view(-280,30)


%plotting current density
Jx = Efx .* map;
Jy = Efx .*map;

figure(23)
quiver(Jx,Jy);
title('Quiver Plot of Current Density (J)')
xlabel('Y axis');
ylabel('X axis');
axis([0 ny 0 nx]);
view(270,90)

% Magnitude of current density
magnitude = ((Jx .^2) + (Jy .^2)) .^0.5;

figure(24)
title('Magnitude of Current Density')
surface(magnitude)
axis tight

currentflow = sum(Jx(nx/2,:));

end
```