ELEC 4700 Assignment-2 Finite Difference Method

Spencer Tigere 101001717

February 28, 2021

Introduction

The purpose of this lab was to examine the Finite Difference Method. Laplace's equation by FD can be used to solve electrostatic potential problems or current flow problems in homogeneous solids; these functionalities were explored. There are two parts to the lab. The first part models voltage in a rectangular region, by use of numerical and analytical methods. Both methods were compared. The second part examines the introduction of a high resistance bottleneck into the rectangular region and the effect this has on the voltage.

Part 1

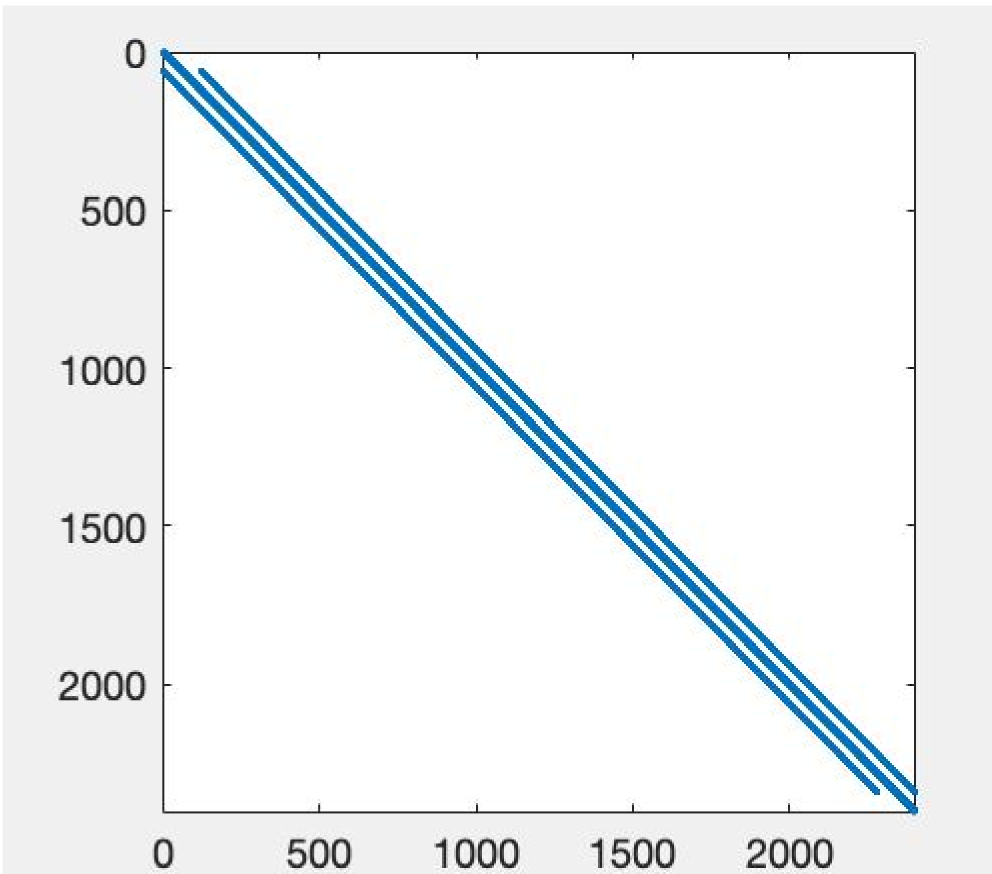

Figure 1. Voltage Model - No fixed BC's

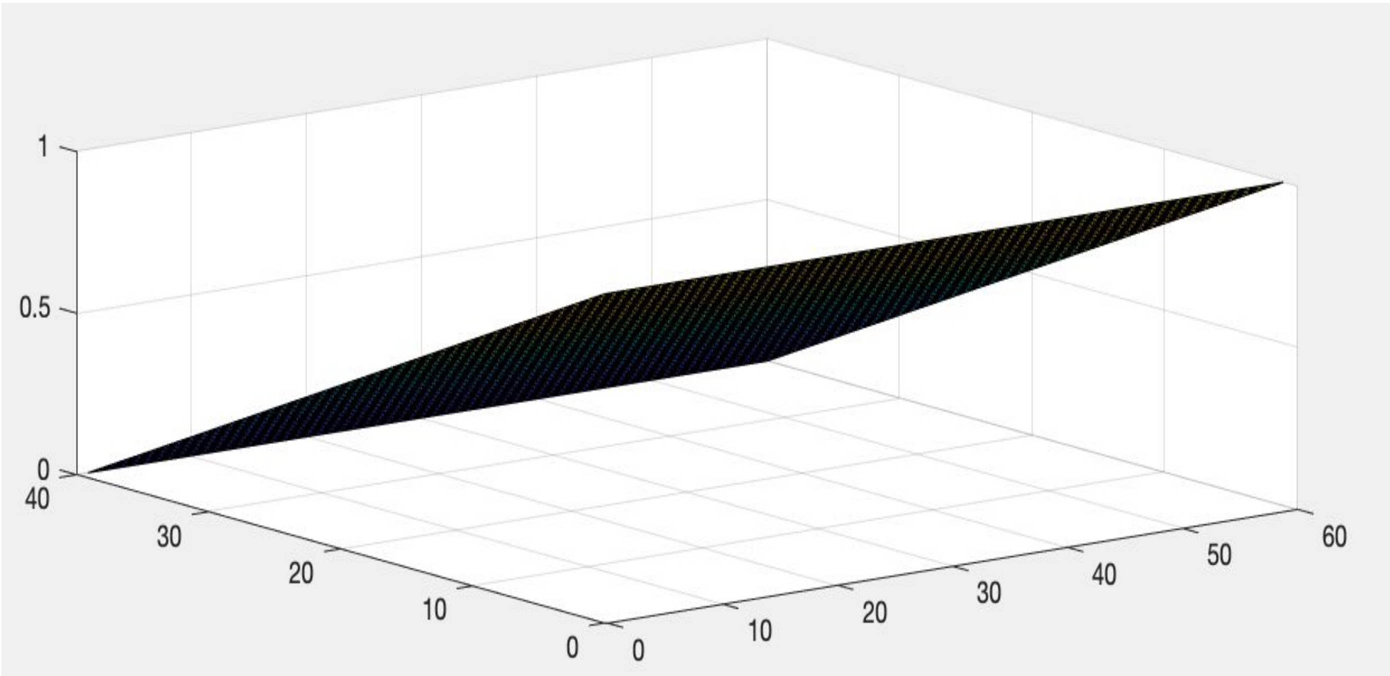

Figure 2. Voltage Model - 2 fixed BC's

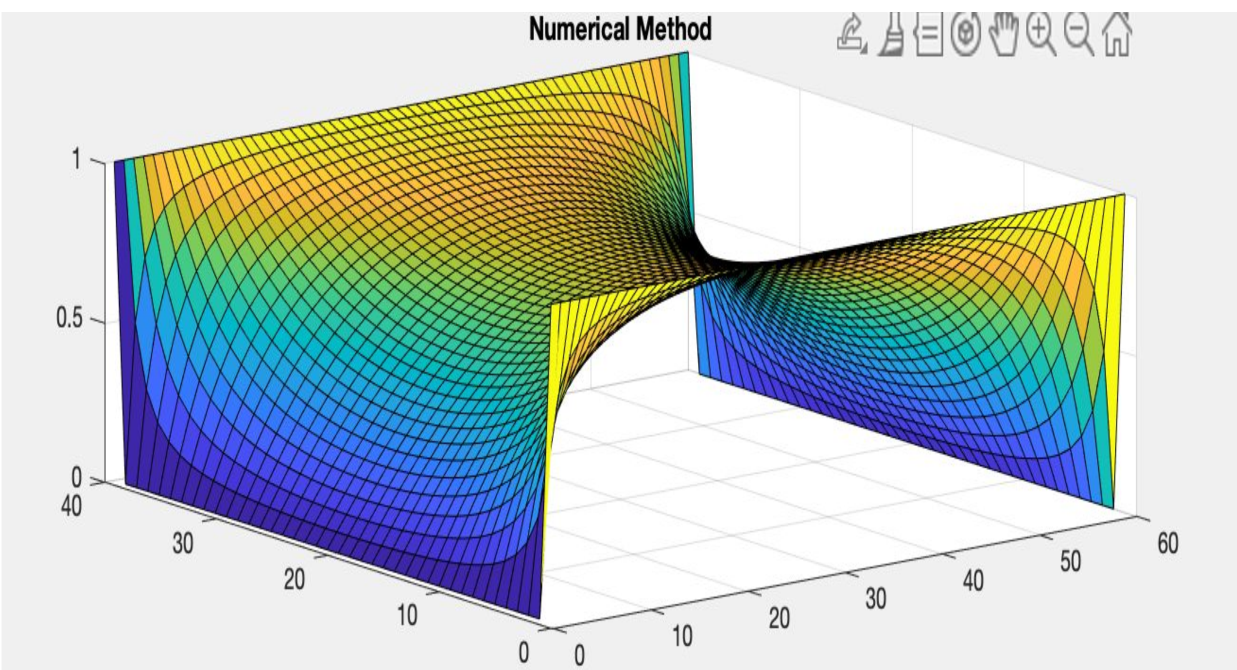

Figure 3. Numerical Method

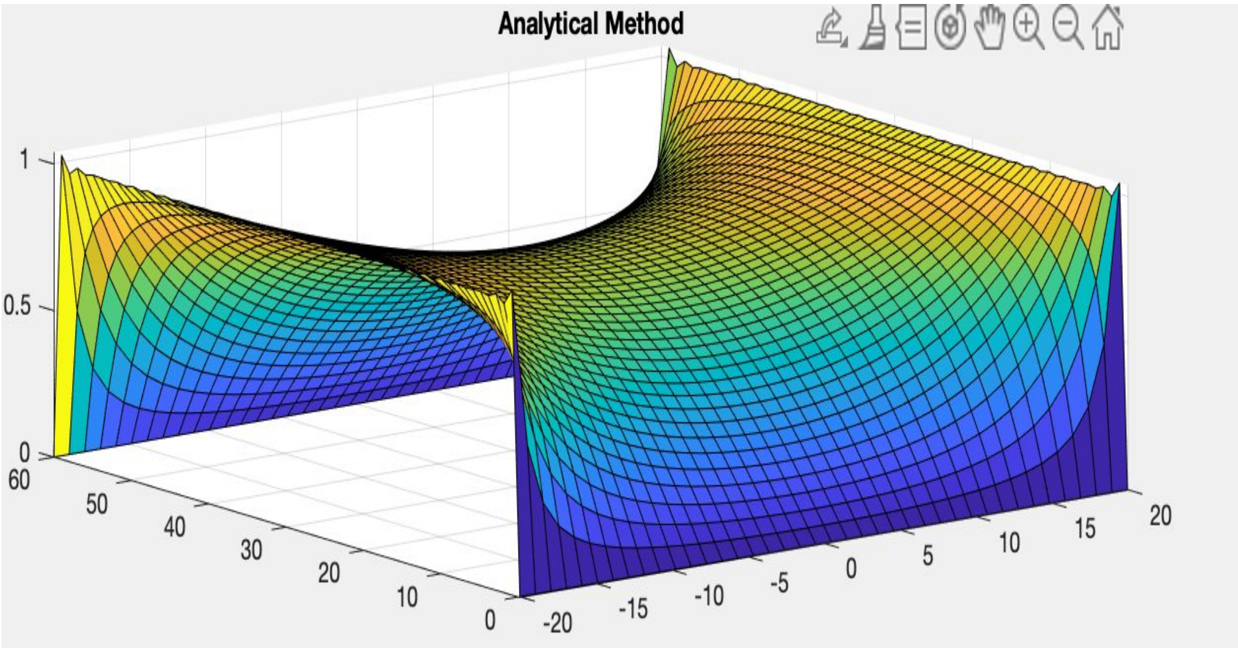

Figure 4. Analytical Method

Comparing Numerical and Analytical methods

It can be observed that both the Numerical and Analytical methods produce similar results. The advantages and disadvantages of both are listed below

Numerical Method Advantages	Numerical Method Disadvantages	Analytical Method Advantages	Analytical Method Disadvantages
-Smaller step size produces more accurate results -No summations	-Method includes many approximations that can decrease accuracy	- Less approximations than Numerical Method	- Method includes many summations which makes it difficult to know when to stop the additions.

Part 2

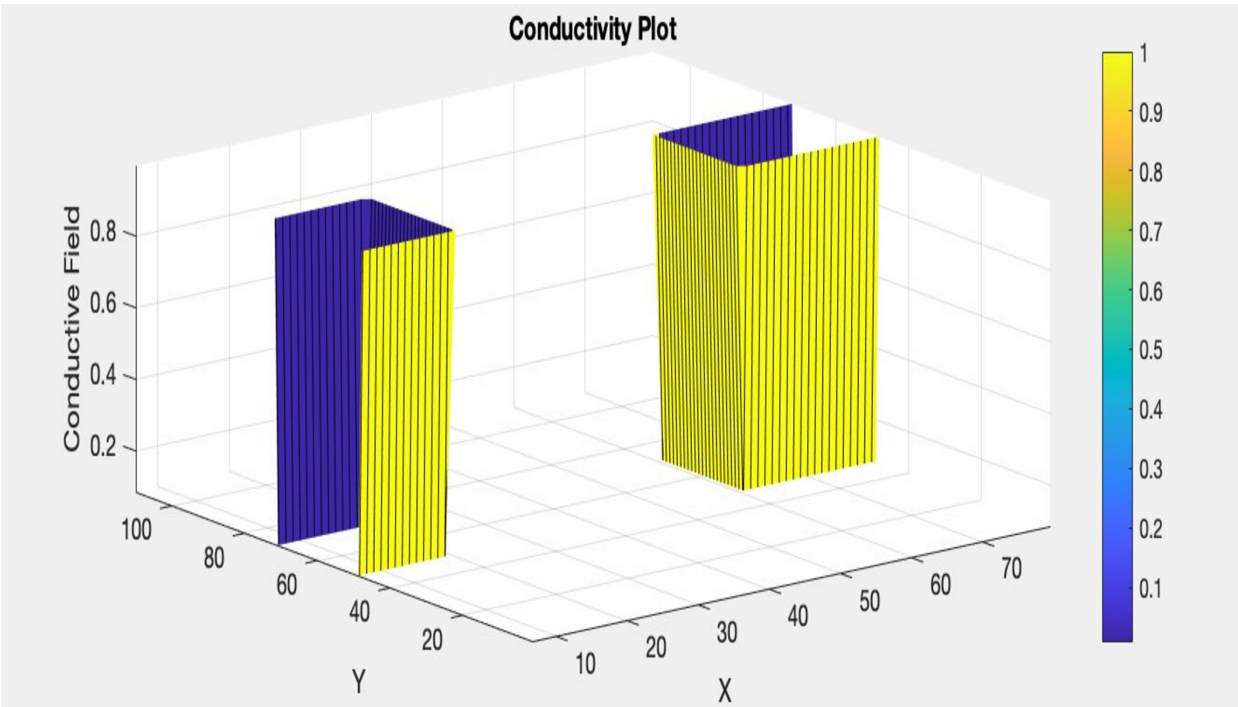

Figure 5. Conductivity Plot

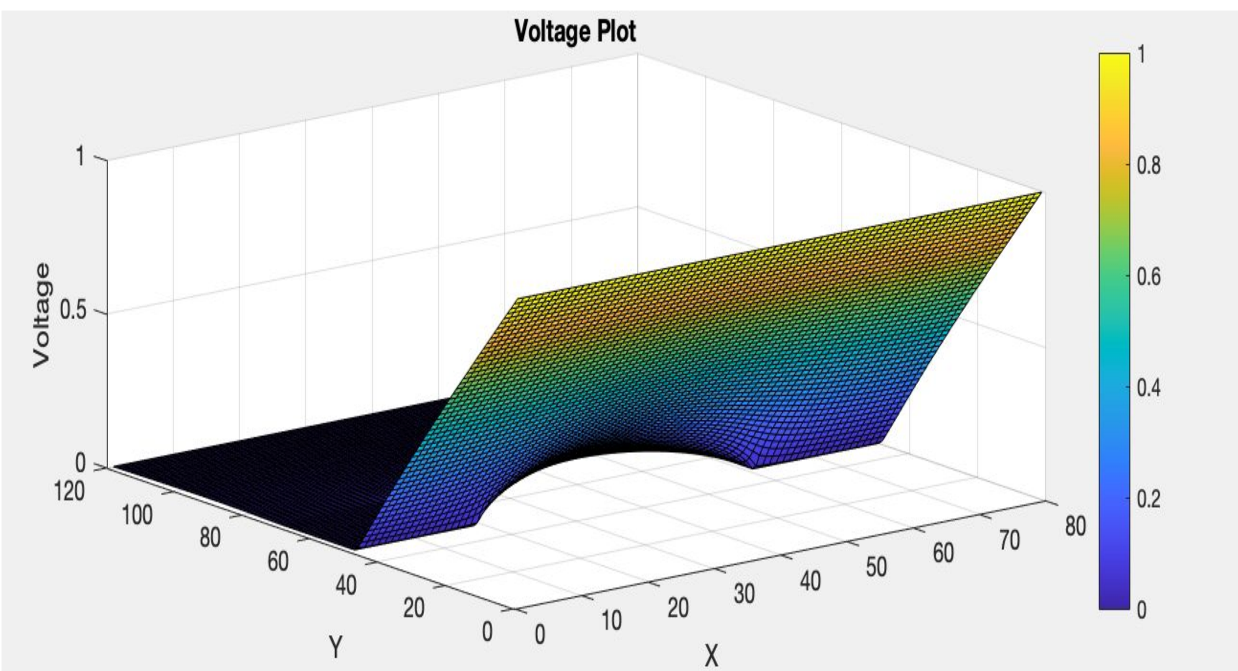

Figure 6. Voltage Plot

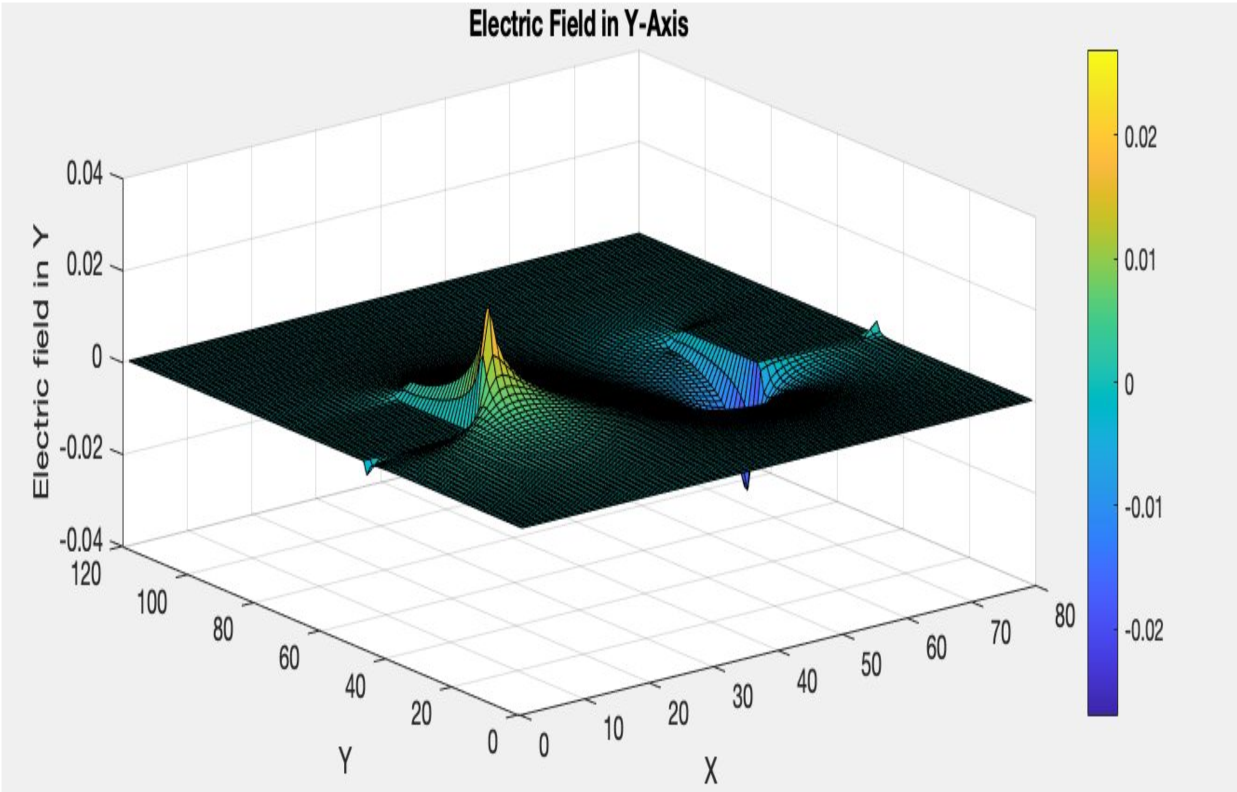

Figure 7. E-Field in Y-Axis

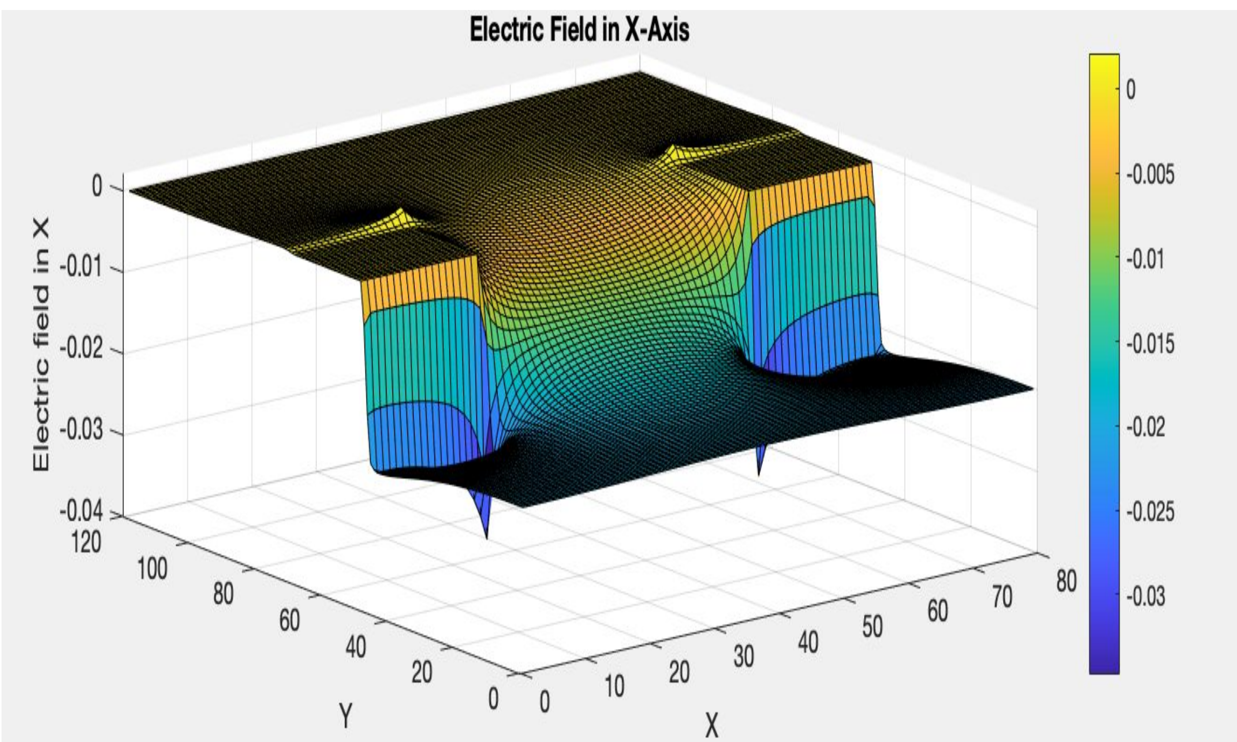

Figure 8. E-Field in X-Axis

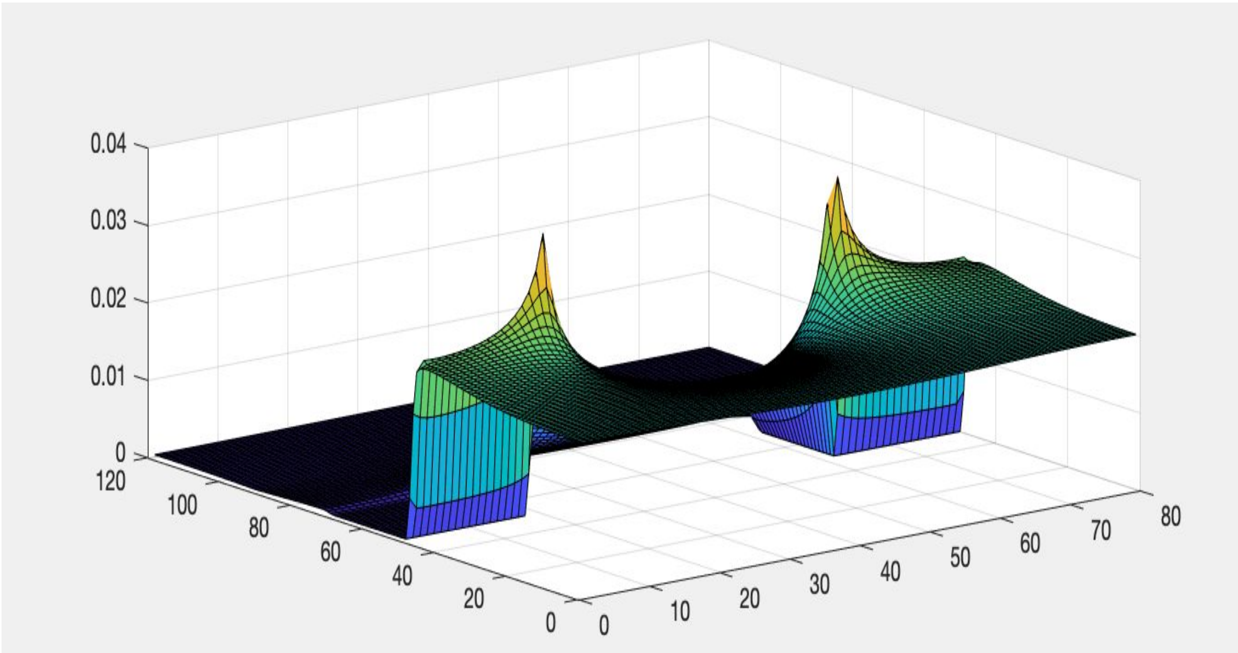

Figure 9. Electric Field (x,y)

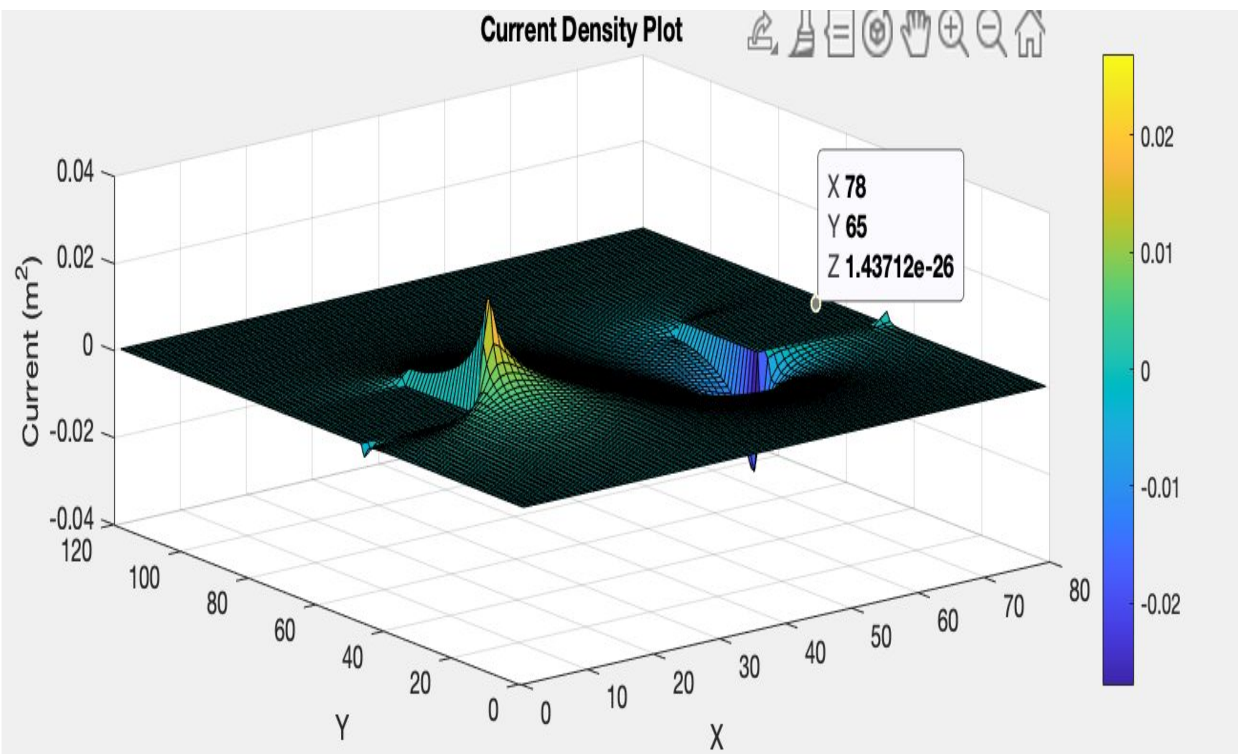

Figure 10. Current Density Plot

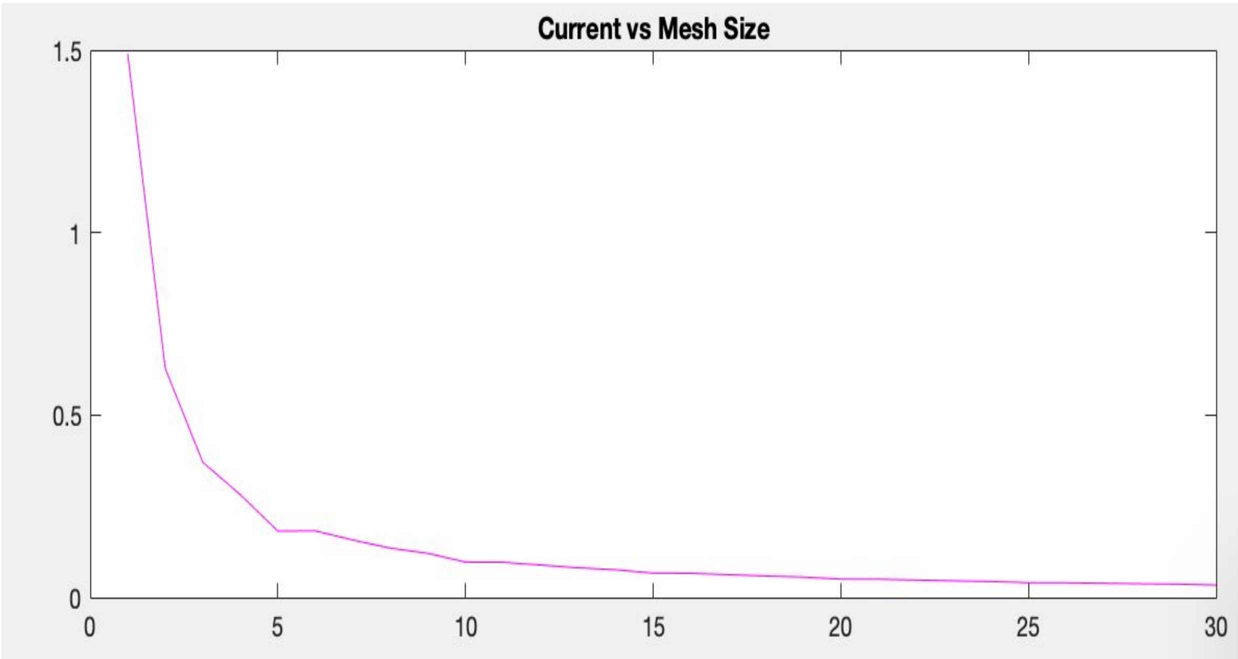

Figure 11. Current vs Mesh Size

Figure 11 shows that as the mesh size increases, the current decreases.

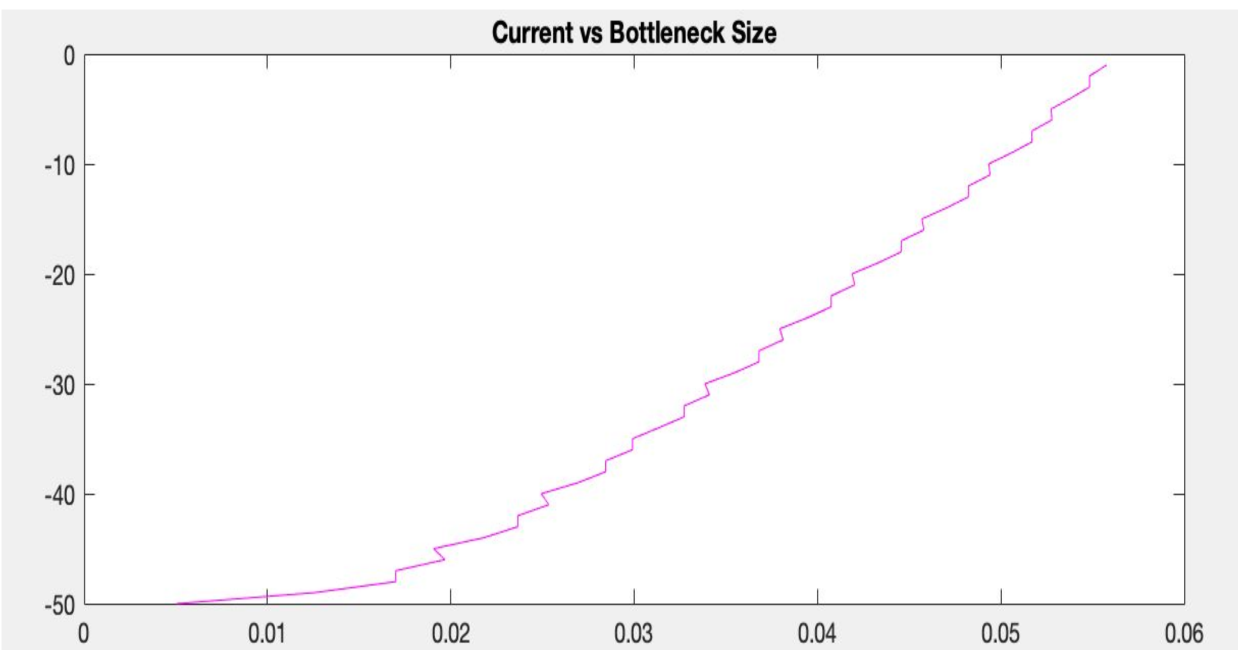

Figure 12. Current vs Bottleneck Size

Figure 12 shows that as the bottleneck width increases, the current increases.

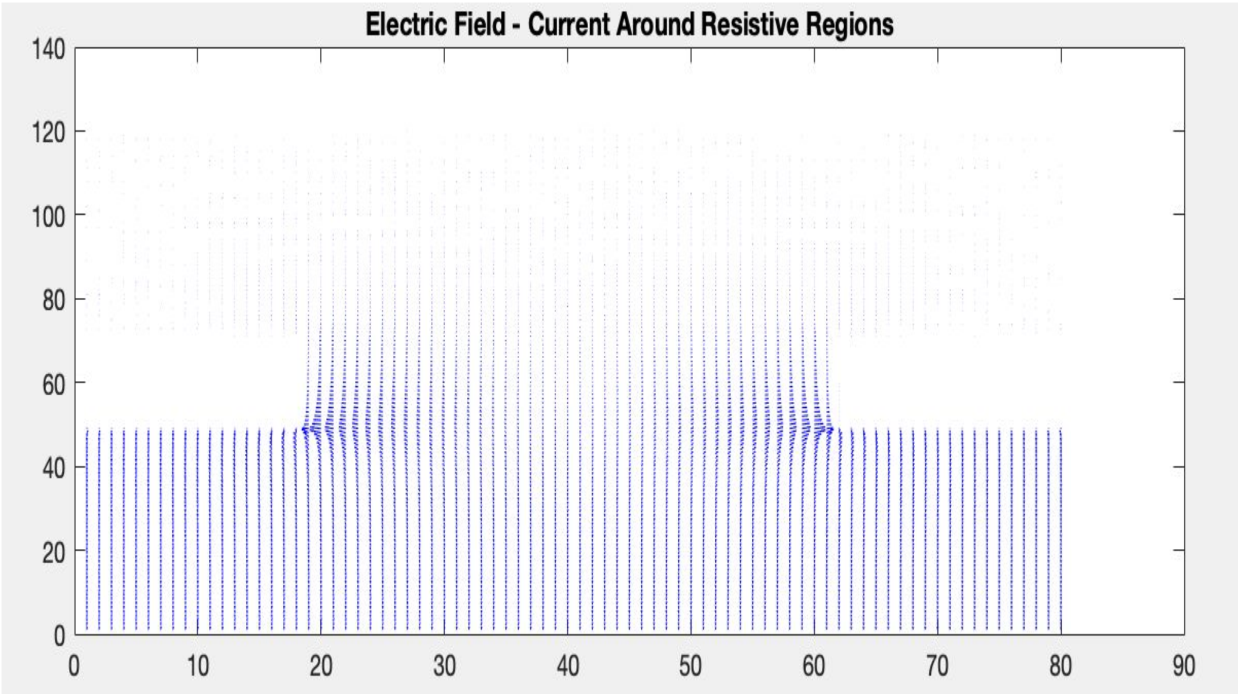

Figure 13. Current around resistive regions

From figure 13, we can see that no current flows in the highly resistance region which is expected.

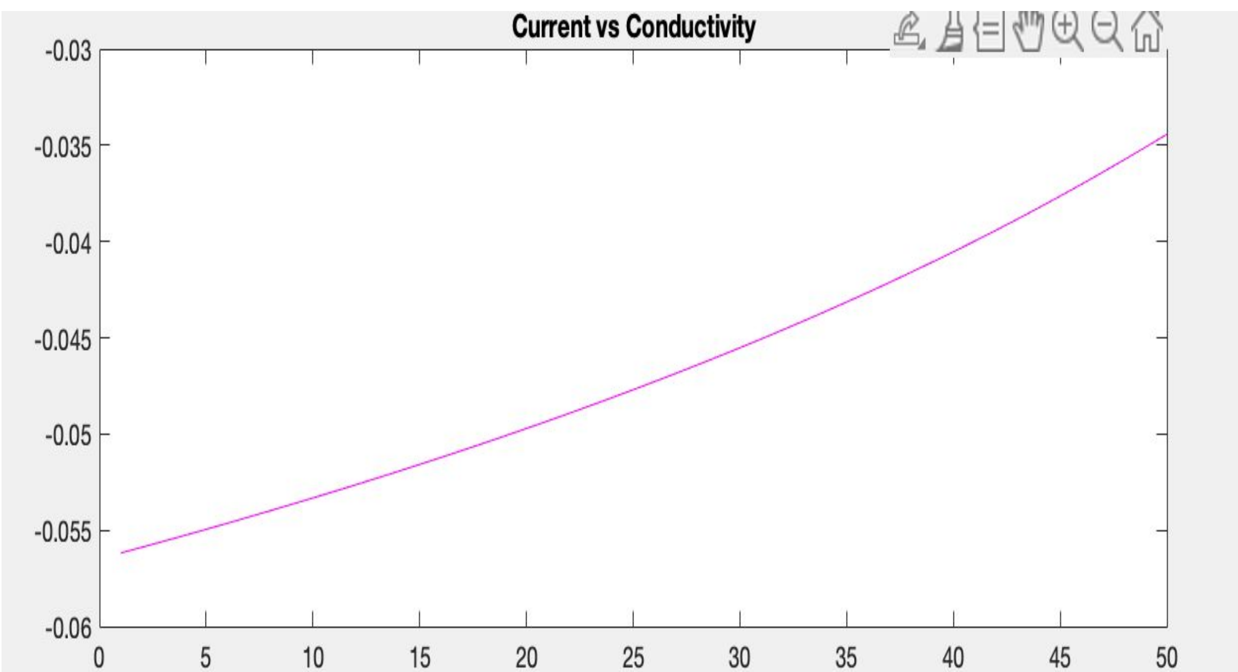

Figure 14. Current vs Conductivity

Conclusion

The lab was successfully completed. The requirements and observations specified in the lab manual were completed and noted in the lab report.

Appendix

```
%ELEC 4700 Assignment 2
%Spencer Tigere 101001717
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%PART 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
clear
set(0,'DefaultFigureWindowStyle','docked')
width_x = 40;
length_y = 60;
G_M = sparse((width_x * length_y), (width_x * length_y));
V_M = zeros(1, (width_x * length_y));
v0 = 1;

for i = 1:width_x
    for j = 1:length_y
    n = j + (i - 1) * length_y;
    nxm = j + ((i-1) - 1) * length_y;
    nxp = j + ((i+1) - 1) * length_y;
    nym = (j-1) + (i - 1) * length_y;
    nyp = (j+1) + (i - 1) * length_y;
    if (i == 1)
    G_M(n, :) = 0;
    G_M(n, n) = 1;
    V_M(1, n) = 1;
    elseif (i == width_x)
    G_M(n, :) = 0;
    G_M(n, n) = 1;
    elseif (j == 1 && i > 1 && i < width_x)
    G_M(n, n) = -1;
    G_M(n, nyp) = 1;
    elseif (j == length_y && i > 1 && i < width_x)
    G_M(n, n) = -1;
    G_M(n, nym) = 1;
    else
    G_M(n, n) = -4;
    G_M(n, nxm) = 1;
    G_M(n, nxp) = 1;
```

```
            G_M(n, nym) = 1;
            G_M(n, nyp) = 1;
            end
            end


end
figure (1);
spy(G_M);
%G and V matrix
G_V = G_M\V_M';
figure (2);
surface0 = zeros(width_x, length_y);

for i = 1:width_x
            for j = 1:length_y
            n = j + (i - 1) * length_y;
            nxm = j + ((i-1) - 1) * length_y;
            nxp = j + ((i+1) - 1) * length_y;
            nym = (j-1) + (i - 1) * length_y;
            nyp = (j+1) + (i - 1) * length_y;
            surface0(i, j) = G_V(n);
            end
end
%Part 1 - Numerical Method
surf(surface0);
G_M2 = sparse((width_x * length_y), (width_x * length_y));
V_M2 = zeros(1, (width_x * length_y));
v_0 = 1;

for i = 1:width_x
            for j = 1:length_y
            n = j + (i - 1) * length_y;
            nxm = j + ((i-1) - 1) * length_y;
            nxp = j + ((i+1) - 1) * length_y;
            nym = (j-1) + (i - 1) * length_y;
            nyp = (j+1) + (i - 1) * length_y;
            if i == 1
            G_M2(n, :) = 0;
            G_M2(n, n) = 1;
            V_M2(1, n) = v_0;
            elseif i == width_x
            G_M2(n, :) = 0;
            G_M2(n, n) = 1;
```

```
            V_M2(1, n) = v_0;
            elseif j == 1
            G_M2(n, :) = 0;
            G_M2(n, n) = 1;
            elseif j == length_y
            G_M2(n, :) = 0;
            G_M2(n, n) = 1;
            else
            G_M2(n, :) = 0;
            G_M2(n, n) = -4;
            G_M2(n, nxm) = 1;
            G_M2(n, nxp) = 1;
            G_M2(n, nym) = 1;
            G_M2(n, nyp) = 1;
            end
            end
end

mAtrix2 = G_M2\V_M2';
figure (3);
surface2 = zeros(width_x, length_y);

for i = 1:width_x
            for j = 1:length_y
            n = j + (i - 1) * length_y;
            nxm = j + ((i-1) - 1) * length_y;
            nxp = j + ((i+1) - 1) * length_y;
            nym = (j-1) + (i - 1) * length_y;
            nyp = (j+1) + (i - 1) * length_y;
            surface2(i, j) = mAtrix2(n);
            end
end
surf(surface2);
title("Numerical Method");

% Part 1 - Analytical Method
ana_sol = zeros(60, 40);
a = 60;
b = 20;
x = linspace(-20,20,40);
y = linspace(0,60,60);
[x_mesh, y_mesh] = meshgrid(x, y);

for n = 1:2:300
```

```
        ana_sol =  (ana_sol + (4 * v0/pi).*(cosh((n * pi * x_mesh)/a) .* sin((n * pi * y_mesh)/a)) ./ (n *
cosh((n * pi * b)/a)));
        figure(4);
        surf(x, y, ana_sol);
        title("Analytical Method");
        pause(0.01);

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PART 2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc
clear
set(0,'DefaultFigureWindowStyle','docked')
width_x = 120;
length_y = 80;
x_val = 80;
y_val = 100;

% G and V matricies
G_M = sparse((width_x * length_y), (width_x * length_y));
V_M = zeros(1, (width_x * length_y));
v_0 = 1;

%Inner and Outer Conductivity
cond_in = 1e-2;
cond_out = 1;

%Conductivity Map
cond_map = ones(width_x, length_y);

%Bottlenecks
bot1 = [(width_x * 0.4), (width_x * 0.6),  length_y, (length_y * 0.75)];
bot2 = [(width_x * 0.4), (width_x * 0.6), 0, (length_y * 0.25)];
for i = 1:width_x
        for j = 1:length_y
        if(i > bot1(1) && i < bot1(2) && ((j < bot2(4)) || (j > bot1(4))))
        cond_map(i,j) = 1e-2;
        end
        end
end

% Plotting Conductivity Map
```

```
figure(5);
surf(cond_map);
colorbar
title('Conductivity Plot');
xlabel('X')
ylabel('Y')
zlabel('Conductive Field')

%Solving Matricies
for i = 1:width_x
	for j = 1:length_y
	n = j + (i - 1) * length_y;
	nxm = j + ((i-1) - 1) * length_y;
	nxp = j + ((i+1) - 1) * length_y;
	nym = (j-1) + (i - 1) * length_y;
	nyp = (j+1) + (i - 1) * length_y;
	%Creating index for each condition needing to be satisifed
	index1 = (i == 1);
	index2 = (i == width_x);
	index3 = (j == 1 && i > 1 && i < width_x);
	index4 = (i == bot1(1));
	index5 = (i == bot1(2));
	index6 = (i > bot1(1) && i < bot1(2));
	index7 = (j == length_y && i > 1 && i < width_x);
	index8 = (i == bot1(2));
	index9 = (i > bot1(1) && i < bot1(2));
	index10 = (i == bot1(1) && ((j < bot2(4)) || (j > bot1(4))));
	index11 = (i == bot1(2) && ((j < bot2(4)) || (j > bot1(4))));
	index12 = (i > bot1(1) && i < bot1(2) && ((j < bot2(4)) || (j > bot1(4))));
	if (index1)
	G_M(n, :) = 0;
	G_M(n, n) = 1;
	V_M(1, n) = 1;
	elseif (index2)
	G_M(n, :) = 0;
	G_M(n, n) = 1;
	elseif (index3)
	if (index4)
	G_M(n, n) = -3;
	G_M(n, nyp) = cond_in;
	G_M(n, nxp) = cond_in;
	G_M(n, nxm) = cond_out;
	elseif (index5)
	G_M(n, n) = -3;
```

```
G_M(n, nyp) = cond_in;
G_M(n, nxp) = cond_out;
G_M(n, nxm) = cond_in;
elseif (index6)
G_M(n, n) = -3;
G_M(n, nyp) = cond_in;
G_M(n, nxp) = cond_in;
G_M(n, nxm) = cond_in;
else
G_M(n, n) = -3;
G_M(n, nyp) = cond_out;
G_M(n, nxp) = cond_out;
G_M(n, nxm) = cond_out;
end
elseif (index7)
if (index4)
G_M(n, n) = -3;
G_M(n, nym) = cond_in;
G_M(n, nxp) = cond_in;
G_M(n, nxm) = cond_out;
elseif (index8)
G_M(n, n) = -3;
G_M(n, nym) = cond_in;
G_M(n, nxp) = cond_out;
G_M(n, nxm) = cond_in;
elseif (index9)
G_M(n, n) = -3;
G_M(n, nym) = cond_in;
G_M(n, nxp) = cond_in;
G_M(n, nxm) = cond_in;
else
G_M(n, n) = -3;
G_M(n, nym) = cond_out;
G_M(n, nxp) = cond_out;
G_M(n, nxm) = cond_out;
end
else
if (index10)
G_M(n, n) = -4;
G_M(n, nyp) = cond_in;
G_M(n, nym) = cond_in;
G_M(n, nxp) = cond_in;
G_M(n, nxm) = cond_out;
elseif (index11)
```

```
            G_M(n, n) = -4;
            G_M(n, nyp) = cond_in;
            G_M(n, nym) = cond_in;
            G_M(n, nxp) = cond_out;
            G_M(n, nxm) = cond_in;
            elseif (index12)
            G_M(n, n) = -4;
            G_M(n, nyp) = cond_in;
            G_M(n, nym) = cond_in;
            G_M(n, nxp) = cond_in;
            G_M(n, nxm) = cond_in;
            else
            G_M(n, n) = -4;
            G_M(n, nyp) = cond_out;
            G_M(n, nym) = cond_out;
            G_M(n, nxp) = cond_out;
            G_M(n, nxm) = cond_out;
            end
            end
            end
end

% Matrix solution
mAtrix1 = G_M\V_M';
surface = zeros(width_x, length_y);

for i = 1:width_x
            for j = 1:length_y
            n = j + (i - 1) * length_y;
            nxm = j + ((i-1) - 1) * length_y;
            nxp = j + ((i+1) - 1) * length_y;
            nym = (j-1) + (i - 1) * length_y;
            nyp = (j+1) + (i - 1) * length_y;
            surface(i, j) = mAtrix1(n);
            end
end

figure (6);
surf(surface);
colorbar
title('Voltage Plot');
xlabel('X')
ylabel('Y')
zlabel('Voltage')
```

```
[E_y1, E_x1] = gradient(surface);
J = cond_map.*gradient(surface);
J_X = cond_map.*(-E_y1);
J_Y = cond_map.*(-E_x1);

% Current Density Plot
figure (7)
surf(J)
colorbar
title('Current Density Plot');
xlabel('X')
ylabel('Y')
zlabel('Current (m^2)')

% Plot of electric Field in the Y
figure (8)
surf (E_y1)
colorbar
title('Electric Field in Y-Axis');
xlabel('X')
ylabel('Y')
zlabel('Electric field in Y')

% Plot of electric field in the X
figure (9)
surf(E_x1)
colorbar
title('Electric Field in X-Axis')
xlabel('X')
ylabel('Y')
zlabel('Electric field in X')

% E-field(x,y) Plot
E_field = sqrt(E_y1.^2 + E_x1.^2);
figure (10)
surf(E_field)
figure (11)
quiver (-E_y1, -E_x1, 'b');
title('Electric Field - Current Around Resistive Regions')

% Finding Current density vs mesh size
clc
set(0,'DefaultFigureWindowStyle','docked')
```

```
clear
num = 30;
width_x = 2;
length_y = 3;
current_density = [];

for num = 1:num
    width_x = 3*num;
    length_y = 2*num;
    V0 = 5;
    G_M = sparse(length_y*width_x,length_y*width_x);
    mAtrix1 = zeros(length_y*width_x,1);
    cond_out = 1;
    cond_in = 1e-2;
    conductivity = cond_out.*ones(length_y,width_x);
    for i = 1:width_x
    for j = 1:length_y
    if((i <= 0.8*width_x && i >= (0.3*width_x) && j <= (0.3*length_y)) || (i <= (0.8*width_x) && i
>= (0.3*width_x) && j >= (0.8*length_y)))
    conductivity(j,i) = cond_in;
    end
    end
    end
    for i = 1:width_x
    for j = 1:length_y
    n = j + (i - 1) * length_y;
    nxm = j + ((i-1) - 1) * length_y;
    nxp = j + ((i+1) - 1) * length_y;
    nym = (j-1) + (i - 1) * length_y;
    nyp = (j+1) + (i - 1) * length_y;
    if(i == 1)
    mAtrix1(n,1) = V0;
    G_M(n,n) = 1;
    elseif(i == width_x)
    mAtrix1(n,1) = 0;
    G_M(n,n) = 1;
    elseif(j == 1)
    G_M(n,n) = -(((conductivity(j,i) + conductivity(j,i-1))/2)+((conductivity(j,i) +
conductivity(j,i+1))/2)+((conductivity(j,i) + conductivity(j+1,i))/2));
    G_M(n,nxm) = (conductivity(j,i) + conductivity(j,i-1))/2;
    G_M(n,nxp) = (conductivity(j,i) + conductivity(j,i+1))/2;
    G_M(n,nyp) = (conductivity(j,i) + conductivity(j+1,i))/2;
    mAtrix1(n,1) = 0;
    elseif(j == length_y)
```

```
            G_M(n,n) = -(((conductivity(j,i) + conductivity(j,i-1))/2)+((conductivity(j,i) +
conductivity(j,i+1))/2)+((conductivity(j,i) + conductivity(j-1,i))/2));
            G_M(n,nxm) = (conductivity(j,i) + conductivity(j,i-1))/2;
            G_M(n,nxp) = (conductivity(j,i) + conductivity(j,i+1))/2;
            G_M(n,nym) = (conductivity(j,i) + conductivity(j-1,i))/2;
            mAtrix1(n,1) = 0;
            else
            G_M(n,n) = -(((conductivity(j,i) + conductivity(j,i-1))/2)+((conductivity(j,i) +
conductivity(j,i+1))/2)+((conductivity(j,i) + conductivity(j-1,i))/2)+((conductivity(j,i) +
conductivity(j+1,i))/2));
            G_M(n,nxm) = (conductivity(j,i) + conductivity(j,i-1))/2;
            G_M(n,nxp) = (conductivity(j,i) + conductivity(j,i+1))/2;
            G_M(n,nym) = (conductivity(j,i) + conductivity(j-1,i))/2;
            G_M(n,nyp) = (conductivity(j,i) + conductivity(j+1,i))/2;
            mAtrix1(n,1) = 0;
            end
            end
            end
            V_M = G_M\mAtrix1;
            for i = 1:width_x
            for j = 1:length_y
            n = (i-1)*length_y+j;
            surface(j,i) = V_M(n,1);
            end
            end
            [Efield_x,Efield_y] = gradient(surface);
            J_xdir = conductivity.*(-Efield_x);
            J_ydir = conductivity.*(-Efield_y);
            current_density(num) = mean(mean((((J_xdir.^2)+(J_ydir.^2)).^0.5)));
end

% Plotting current density vs mesh size
figure(12)
plot(1:num,current_density,'m')
title('Current vs Mesh Size')
clear
num = 50;
current_density = [];
for num = 1:num
            width_x = 90;
            length_y = 60;
            V0 = 5;
            G_M = sparse(length_y*width_x,length_y*width_x);
            mAtrix1 = zeros(length_y*width_x,1);
```

```
cond_out = 1;
cond_in = 0.01;
conductivity = cond_out.*ones(length_y,width_x);

for i = 1:width_x
for j = 1:length_y
if((i <= 0.8*width_x && i >= 0.3*width_x && j <= 0.01*num*length_y) || (i <= (1-num*0.01)*length_y && i >= 0.25*width_x && j >= (1-num*0.01)*length_y))
conductivity(j,i) = cond_in;
end
end
end

for i = 1:width_x
for j = 1:length_y
n = j + (i - 1) * length_y;
nxm = j + ((i-1) - 1) * length_y;
nxp = j + ((i+1) - 1) * length_y;
nym = (j-1) + (i - 1) * length_y;
nyp = (j+1) + (i - 1) * length_y;
if(i == 1)
mAtrix1(n,1) = V0;
G_M(n,n) = 1;
elseif(i == width_x)
mAtrix1(n,1) = 0;
G_M(n,n) = 1;
elseif(j == 1)
G_M(n,n) = -(((conductivity(j,i) + conductivity(j,i-1))/2)+((conductivity(j,i) + conductivity(j,i+1))/2)+((conductivity(j,i) + conductivity(j+1,i))/2));
G_M(n,nxm) = (conductivity(j,i) + conductivity(j,i-1))/2;
G_M(n,nxp) = (conductivity(j,i) + conductivity(j,i+1))/2;
G_M(n,nyp) = (conductivity(j,i) + conductivity(j+1,i))/2;

mAtrix1(n,1) = 0;
elseif(j == length_y)
G_M(n,n) = -(((conductivity(j,i) + conductivity(j,i-1))/2)+((conductivity(j,i) + conductivity(j,i+1))/2)+((conductivity(j,i) + conductivity(j-1,i))/2));
G_M(n,nxm) = (conductivity(j,i) + conductivity(j,i-1))/2;
G_M(n,nxp) = (conductivity(j,i) + conductivity(j,i+1))/2;
G_M(n,nym) = (conductivity(j,i) + conductivity(j-1,i))/2;

mAtrix1(n,1) = 0;
else
```

```
            G_M(n,n) = -(((conductivity(j,i) + conductivity(j,i-1))/2)+((conductivity(j,i) +
conductivity(j,i+1))/2)+((conductivity(j,i) + conductivity(j-1,i))/2)+((conductivity(j,i) +
conductivity(j+1,i))/2));
            G_M(n,nxm) = ((conductivity(j,i) + conductivity(j,i-1))/2);
            G_M(n,nxp) = (conductivity(j,i) + conductivity(j,i+1))/2;
            G_M(n,nym) = (conductivity(j,i) + conductivity(j-1,i))/2;
            G_M(n,nyp) = (conductivity(j,i) + conductivity(j+1,i))/2;
            mAtrix1(n,1) = 0;
            end
            end
            end

            % V matrix solution
            V_M = G_M\mAtrix1;

            for i = 1:width_x
            for j = 1:length_y
            n = j + (i - 1) * length_y;
            nxm = j + ((i-1) - 1) * length_y;
            nxp = j + ((i+1) - 1) * length_y;
            nym = (j-1) + (i - 1) * length_y;
            nyp = (j+1) + (i - 1) * length_y;
            surface(j,i) = V_M(n,1);
            end
            end

            [Efield_x,Efield_y] = gradient(surface);
            J_xdir = conductivity.*(-Efield_x);
            J_ydir = conductivity.*(-Efield_y);
            current_density(num) = mean(mean((((J_xdir.^2)+(J_ydir.^2)).^0.5)));
end
% Current density vs Bottleneck size
figure(13)
plot(current_density,(-1)*(1:num), 'm')
title('Current vs Bottleneck Size')
clear
num = 50;
current_density = [];

for num = 1:num

            width_x = 90;
            length_y = 60;
            V0 = 5;
```

```
G_M = sparse(length_y*width_x,length_y*width_x);
mAtrix1 = zeros(length_y*width_x,1);
cond_out = 1;
cond_in = 1.02-num*0.02;
conductivity = cond_out.*ones(length_y,width_x);

for i = 1:width_x
for j = 1:length_y
if((i <= 0.8*width_x && i >= 0.3*width_x && j <= 0.3*length_y) || (i <= 0.8*width_x && i >=
0.3*width_x && j >= 0.8*length_y))
conductivity(j,i) = cond_in;
end
end
end

for i = 1:width_x
for j = 1:length_y
n = j + (i - 1) * length_y;
nxm = j + ((i-1) - 1) * length_y;
nxp = j + ((i+1) - 1) * length_y;
nym = (j-1) + (i - 1) * length_y;
nyp = (j+1) + (i - 1) * length_y;
if(i == 1)
mAtrix1(n,1) = V0;
G_M(n,n) = 1;
elseif(i == width_x)
mAtrix1(n,1) = 0;
G_M(n,n) = 1;
elseif(j == 1)
G_M(n,n) = -(((conductivity(j,i) + conductivity(j,i-1))/2)+((conductivity(j,i) +
conductivity(j,i+1))/2)+((conductivity(j,i) + conductivity(j+1,i))/2));
G_M(n,nxm) = (conductivity(j,i) + conductivity(j,i-1))/2;
G_M(n,nxp) = (conductivity(j,i) + conductivity(j,i+1))/2;
G_M(n,nyp) = (conductivity(j,i) + conductivity(j+1,i))/2;

mAtrix1(n,1) = 0;
elseif(j == length_y)
G_M(n,n) = -(((conductivity(j,i) + conductivity(j,i-1))/2)+((conductivity(j,i) +
conductivity(j,i+1))/2)+((conductivity(j,i) + conductivity(j-1,i))/2));
G_M(n,nxm) = (conductivity(j,i) + conductivity(j,i-1))/2;
G_M(n,nxp) = (conductivity(j,i) + conductivity(j,i+1))/2;
G_M(n,nym) = (conductivity(j,i) + conductivity(j-1,i))/2;

mAtrix1(n,1) = 0;
```

```
        else
        G_M(n,n) = -(((conductivity(j,i) + conductivity(j,i-1))/2)+((conductivity(j,i) +
conductivity(j,i+1))/2)+((conductivity(j,i) + conductivity(j-1,i))/2)+((conductivity(j,i) +
conductivity(j+1,i))/2));
        G_M(n,nxm) = (conductivity(j,i) + conductivity(j,i-1))/2;
        G_M(n,nxp) = (conductivity(j,i) + conductivity(j,i+1))/2;
        G_M(n,nym) = (conductivity(j,i) + conductivity(j-1,i))/2;
        G_M(n,nyp) = (conductivity(j,i) + conductivity(j+1,i))/2;
        mAtrix1(n,1) = 0;
        end
        end
        end

        V_M = G_M\mAtrix1;

        for i = 1:width_x
        for j = 1:length_y
        n = j + (i - 1) * length_y;
        nxm = j + ((i-1) - 1) * length_y;
        nxp = j + ((i+1) - 1) * length_y;
        nym = (j-1) + (i - 1) * length_y;
        nyp = (j+1) + (i - 1) * length_y;
        surface(j,i) = V_M(n,1);
        end
        end

        [Efield_x,Efield_y] = gradient(surface);
        J_xdir = conductivity.*(-Efield_x);
        J_ydir = conductivity.*(-Efield_y);
        current_density(num) = mean(mean((((J_xdir.^2)+(J_ydir.^2)).^0.5)));
end
% Plotting the current density vs the conductivity
figure(14)
plot(1:num,(-1)*current_density,'m')
title('Current vs Conductivity')
```