ELEC 4700 - Assignment 2

Table of Contents

Completed By: Joel Demetre (100943543)

Due: Feburary 25th 2018

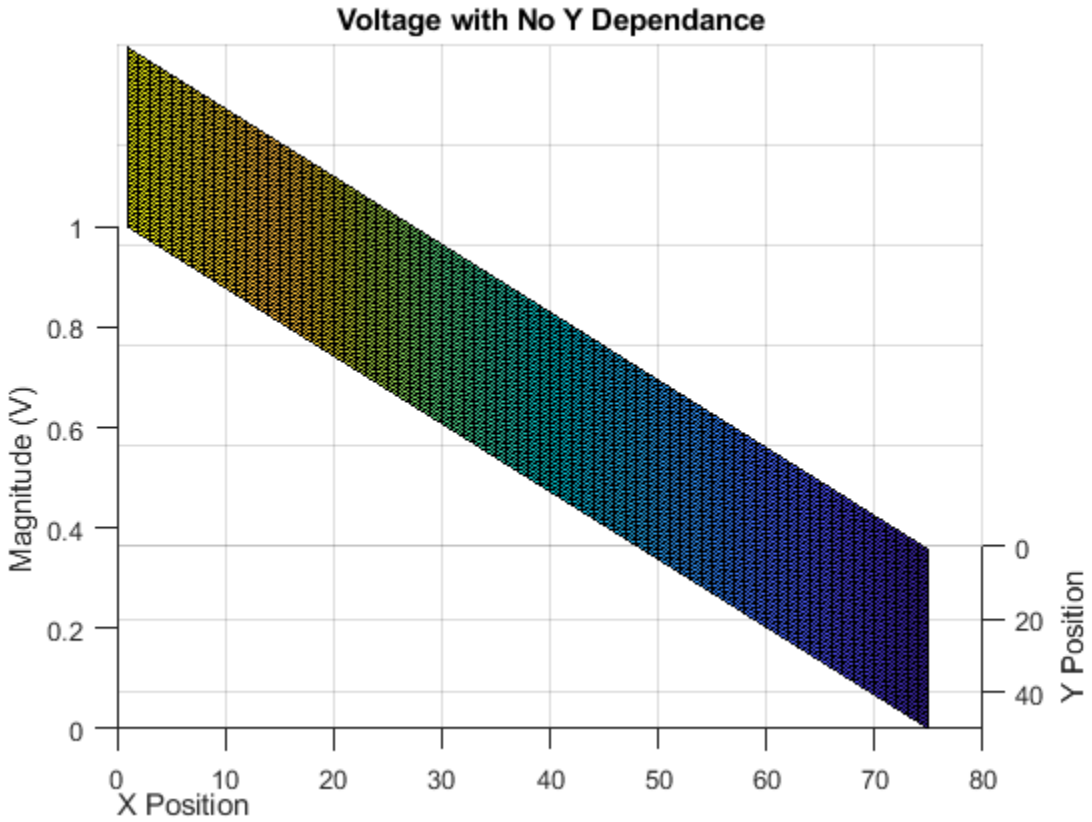
Question 1

Part A

This 2 Dimensional structure is treated as a 1 Dimensional dependancy in the x Direction.

```
clear all;
close all;

v0 = 1;
nx = 75;
ny = 50;

Z = zeros(nx*ny, nx*ny);
B = zeros(1,nx*ny);
%Left BC
B(1,1:ny) = 1;


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

        if i == 1 || i == nx
           Z(n,:) = 0;
           Z(n,n) = 1;
        else
            Z(n,n) = -2;
            Z(n,j + (i-2)*ny) = 1;
            Z(n,j + (i)*ny) = 1;
        end
    end
end
```

```
k1 = Z\B';
V1 = zeros(nx,ny);
for i = 1:nx
   for j = 1:ny
    n = j + (i-1)*ny;
    V1(i,j) = k1(n);
   end
end

figure(1);
surf(V1);
hold on;
title('Voltage with No Y Dependance');
xlabel('Y Position');
ylabel('X Position');
view([90 20]);
zlabel('Magnitude (V)');
grid on;
hold off;
```

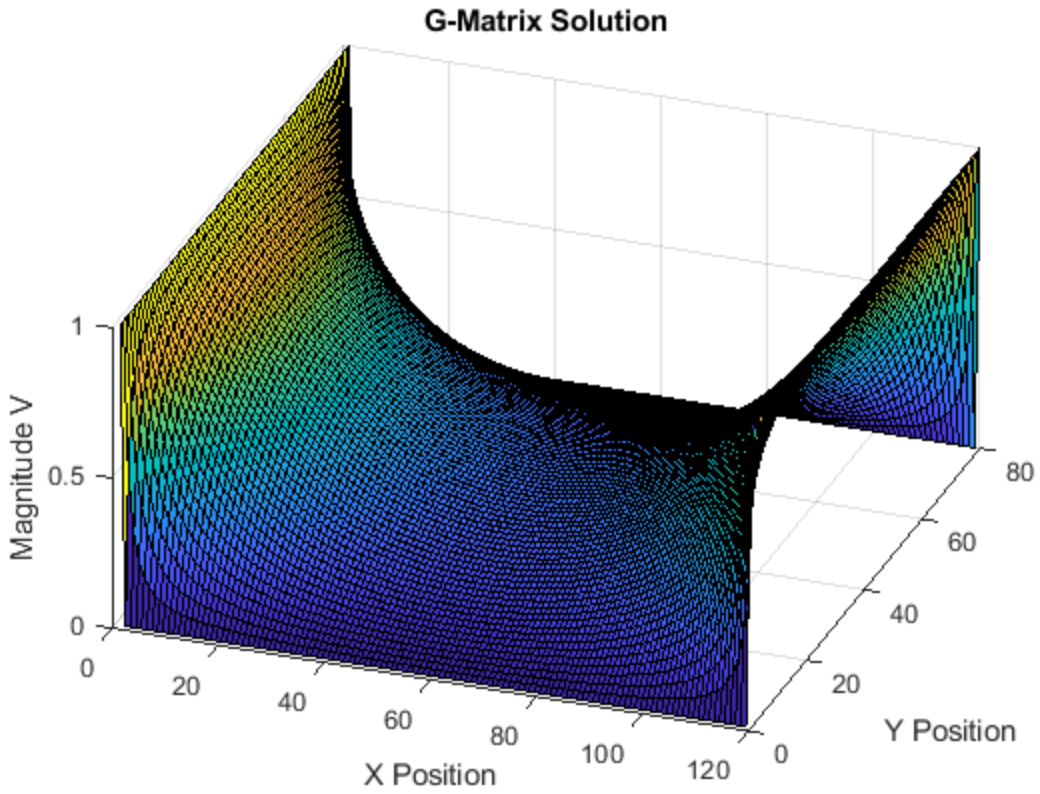


Part B

The case where the top and bottom are set 0 volts and the left and right sides are set to V_o in this case V_o is 1 Volt.

```
nx = 80;
```

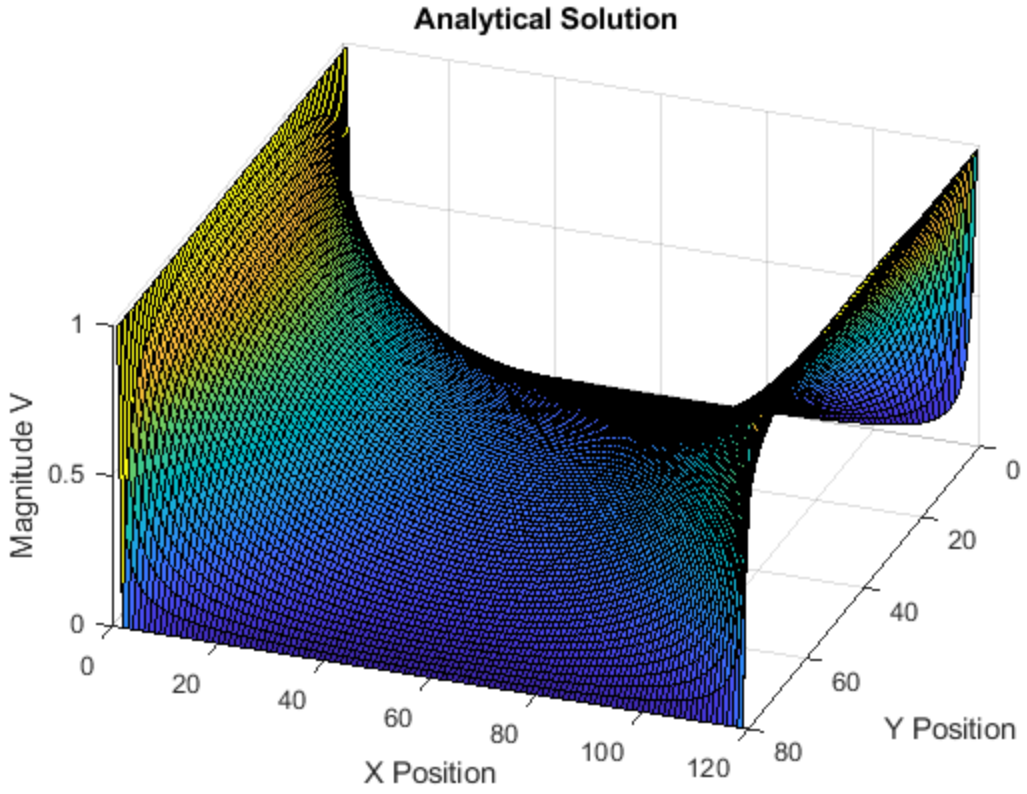
```
ny = 120;

B = sparse(1,nx*ny);
G = sparse(nx*ny, nx*ny);
%Left BC
B(1, (1:ny:nx*ny)) = 1;
%Right BC
B(1, (ny:ny:nx*ny)) = 1;

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

        if i == 1 || i == nx
           G(n,:) = 0;
           G(n,n) = 1;
        elseif j == ny || j == 1
           G(n,:) = 0;
           G(n,n) = 1;
        else
            G(n,n) = -4;
            G(n,j + (i-2)*ny) = 1;
            G(n,j-1 + (i-1)*ny) = 1;
            G(n,j+1 + (i-1)*ny) = 1;
            G(n,j + (i)*ny) = 1;
        end
    end
end

k = G\B';
V = zeros(nx,ny);
for i = 1:nx
   for j = 1:ny
    n = j + (i-1)*ny;
    V(i,j) = k(n);
   end
end

figure(2);
hold on;
xlabel('X Position');
ylabel('Y Position');
zlabel('Magnitude V');
title('G-Matrix Solution');
grid on;
view([20 45]);
surf(V);
hold off;
```



The analytical solution to the problem is shown below:

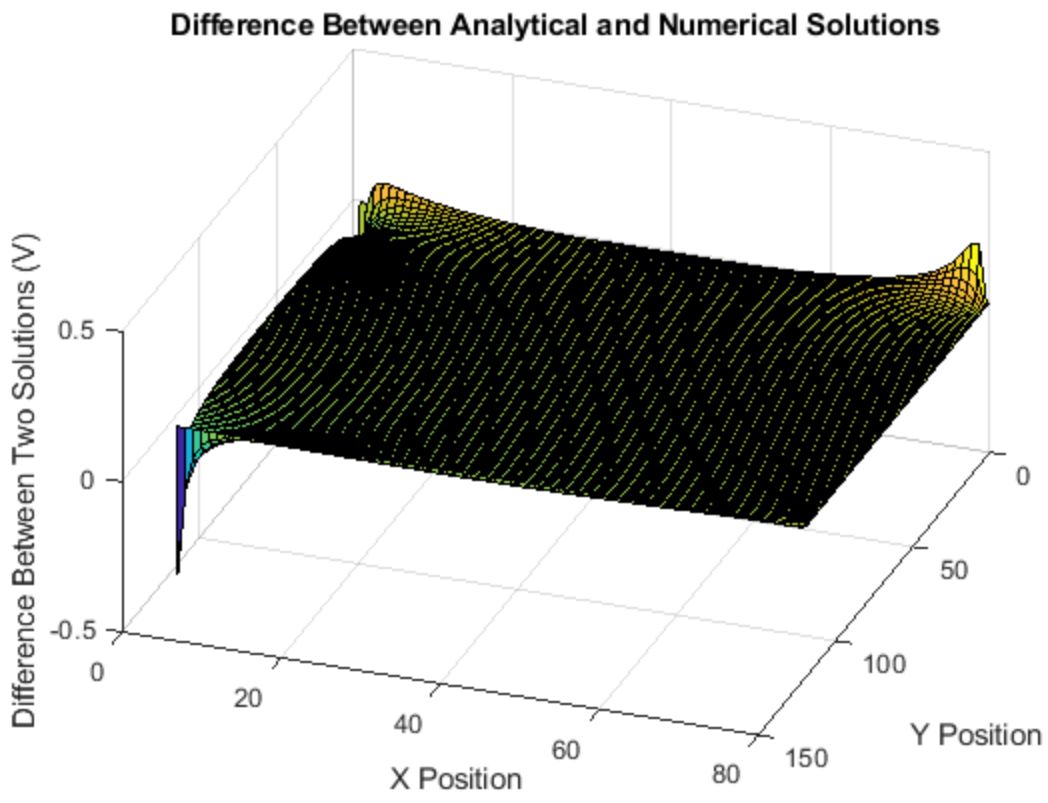
```
a = nx;
b = ny;
Vana = zeros(ny,nx);
for kp = 1:2:151
  for i = 1:ny
    for j = 1:nx
        if i == 1 || i == ny
           Vana(i,j) = 1;
        elseif  ~isnan(4/pi*cosh(kp*pi*i/a).*sin(kp*pi*j/a)/
cosh(kp*pi*b/a)/kp)
          Temp2 = 4*cosh(kp*pi*abs(i-ny)/a)*sin(kp*pi*abs(j-nx)/a)/
cosh(kp*pi*b/a)/pi/kp;
          Temp1 = 4*cosh(kp*pi*i/a)*sin(kp*pi*j/a)/cosh(kp*pi*b/a)/pi/
kp;
          Vana(i,j) = Vana(i,j) + Temp1 + Temp2;
       end
     end
  end
end

figure(3);
hold on;
xlabel('Y Position');
ylabel('X Position');
```

```
zlabel('Magnitude V');
title('Analytical Solution');
surf(Vana);
grid on;
view([110 45]);
hold off;
```



The larger the number of mesh points the more accurate the results for the interpolation between the points, in this case there is no scaling for length and width and rather everything is a ratio of length vs width. The numerical and analytical solutions match quite well, the numerical technique is slightly different from the analytical technqiue however these differences are quite small, for example the average difference is of 0.0107. The most major differences occur at the corners of the plot. The 4 corners seem to have the largest amount of difference between analytical and numerical solutions. The figure below shows the graph of the difference between the analytical and numerical solutions:

```
figure(4);
hold on;
xlabel('Y Position');
ylabel('X Position');
zlabel('Difference Between Two Solutions (V)');
title('Difference Between Analytical and Numerical Solutions');
surf(V-Vana');
grid on;
view([110 45]);
hold off;
```



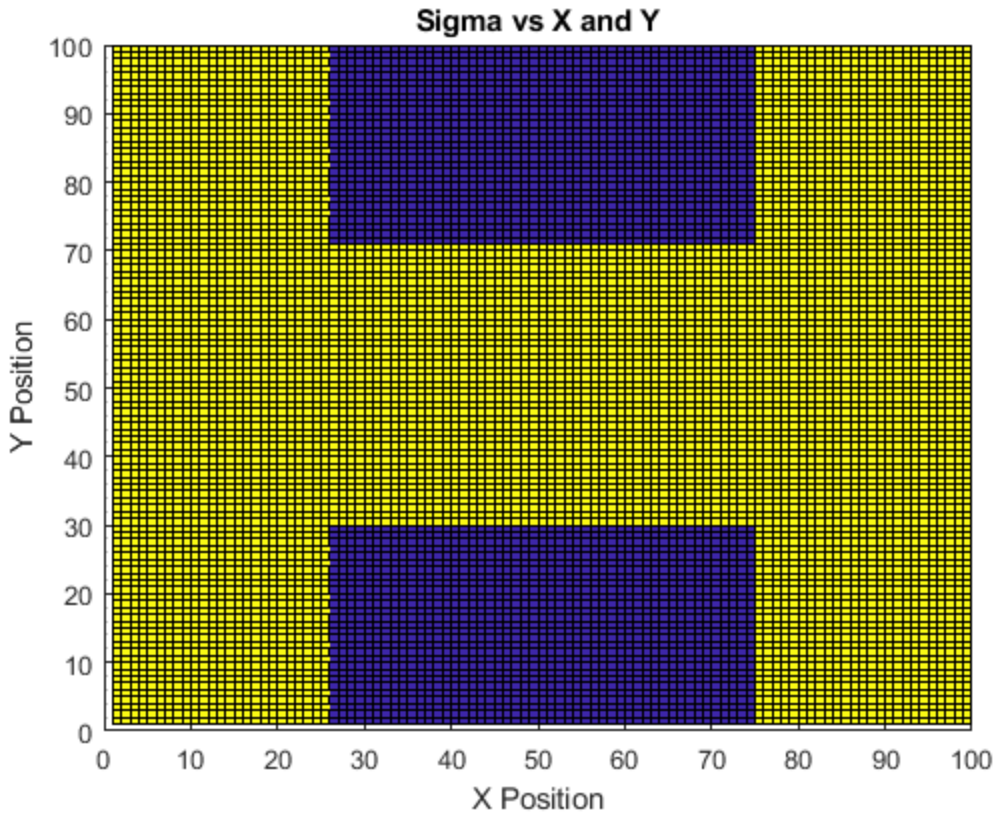
Question 2

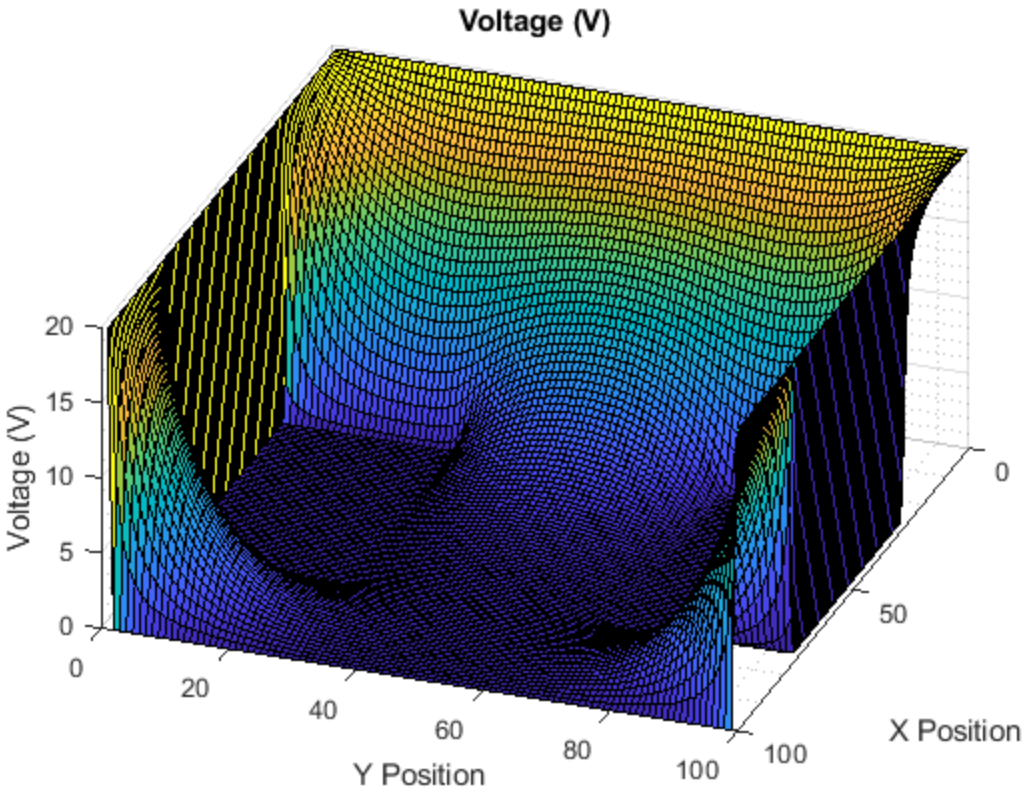
Part A

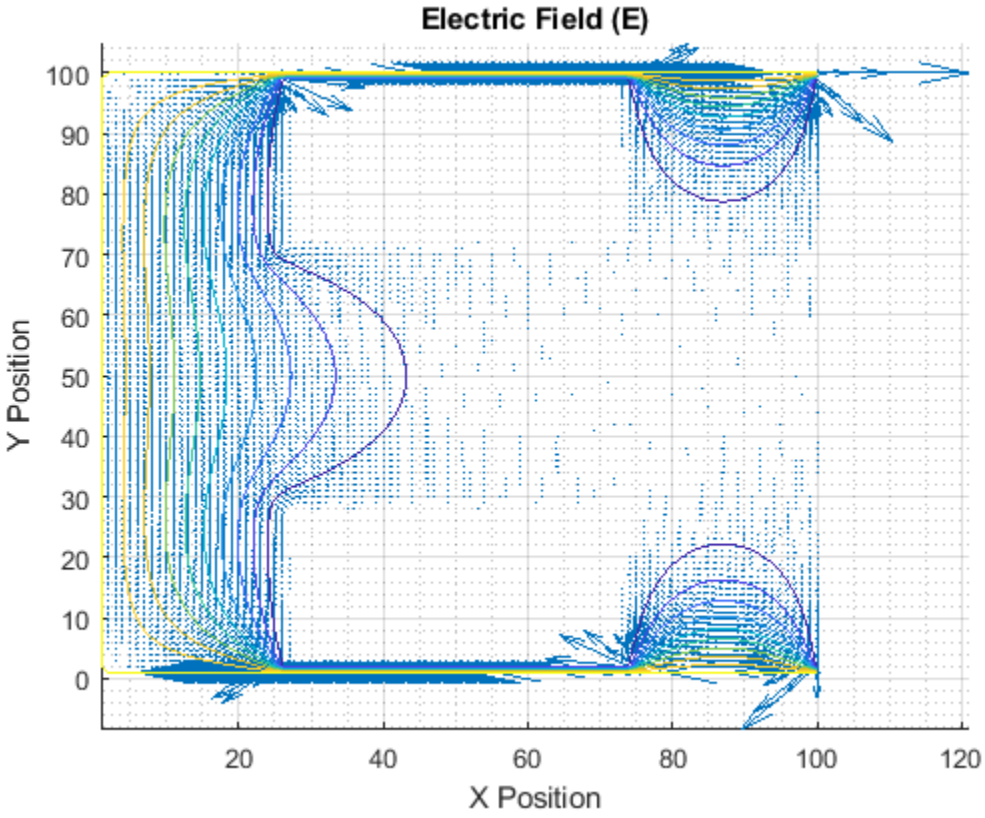
```
nx = 100;
ny = 100;
LC = 1;
RC = 0;
numIterations = 10000;
L = 100;
W = 100;
wb = 30;
lb = 50;



sigma = zeros(nx, ny);
sigma(:,:) = 1;
for i = 1:nx
    if i > (nx/L)*(L-lb)/2 && i < (nx/L)*(L+lb)/2
        for j = 1:ny
            if j < (ny/W)*wb || j> ceil((ny/W)*(W-wb))
               sigma(i,j) = 0.01;
            end
        end
```

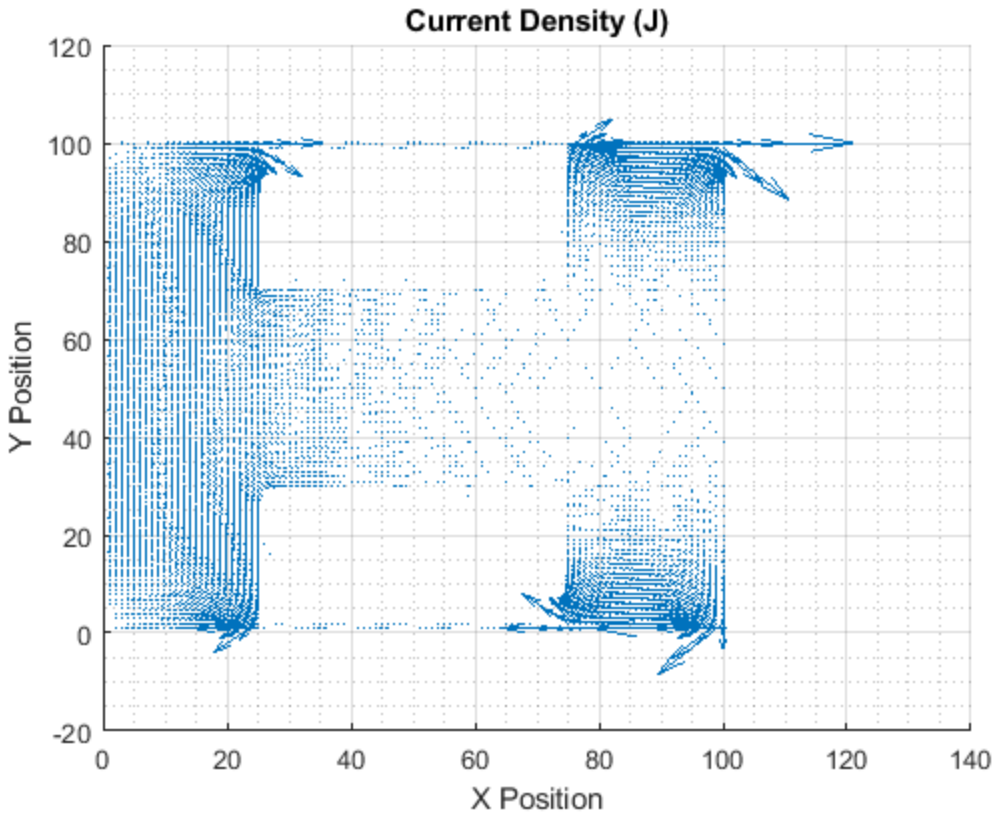
```
    end
end

deltax  = L/nx;
deltay  = W/ny;

Vold = zeros(nx, ny);
Vnew = zeros(nx, ny);
Vold(1,:) = 20;
Vold(nx,:) = 0;
Vold(:,1) = 20;
Vold(:,ny) = 20;

for kt= 1:numIterations
Vold = Vnew;
Vnew(1,:) = 20;
Vnew(nx,:) = 0;
Vnew(:,1) = 20;
Vnew(:,ny) = 20;

for i=2:nx-1
   for j=2:ny-1
      Vnew(i,j) = ((Vold(i-1, j)*sigma(i-1,j) + Vold(i+1, j)*sigma(i
+1,j))/deltax^2 + (Vold(i, j-1)*sigma(i,j-1) + Vold(i, j+1)*sigma(i,j
+1))/deltay^2)*deltax^2*deltay^2/(deltax^2 + deltay^2)/2;
   end
end
end

[Ex, Ey] = gradient(Vnew);
[X,Y] = meshgrid(L/nx:L/nx:L,W/ny:W/ny:W);
figure(5);
hold on;
surf(sigma');
title('Sigma vs X and Y');
xlabel('X Position');
ylabel('Y Position');
zlabel('Sigma Value');
grid on;
grid(gca,'minor');
hold off;

figure(6);
hold on;
grid on;
title('Voltage (V)');
xlabel('X Position');
ylabel('Y Position');
zlabel('Voltage (V)');
grid(gca,'minor');
view([110 45]);
surf(X',Y',Vnew);
hold off;
```

```
figure(7);
hold on;
grid on;
title('Electric Field (E)');
xlabel('X Position');
ylabel('Y Position');
zlabel('Electric Field (E)');
grid(gca,'minor');
quiver(X',Y',Ex, Ey,'LineWidth', 1,'AutoScaleFactor',15);
contour(X',Y',Vnew);
hold off;
eFlowx = sigma .* Ex;
eFlowy = sigma .* Ey;


figure(8);
hold on;
grid on;
title('Current Density (J)');
xlabel('X Position');
ylabel('Y Position');
zlabel('Current Density');
grid(gca,'minor');
quiver(X,Y,eFlowx', eFlowy','AutoScaleFactor',15);
hold off;
```




Voltage (V)
Voltage (V)
20
15
10
5
0
0
20
40
60
80
100
Y Position
0
50
100
X Position

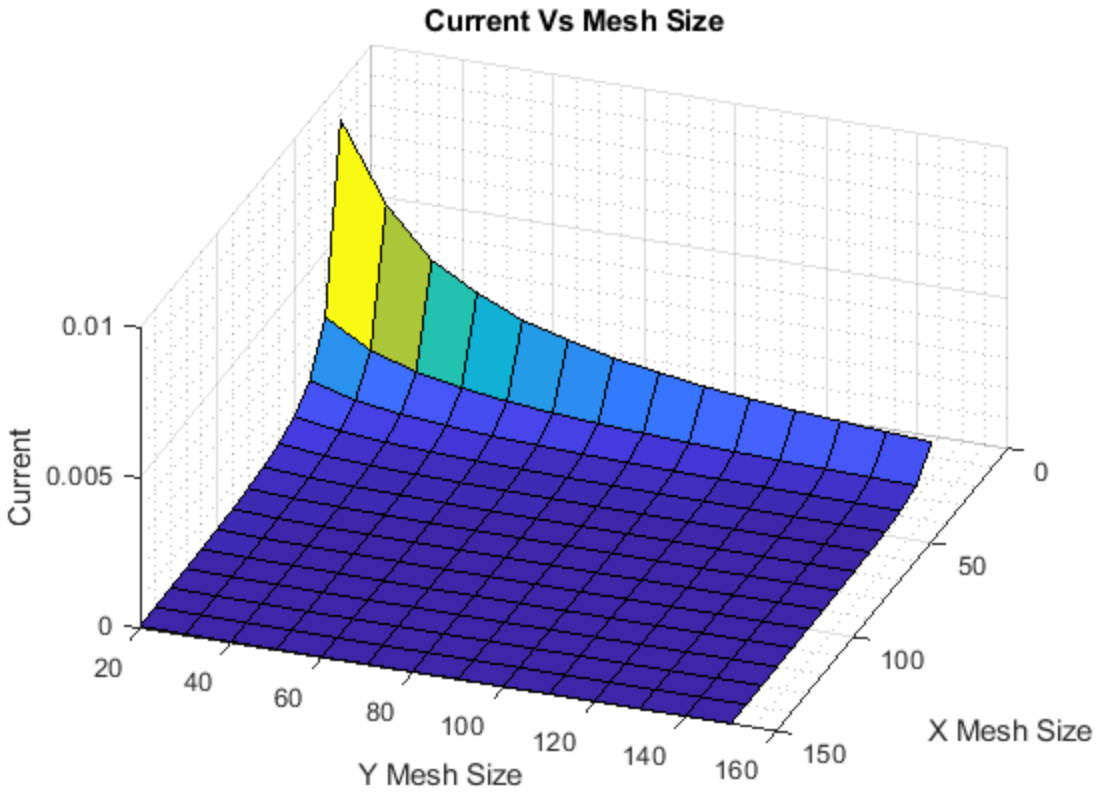

Electric Field (E)
Y Position
100
90
80
70
60
50
40
30
20
10
0
20
40
60
80
100
120
X Position



Part B

```
clear nx ny Vnew C0 Cnx Curr eFlowx eFlowy Ex Ey L;
Curr = zeros(14,14);
for klx = 20:10:150
for kly = 20:10:150
nx = klx;
ny = kly;
LC = 1;
RC = 0;
numIterations = 5000;
L = 100;
W = 100;
wb = 30;
lb = 50;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


sigma = zeros(nx, ny);
sigma(:,:) = 1;
for i = 1:nx
    if i > (nx/L)*(L-lb)/2 && i < (nx/L)*(L+lb)/2
        for j = 1:ny
            if j < (ny/W)*wb || j> ceil((ny/W)*(W-wb))
               sigma(i,j) = 0.01;
```

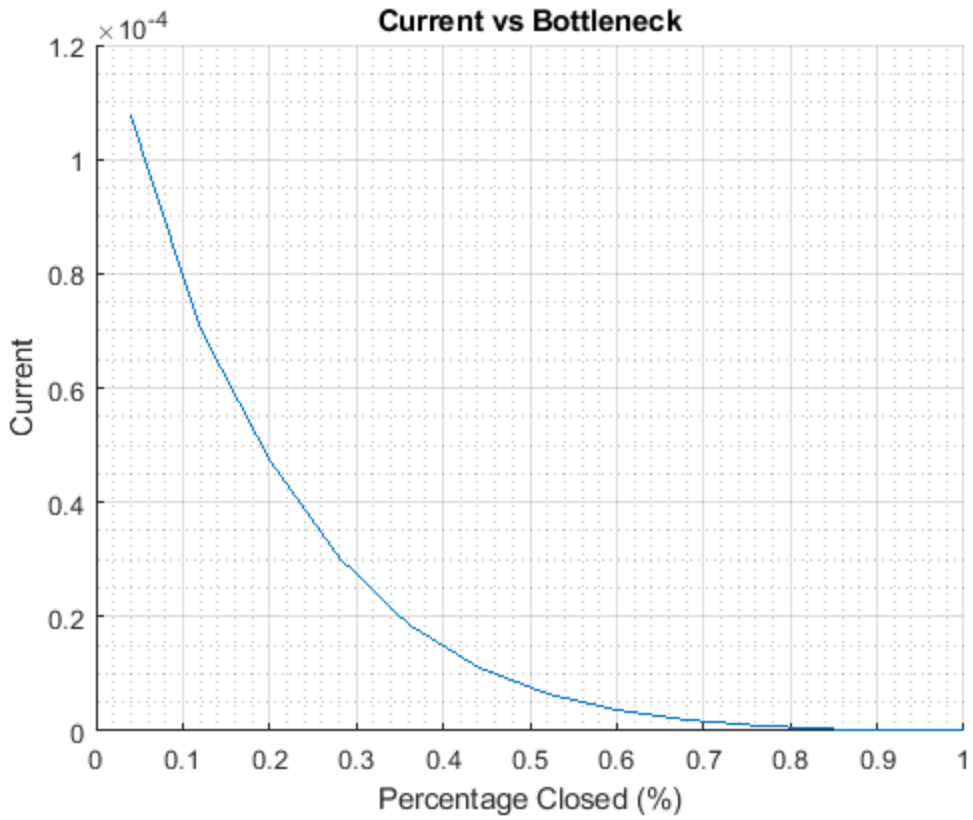
```
            end
        end
    end
end

deltax  = L/nx;
deltay  = W/ny;

Vold = zeros(nx, ny);
Vnew = zeros(nx, ny);
Vold(1,:) = 20;
Vold(nx,:) = 0;
Vold(:,1) = 20;
Vold(:,ny) = 20;

for kt= 1:numIterations
Vold = Vnew;
Vnew(1,:) = 20;
Vnew(nx,:) = 0;
Vnew(:,1) = 20;
Vnew(:,ny) = 20;

for i=2:nx-1
   for j=2:ny-1
      Vnew(i,j) = ((Vold(i-1, j)*sigma(i-1,j) + Vold(i+1, j)*sigma(i
+1,j))/deltax^2 + (Vold(i, j-1)*sigma(i,j-1) + Vold(i, j+1)*sigma(i,j
+1))/deltay^2)*deltax^2*deltay^2/(deltax^2 + deltay^2)/2;
   end
end
end
[Ex, Ey] = gradient(Vnew);
eFlowx = sigma .* Ex;

C0 = sum(eFlowx(2,:));
Cnx = sum(eFlowx(nx-1,:));
Curr(1+ (klx-20)/10,1 + (kly-20)/10) = (C0 + Cnx) * 0.5;
end
end

figure(9);
hold on;
grid on;
title('Current Vs Mesh Size');
xlabel('X Mesh Size');
ylabel('Y Mesh Size');
zlabel('Current');
view([110 45]);
grid(gca,'minor');
surf(20:10:150,20:10:150,Curr);
hold off;
```



Part C

```
clear nx ny Vnew C0 Cnx Curr eFlowx eFlowy Ex Ey L;

for klp = 2:4:50
nx = 200;
ny = 200;
LC = 1;
RC = 0;
numIterations = 10000;
L = 100;
W = 100;
wb = klp;
lb = 50;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


sigma = zeros(nx, ny);
sigma(:,:) = 1;
for i = 1:nx
    if i > (nx/L)*(L-lb)/2 && i < (nx/L)*(L+lb)/2
        for j = 1:ny
            if j < (ny/W)*wb || j> ceil((ny/W)*(W-wb))
               sigma(i,j) = 0.01;
            end
```

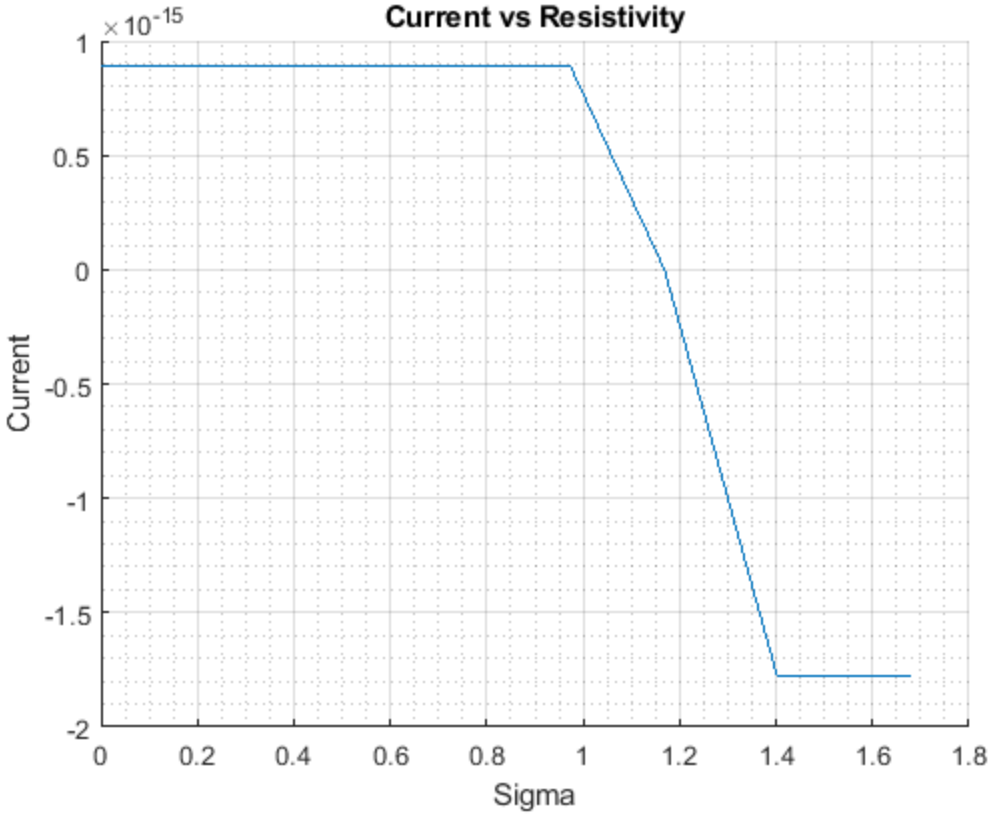
```
        end
    end
end

deltax  = L/nx;
deltay  = W/ny;

Vold = zeros(nx, ny);
Vnew = zeros(nx, ny);
Vold(1,:) = 20;
Vold(nx,:) = 0;
Vold(:,1) = 20;
Vold(:,ny) = 20;

for kt= 1:numIterations
Vold = Vnew;
Vnew(1,:) = 20;
Vnew(nx,:) = 0;
Vnew(:,1) = 20;
Vnew(:,ny) = 20;

for i=2:nx-1
   for j=2:ny-1
      Vnew(i,j) = ((Vold(i-1, j)*sigma(i-1,j) + Vold(i+1, j)*sigma(i
+1,j))/deltax^2 + (Vold(i, j-1)*sigma(i,j-1) + Vold(i, j+1)*sigma(i,j
+1))/deltay^2)*deltax^2*deltay^2/(deltax^2 + deltay^2)/2;
   end
end
end

[Ex, Ey] = gradient(Vnew);
eFlowx = sigma .* Ex;
eFlowy = sigma .* Ey;

C0 = sum(eFlowx(2,:));
Cnx = sum(eFlowx(nx-1,:));
Curr(klp) = (C0 + Cnx) * 0.5;
end

figure(10);
hold on;
title('Current vs Bottleneck');
xlabel('Percentage Closed (%)');
ylabel('Current');
plot((2:4:50)./(L/2), Curr(2:4:50));
grid on;
grid(gca,'minor');
hold off;
```



Part D

```
clear nx ny Vnew C0 Cnx Curr eFlowx eFlowy Ex Ey L;
counter = 1;
klp = 0.00001;
while(klp < 4)
nx = 100;
ny = 100;
LC = 1;
RC = 0;
numIterations = 5000;
L = 100;
W = 100;
wb = klp;
lb = 50;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


sigma = zeros(nx, ny);
sigma(:,:) = 1;
for i = 1:nx
    if i > (nx/L)*(L-lb)/2 && i < (nx/L)*(L+lb)/2
        for j = 1:ny
            if j < (ny/W)*wb || j> ceil((ny/W)*(W-wb))
               sigma(i,j) = klp;
```

```
            end
        end
    end
end

deltax  = L/nx;
deltay  = W/ny;

Vold = zeros(nx, ny);
Vnew = zeros(nx, ny);
Vold(1,:) = 20;
Vold(nx,:) = 0;
Vold(:,1) = 20;
Vold(:,ny) = 20;

for kt= 1:numIterations
Vold = Vnew;
Vnew(1,:) = 20;
Vnew(nx,:) = 0;
Vnew(:,1) = 20;
Vnew(:,ny) = 20;

for i=2:nx-1
   for j=2:ny-1
      Vnew(i,j) = ((Vold(i-1, j)*sigma(i-1,j) + Vold(i+1, j)*sigma(i
+1,j))/deltax^2 + (Vold(i, j-1)*sigma(i,j-1) + Vold(i, j+1)*sigma(i,j
+1))/deltay^2)*deltax^2*deltay^2/(deltax^2 + deltay^2)/2;
   end
end
end

[Ex, Ey] = gradient(Vnew);
eFlowx = sigma .* Ex;
eFlowy = sigma .* Ey;

C0 = sum(eFlowx(2,:));
Cnx = sum(eFlowx(nx-1,:));
Curr(2, counter) = (C0 + Cnx) * 0.5;
Curr(1, counter) = klp;
counter = counter + 1;
klp = klp*1.2;
end

figure(11);
hold on;
title('Current vs Resistivity');
xlabel('Sigma');
ylabel('Current');
grid on;
grid(gca,'minor');
plot(Curr(1,:), Curr(2,:));
hold off;
```



Published with MATLAB® R2017b