ELEC 4700 Assignment 2

Table of Contents

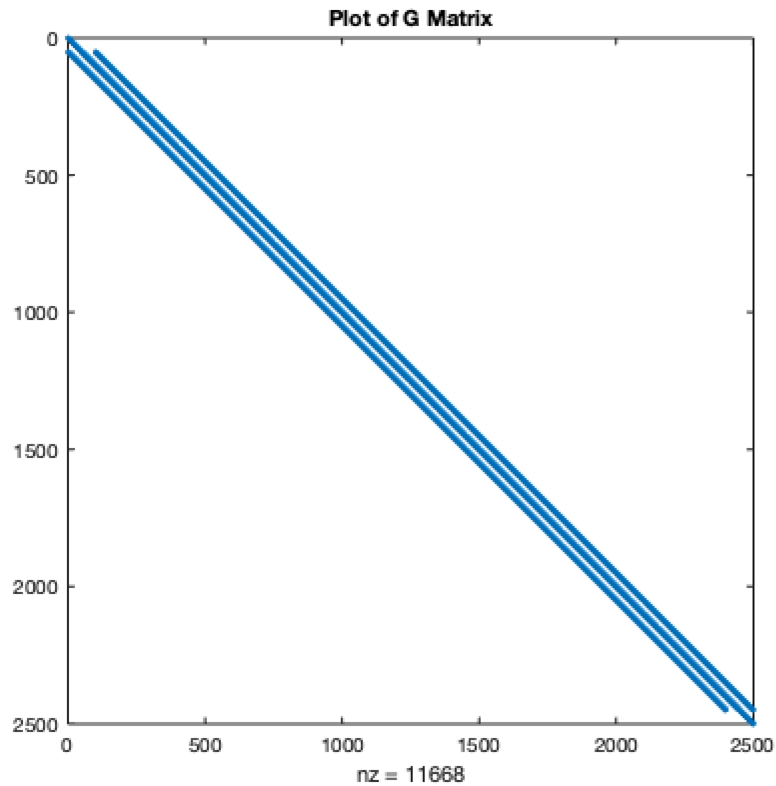
1 - Electrostatic Potential

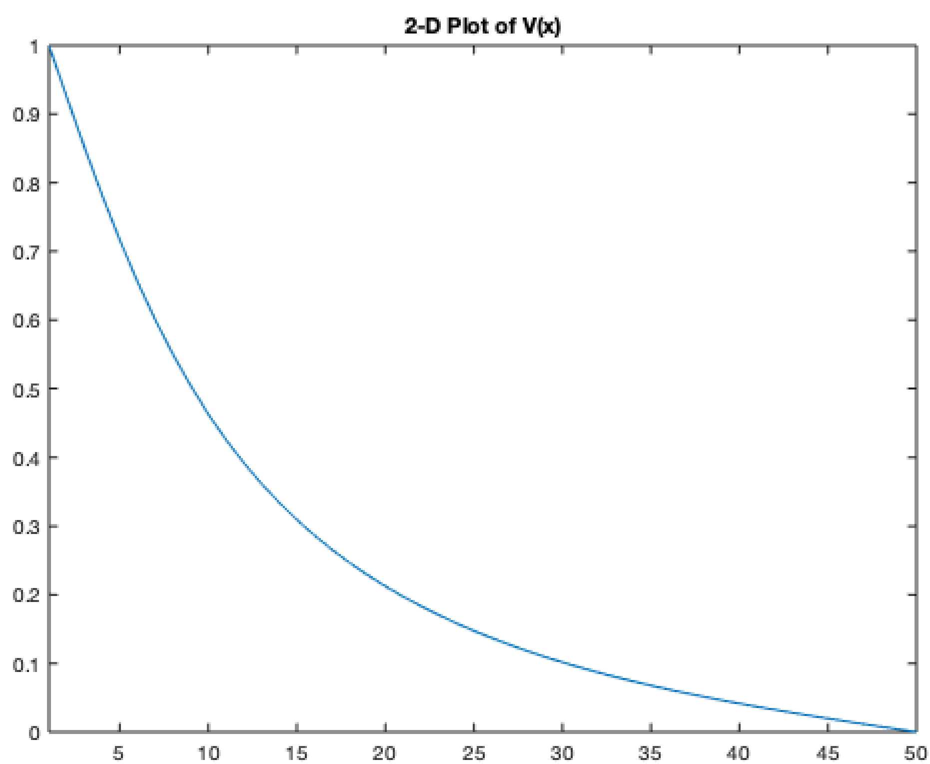
1.a - 1-D Boundary Condition

The electrostatic potential in a rectangular material can be solved using the matrix form of the finite difference method for a one dimentional boundary condition which sets one edge at some voltage, V0, and the opposite edge at zero volts.

```
% Setup
clear
clc

% Define Variables

V0 = 1;

% Define Matrix

nx = 50;
ny = nx;
G = sparse(nx*ny,nx*ny);
B = zeros(nx*ny,1);

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

        if i == 1
            G(n,:) = 0;
            G(n,n) = 1;
            B(n) = V0;
        elseif i == nx
            G(n,:) = 0;
            G(n,n) = 1;
            B(n) = 0;
        elseif j == 1
            G(:,n) = 0;
```

```
            G(n,n) = 1;
            B(n) = 0;
        elseif j == ny
            G(:,n) = 0;
            G(n,n) = 1;
            B(n) = 0;
        else
            nxm = j + (i-2)*ny;
            nxp = j + (i)*ny;
            nym = j-1 + (i-1)*ny;
            nyp = j+1 + (i-1)*ny;

            G(n,n) = -4;
            G(n,nxm) = 1;
            G(n,nxp) = 1;
            G(n,nym) = 1;
            G(n,nyp) = 1;
        end
    end
end

figure(1)
spy(G)
title('Plot of G Matrix')

solve = G\B;
data_z = zeros(nx,ny);
data_x = linspace(1,nx,nx);
data_y = linspace(1,ny,ny);

for i = 1:nx
    for j = 1:ny
        position = j + (i-1)*ny;
        data_z(i,j) = solve(position,1);
    end
end
figure(2)
plot(data_x,data_z(:,10))
xlim([1 50])
title('2-D Plot of V(x)')
```


Plot of G Matrix
0
500
1000
1500
2000
2500
0
500
1000
1500
2000
2500
nz = 11668


2-D Plot of V(x)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
5
10
15
20
25
30
35
40
45
50

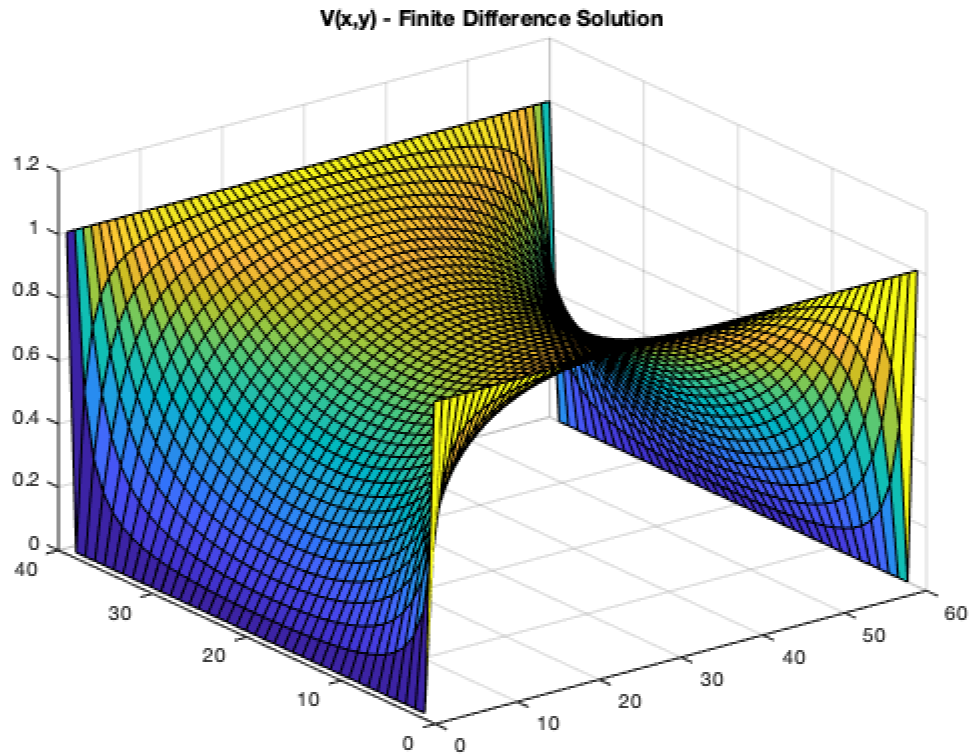
1.b - 2-D Boundary Condition

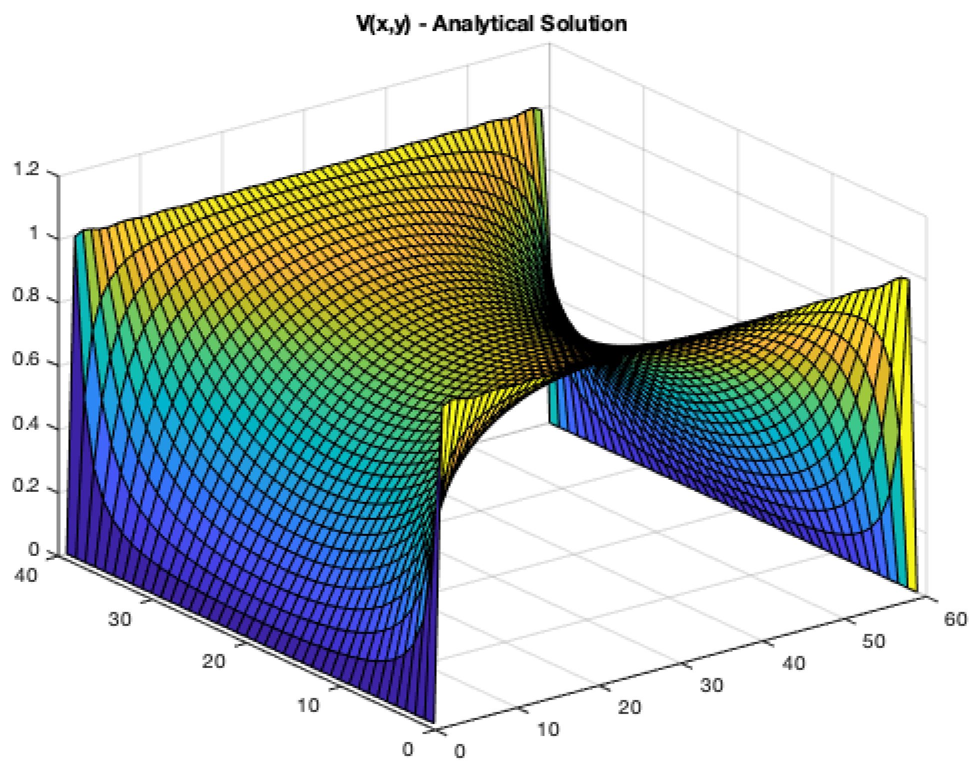
The same problem can now be solved again using the same method, except this time for a 2-D boundary condition which places two opposite sides at voltage V0, and the remaining two sides at zero volts.

The finite difference solution is then compaired to the analytical solution to ensure they provide the same results. The analytical analysis was completed for a series the appropriate length to provided the most accurate results.

```
% Define Matrix
nx = 40;
ny = 60;
G = sparse(nx*ny,nx*ny);
B = zeros(nx*ny,1);

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

        if i == 1
            G(n,:) = 0;
            G(n,n) = 1;
            B(n) = V0;
        elseif i == nx
            G(n,:) = 0;
            G(n,n) = 1;
            B(n) = V0;
        elseif j == 1
            G(:,n) = 0;
            G(n,n) = 1;
            B(n) = 0;
        elseif j == ny
            G(:,n) = 0;
            G(n,n) = 1;
            B(n) = 0;
        else
            nxm = j + (i-2)*ny;
            nxp = j + (i)*ny;
            nym = j-1 + (i-1)*ny;
            nyp = j+1 + (i-1)*ny;

            G(n,n) = -4;
            G(n,nxm) = 1;
            G(n,nxp) = 1;
            G(n,nym) = 1;
            G(n,nyp) = 1;
        end
    end
end

solve = G\B;
data_z = zeros(nx,ny);
data_x = linspace(1,nx,nx);
```

```
data_y = linspace(1,ny,ny);

for i = 1:nx
    for j = 1:ny
        position = j + (i-1)*ny;
        data_z(i,j) = solve(position,1);
    end
end
figure(3)
surf(data_y,data_x,data_z)
zlim([0 1.2])
title('V(x,y) - Finite Difference Solution')


% Analytical Solution - 60x40 Mesh
L = 60;
W = 40;
x = linspace(-W/2,W/2,nx);
y = linspace(0,L,ny);
a = L;
b = W/2;
V = zeros(L,W);
[X,Y] = meshgrid(x,y);


for n = 1:2:565
    V = V + ((1/n)*(cosh(n*pi*X./a)./cosh(n*pi*b./a)).*sin(n*pi*Y./
a));
end

    V = (4*V0/pi).*V';

figure(4)
surf(V)
zlim([0 1.2])
title('V(x,y) - Analytical Solution')
```


V(x,y) - Finite Difference Solution
1.2
1
0.8
0.6
0.4
0.2
0
40
30
20
10
0
0
10
20
30
40
50
60


V(x,y) - Analytical Solution
1.2
1
0.8
0.6
0.4
0.2
0
40
30
20
10
0
0
10
20
30
40
50
60

2 - Current Flow

2.a - Conductivity, Potential, Electric Field, and Current Density

The matrix form of the finite differance method used earlier can now be utilized to solve for the current flow through bottle-neck. Firstly a matrix of conductivity is created which defines the location and resistance of the bottle neck. The potential can then be fond using the finite differance method, and the three equations below can then be implemented to solve for the electric field and the current density.

$$E_x = -dV/dx$$

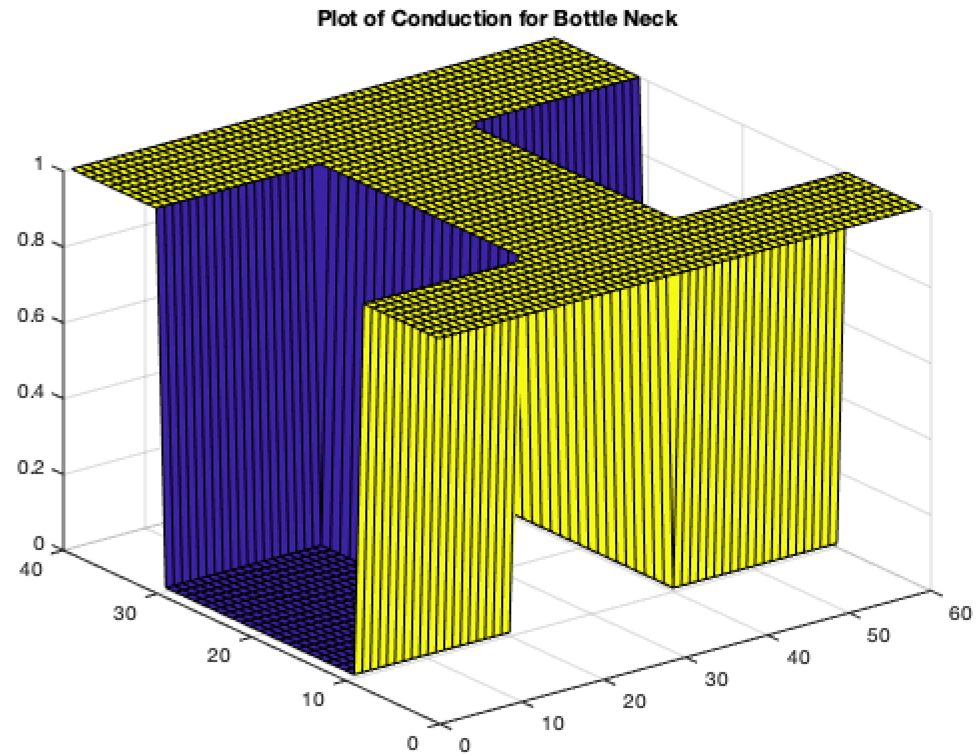
$$E_y = -dV/dy$$

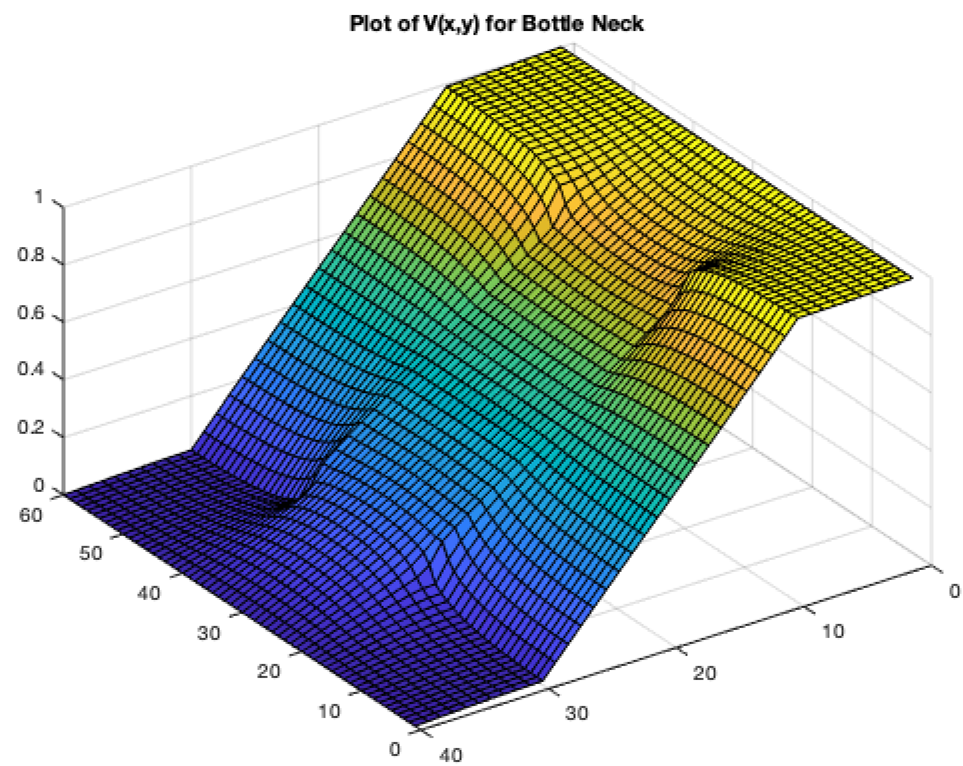
$$J(x, y) = \sigma * V(x, y)$$

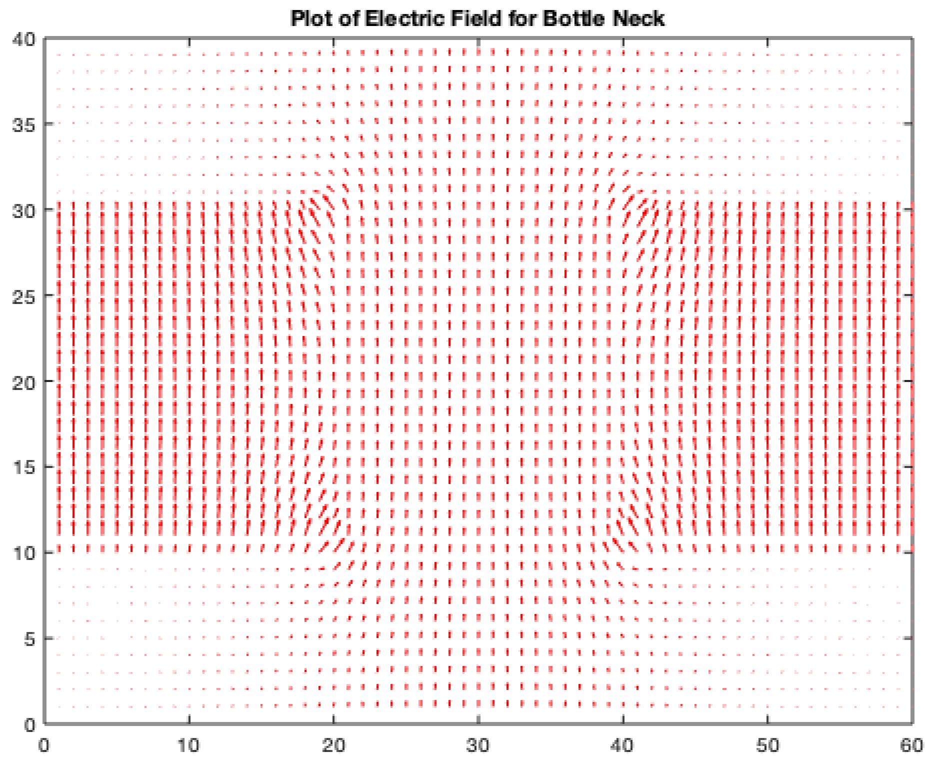
```
% Define Matrix

nx = 40;
ny = 60;
G = sparse(nx*ny,nx*ny);
B = zeros(nx*ny,1);
S = ones(nx,ny);


for i = 1:nx
    for j = 1:ny
        if i >= 10 && i <= 30
            if j <= 20 || j >= 40
                S(i,j) = 1e-2;
            end
        end
    end
end

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

        if i == 1
            G(n,:) = 0;
            G(n,n) = 1;
            B(n) = V0;
        elseif i == nx
            G(n,:) = 0;
            G(n,n) = 1;
            B(n) = 0;


        elseif j == 1
            nxm = j + (i-2)*ny;
```

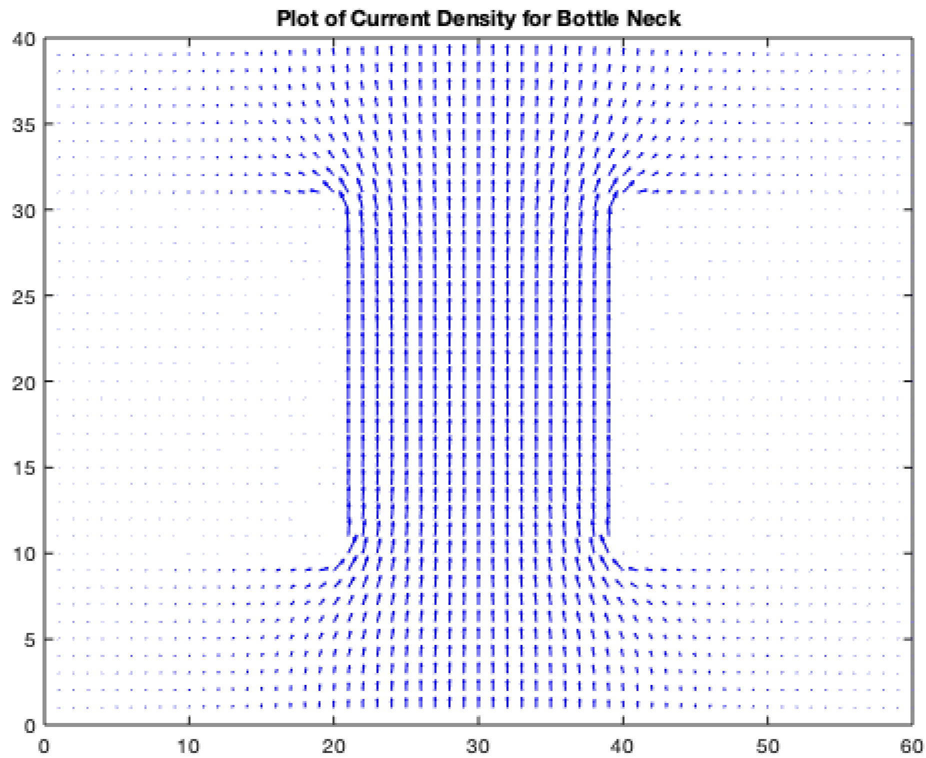
```
            nxp = j + (i)*ny;
            nyp = j+1 + (i-1)*ny;

            rxm = (S(i,j) + S(i-1,j))/2;
            rxp = (S(i,j) + S(i+1,j))/2;
            ryp = (S(i,j) + S(i,j+1))/2;

            G(n,n) = -(rxm+rxp+ryp);
            G(n,nxm) = rxm;
            G(n,nxp) = rxp;
            G(n,nyp) = ryp;

        elseif j == ny
            nxm = j + (i-2)*ny;
            nxp = j + (i)*ny;
            nym = j-1 + (i-1)*ny;

            rxm = (S(i,j) + S(i-1,j))/2;
            rxp = (S(i,j) + S(i+1,j))/2;
            rym = (S(i,j) + S(i,j-1))/2;

            G(n,n) = -(rxm+rxp+rym);
            G(n,nxm) = rxm;
            G(n,nxp) = rxp;
            G(n,nym) = rym;

        else
            nxm = j + (i-2)*ny;
            nxp = j + (i)*ny;
            nym = j-1 + (i-1)*ny;
            nyp = j+1 + (i-1)*ny;

            rxm = (S(i,j) + S(i-1,j))/2;
            rxp = (S(i,j) + S(i+1,j))/2;
            rym = (S(i,j) + S(i,j-1))/2;
            ryp = (S(i,j) + S(i,j+1))/2;

            G(n,n) = -(rxm+rxp+rym+ryp);
            G(n,nxm) = rxm;
            G(n,nxp) = rxp;
            G(n,nym) = rym;
            G(n,nyp) = ryp;
        end
    end
end

solve = G\B;
data_z = zeros(nx,ny);
data_x = linspace(1,nx,nx);
data_y = linspace(1,ny,ny);

figure(5)
surf(data_y,data_x,S)
title('Plot of Conduction for Bottle Neck')
```

```
for i = 1:nx
    for j = 1:ny
        position = j + (i-1)*ny;
        data_z(i,j) = solve(position,1);
    end
end

figure(6)
surf(data_y,data_x,data_z)
view(-125,45)
title('Plot of V(x,y) for Bottle Neck')

% Since the electric field can is related to the change in voltage
 over a
% given distance, we can find the E field in both the x and the y
% directions by taking the derivative (gradient) of the obtained
 voltage
% matrix in these directions.

[Ex,Ey] = gradient(-1.*data_z);

figure(7)
quiver(Ex, Ey, 'r')
xlim([0 60])
ylim([0 40])
title('Plot of Electric Field for Bottle Neck')

% The current density in a material can be obtained by multiplying the
% conduction of that material at any given point by the electric field
 at
% that same point.

Jx = S.*Ex;
Jy = S.*Ey;

figure(8)
quiver(Jx, Jy, 'b')
xlim([0 60])
ylim([0 40])
title('Plot of Current Density for Bottle Neck')
```


Plot of Conduction for Bottle Neck
1
0.8
0.6
0.4
0.2
0
40
30
20
10
0
0
10
20
30
40
50
60


Plot of V(x,y) for Bottle Neck
1
0.8
0.6
0.4
0.2
0
60
50
40
30
20
10
0
40
30
20
10
0


Plot of Electric Field for Bottle Neck
0
5
10
15
20
25
30
35
40
0
10
20
30
40
50
60


Plot of Current Density for Bottle Neck
0
5
10
15
20
25
30
35
40
0
10
20
30
40
50
60

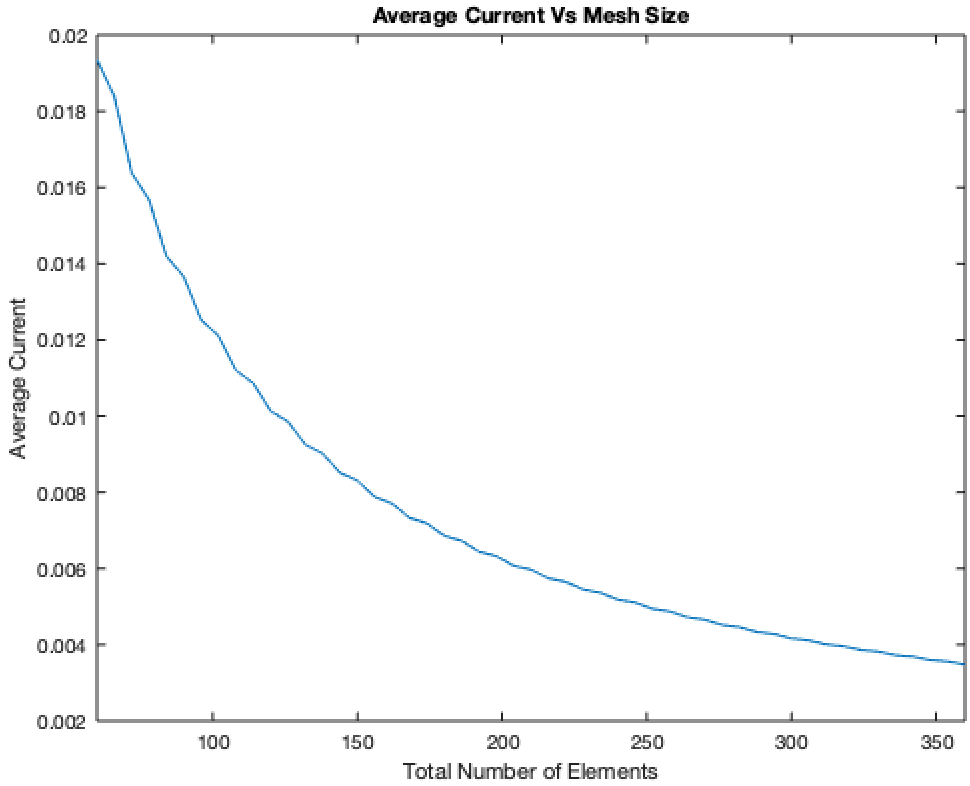
2.b - Varying Mesh Size

The process can then be repeated for varying mesh sizes, and a graph showing the relationship between the mesh size and the average current flow in the material can be obtained.

```
A = zeros(1,51);
for size = 10:60
    nx = 2*size;
    ny = 3*size;
    G = sparse(nx*ny,nx*ny);
    B = zeros(nx*ny,1);
    S = ones(nx,ny);


    for i = 1:nx
        for j = 1:ny
            if i >= (size/2) && i <= (3*size/2)
                if j <= (size) || j >= (2*size)
                    S(i,j) = 1e-2;
                end
            end
        end
    end

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

            if i == 1
                G(n,:) = 0;
                G(n,n) = 1;
                B(n) = V0;
            elseif i == nx
                G(n,:) = 0;
                G(n,n) = 1;
                B(n) = 0;


            elseif j == 1
                nxm = j + (i-2)*ny;
                nxp = j + (i)*ny;
                nyp = j+1 + (i-1)*ny;

                rxm = (S(i,j) + S(i-1,j))/2;
                rxp = (S(i,j) + S(i+1,j))/2;
                ryp = (S(i,j) + S(i,j+1))/2;

                G(n,n) = -(rxm+rxp+ryp);
                G(n,nxm) = rxm;
                G(n,nxp) = rxp;
                G(n,nyp) = ryp;

            elseif j == ny
```

```
                nxm = j + (i-2)*ny;
                nxp = j + (i)*ny;
                nym = j-1 + (i-1)*ny;

                rxm = (S(i,j) + S(i-1,j))/2;
                rxp = (S(i,j) + S(i+1,j))/2;
                rym = (S(i,j) + S(i,j-1))/2;

                G(n,n) = -(rxm+rxp+rym);
                G(n,nxm) = rxm;
                G(n,nxp) = rxp;
                G(n,nym) = rym;

            else
                nxm = j + (i-2)*ny;
                nxp = j + (i)*ny;
                nym = j-1 + (i-1)*ny;
                nyp = j+1 + (i-1)*ny;

                rxm = (S(i,j) + S(i-1,j))/2;
                rxp = (S(i,j) + S(i+1,j))/2;
                rym = (S(i,j) + S(i,j-1))/2;
                ryp = (S(i,j) + S(i,j+1))/2;

                G(n,n) = -(rxm+rxp+rym+ryp);
                G(n,nxm) = rxm;
                G(n,nxp) = rxp;
                G(n,nym) = rym;
                G(n,nyp) = ryp;
            end
        end
    end

    solve = G\B;
    data_z = zeros(nx,ny);

    for i = 1:nx
        for j = 1:ny
            position = j + (i-1)*ny;
            data_z(i,j) = solve(position,1);
        end
    end

    [Ex,Ey] = gradient(-1.*data_z);
    Jx = S.*Ex;
    Jy = S.*Ey;

    % For current we will take the average of the current density
 across the
    % area of the material.

    J = sqrt(Jx.^2 + Jy.^2);
    A(size-9) = mean(mean(J));
end
```

```
figure(9)
plot(linspace(60,360,51),A)
xlim([60 360])
xlabel('Total Number of Elements')
ylabel('Average Current')
title('Average Current Vs Mesh Size')
```



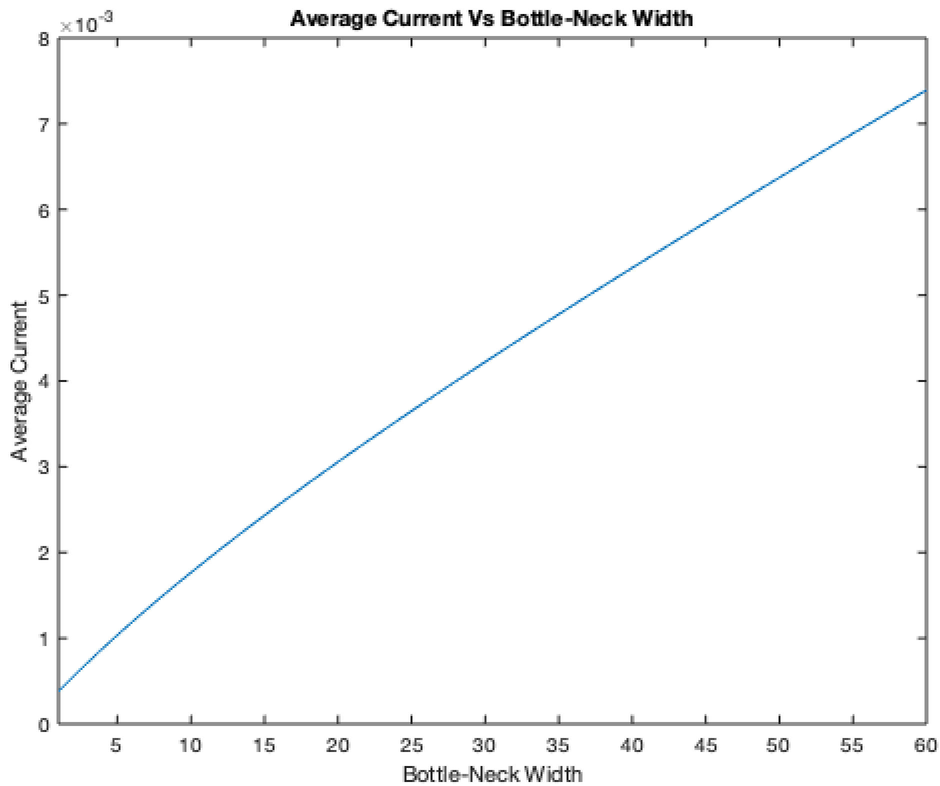
2.c - Varying Bottle Neck Size

The process can then be repeated for varying bottle-neck width, and a graph showing the relationship between the bottle-neck width and the average current flow in the material can be obtained.

```
A = zeros(1,60);
for W = 1:60
    nx = 80;
    ny = 120;
    G = sparse(nx*ny,nx*ny);
    B = zeros(nx*ny,1);
    S = ones(nx,ny);


    for i = 1:nx
        for j = 1:ny
            if i >= (32) && i <= (3*size/2)
                if j <= 60-(W/2) || j > 60+(W/2)
                    S(i,j) = 1e-2;
```

```
            end
        end
    end
end

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

        if i == 1
            G(n,:) = 0;
            G(n,n) = 1;
            B(n) = V0;
        elseif i == nx
            G(n,:) = 0;
            G(n,n) = 1;
            B(n) = 0;


        elseif j == 1
            nxm = j + (i-2)*ny;
            nxp = j + (i)*ny;
            nyp = j+1 + (i-1)*ny;

            rxm = (S(i,j) + S(i-1,j))/2;
            rxp = (S(i,j) + S(i+1,j))/2;
            ryp = (S(i,j) + S(i,j+1))/2;

            G(n,n) = -(rxm+rxp+ryp);
            G(n,nxm) = rxm;
            G(n,nxp) = rxp;
            G(n,nyp) = ryp;

        elseif j == ny
            nxm = j + (i-2)*ny;
            nxp = j + (i)*ny;
            nym = j-1 + (i-1)*ny;

            rxm = (S(i,j) + S(i-1,j))/2;
            rxp = (S(i,j) + S(i+1,j))/2;
            rym = (S(i,j) + S(i,j-1))/2;

            G(n,n) = -(rxm+rxp+rym);
            G(n,nxm) = rxm;
            G(n,nxp) = rxp;
            G(n,nym) = rym;

        else
            nxm = j + (i-2)*ny;
            nxp = j + (i)*ny;
            nym = j-1 + (i-1)*ny;
            nyp = j+1 + (i-1)*ny;

            rxm = (S(i,j) + S(i-1,j))/2;
```

```
                rxp = (S(i,j) + S(i+1,j))/2;
                rym = (S(i,j) + S(i,j-1))/2;
                ryp = (S(i,j) + S(i,j+1))/2;

                G(n,n) = -(rxm+rxp+rym+ryp);
                G(n,nxm) = rxm;
                G(n,nxp) = rxp;
                G(n,nym) = rym;
                G(n,nyp) = ryp;
            end
        end
    end

    solve = G\B;
    data_z = zeros(nx,ny);

    for i = 1:nx
        for j = 1:ny
            position = j + (i-1)*ny;
            data_z(i,j) = solve(position,1);
        end
    end

    [Ex,Ey] = gradient(-1.*data_z);
    Jx = S.*Ex;
    Jy = S.*Ey;

    % For current we will take the average of the current density
 across the
    % area of the material.

    J = sqrt(Jx.^2 + Jy.^2);
    A(W) = mean(mean(J));
end
figure(10)
plot(linspace(1,60,60),A)
xlim([1 60])
xlabel('Bottle-Neck Width')
ylabel('Average Current')
title('Average Current Vs Bottle-Neck Width')
```



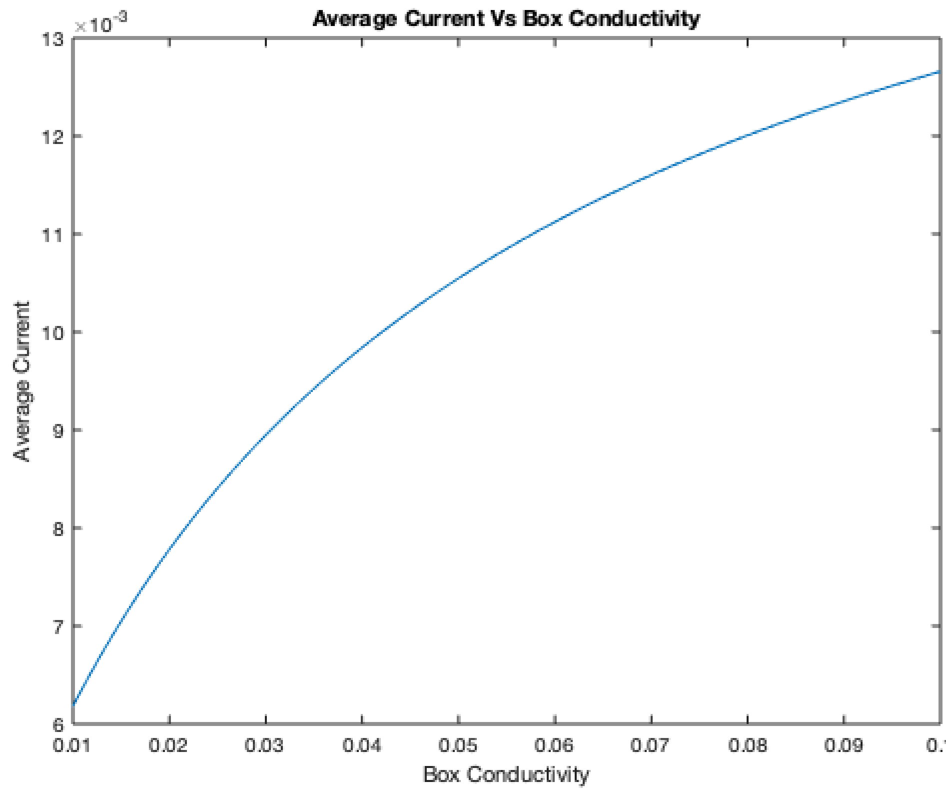
2.d - Varying Box Conductivity

The process can then be repeated for varying box conductivity, and a graph showing the relationship between the conductivity and the average current flow in the material can be obtained.

```
A = zeros(1,100);
for C = 1:100
    nx = 80;
    ny = 120;
    G = sparse(nx*ny,nx*ny);
    B = zeros(nx*ny,1);
    S = ones(nx,ny);


    for i = 1:nx
        for j = 1:ny
            if i >= 10 && i <= 30
                if j <= 20 || j >= 40
                    S(i,j) = 0.01*C;
                end
            end
        end
    end

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

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

if i == 1
    G(n,:) = 0;
    G(n,n) = 1;
    B(n) = V0;
elseif i == nx
    G(n,:) = 0;
    G(n,n) = 1;
    B(n) = 0;


elseif j == 1
    nxm = j + (i-2)*ny;
    nxp = j + (i)*ny;
    nyp = j+1 + (i-1)*ny;

    rxm = (S(i,j) + S(i-1,j))/2;
    rxp = (S(i,j) + S(i+1,j))/2;
    ryp = (S(i,j) + S(i,j+1))/2;

    G(n,n) = -(rxm+rxp+ryp);
    G(n,nxm) = rxm;
    G(n,nxp) = rxp;
    G(n,nyp) = ryp;

elseif j == ny
    nxm = j + (i-2)*ny;
    nxp = j + (i)*ny;
    nym = j-1 + (i-1)*ny;

    rxm = (S(i,j) + S(i-1,j))/2;
    rxp = (S(i,j) + S(i+1,j))/2;
    rym = (S(i,j) + S(i,j-1))/2;

    G(n,n) = -(rxm+rxp+rym);
    G(n,nxm) = rxm;
    G(n,nxp) = rxp;
    G(n,nym) = rym;

else
    nxm = j + (i-2)*ny;
    nxp = j + (i)*ny;
    nym = j-1 + (i-1)*ny;
    nyp = j+1 + (i-1)*ny;

    rxm = (S(i,j) + S(i-1,j))/2;
    rxp = (S(i,j) + S(i+1,j))/2;
    rym = (S(i,j) + S(i,j-1))/2;
    ryp = (S(i,j) + S(i,j+1))/2;

    G(n,n) = -(rxm+rxp+rym+ryp);
    G(n,nxm) = rxm;
    G(n,nxp) = rxp;
```

```
                G(n,nym) = rym;
                G(n,nyp) = ryp;
            end
        end
    end

    solve = G\B;
    data_z = zeros(nx,ny);

    for i = 1:nx
        for j = 1:ny
            position = j + (i-1)*ny;
            data_z(i,j) = solve(position,1);
        end
    end

    [Ex,Ey] = gradient(-1.*data_z);
    Jx = S.*Ex;
    Jy = S.*Ey;

    % For current we will take the average of the current density
 across the
    % area of the material.

    J = sqrt(Jx.^2 + Jy.^2);
    A(C) = mean(mean(J));
end
figure(11)
plot(linspace(0.001,0.10,100),A)
xlim([0.01 0.1])
xlabel('Box Conductivity')
ylabel('Average Current')
title('Average Current Vs Box Conductivity')
```



Published with MATLAB® R2019b