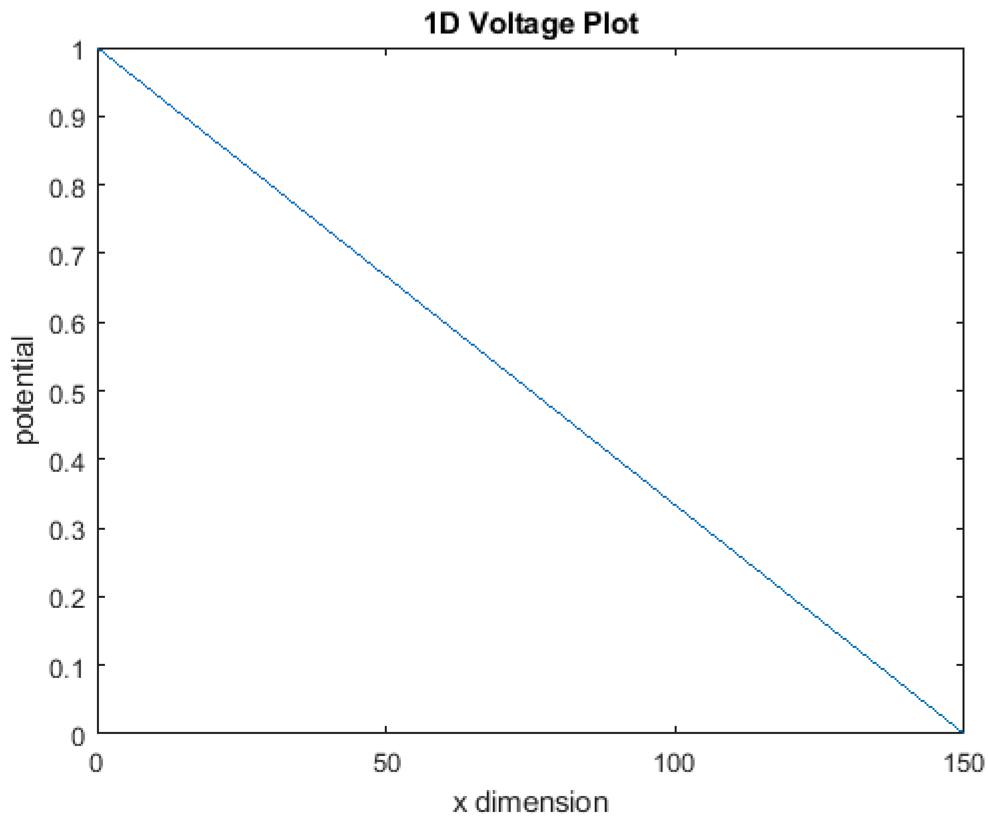
Laplace's Equation - Part 1: One Dimensional Case

The first part of this problem is a solution to Laplace's equation along a 1-D section of a material with one side connected to a voltage source and the other end grounded. We expect to have a linear solution here. We are not using boundary conditions where the y-derivative of the potential is smooth at the upper boundary, so this is strictly a 1-D problem.

```
%initializing matrices
nx = 150;
ny = 100;
G = sparse(nx);
B = zeros(1,nx);
for i = 1:nx
    %Set up G-Matrix
     if i == 1
        G(i,i) = 1;
        B(i) = 1;
     elseif i == nx
        G(i,i) = 1;
     else
        G(i,i+1) = 1;
        G(i,i-1) = 1;
        G(i,i) = -2;
    end
end
%solve for V and plot
V = G\B';
X = linspace(0,150,length(V));
figure(1)
plot(X,V)
title('1D Voltage Plot')
xlabel('x dimension')
ylabel('potential')
```



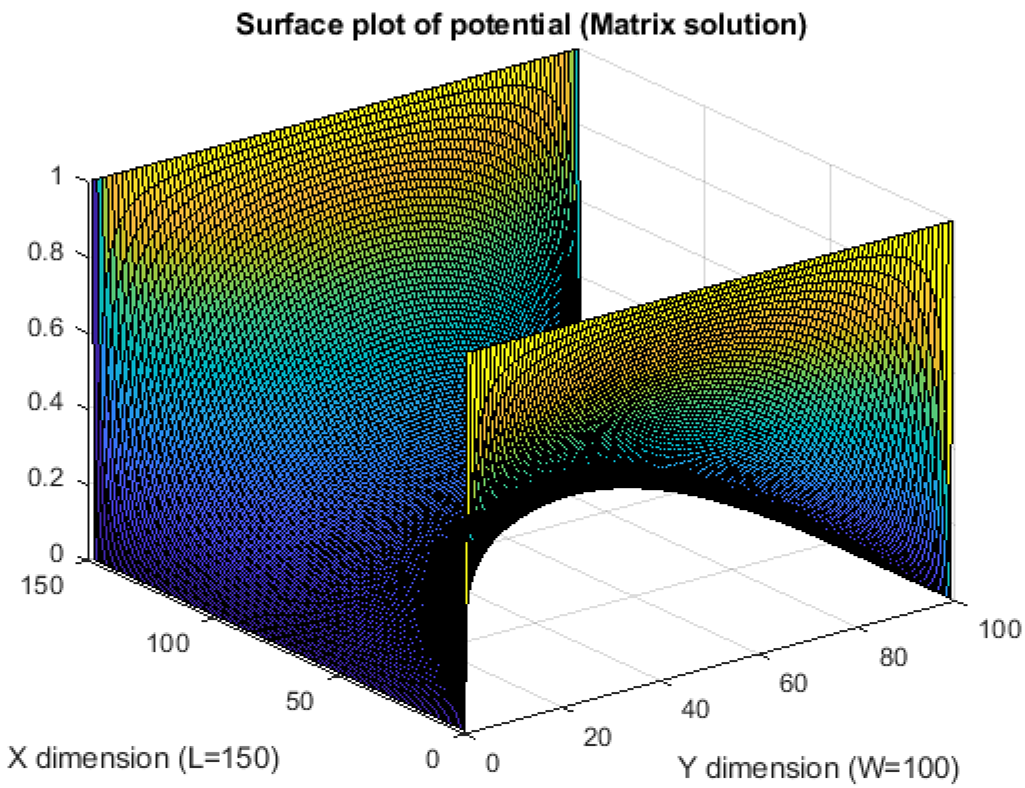
Published with MATLAB® R2018a

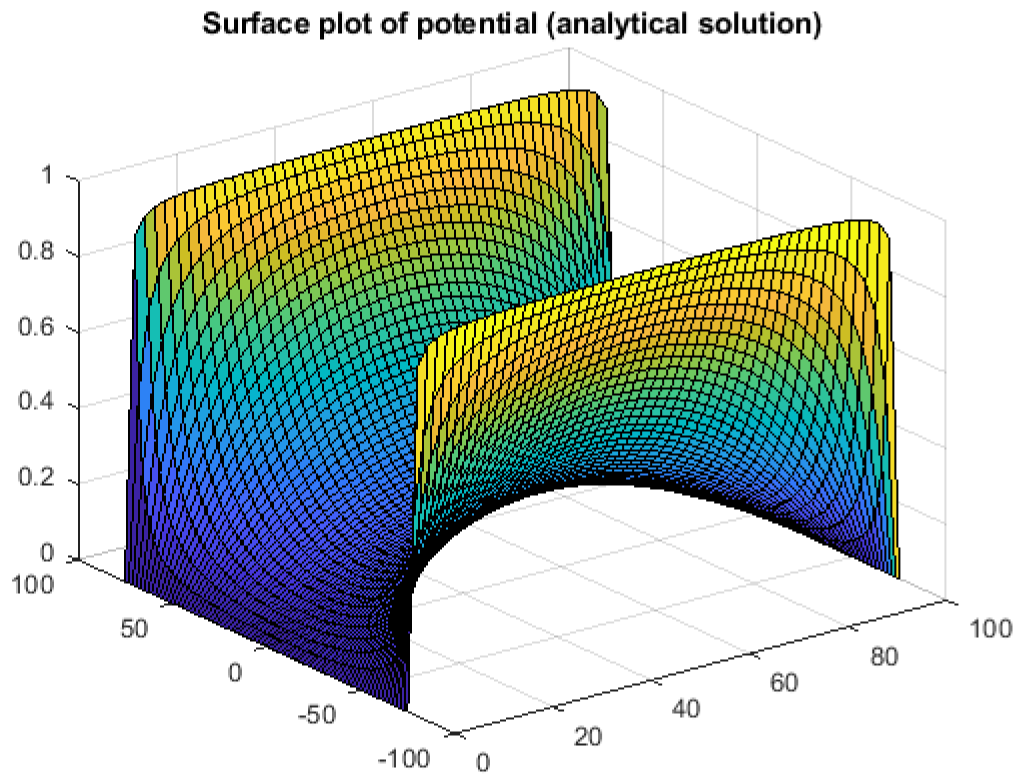
Part Two: Two Dimensional Case

Here we are solving a 2-dimensional version of Laplace's equation in what is essentially a rectangular pipe. Our goal is to compare our numerical solution to the analytical solution that can be obtained with a separation of variables and fourier analysis. Solving the differential equations on paper can be annoying, so a numerical solution is appropriate here. This approach would be significantly harder in a different coordinate system, but for rectangular coordinates, its OK. There is an inherent error involved with both methods. The analytical solution involves an infinite sum, but since that's not physically realizable, I cut it off at 100. On my plot of the analytical solution, the top edges are still rounded at the corners, and they aren't at perfectly 90 degrees. The finite difference solution I obtained looks really good, but with a wider mesh it would look much less smooth.

```
nx = 150;
ny = 100;
G = sparse(nx*ny, nx*ny);
B = zeros(1,nx*ny);
cMap = ones(nx,ny);

for i = 1:nx
    for j = 1:ny
        %Setting up the G-Matrix
        n = j + (i-1)*ny;
        if i == 1
            G(n,n) = 1;
            B(n) = 1;
        elseif i == nx
            G(n,n) = 1;
            B(n) = 1;
        elseif j == 1
            G(n,n) = 1;
        elseif j == ny
            G(n,n) = 1;
        else
            nxm = j + (i-2)*ny;
            nxp = j + (i)*ny;
            nym = j-1 + (i-1)*ny;
            nyp = j+1 + (i-1)*ny;
            G(n,nxp) = 1;
            G(n,nyp) = 1;
            G(n,nxm) = 1;
            G(n,nym) = 1;
            G(n,n) = -4;
        end
    end
end
%solving for matrix of potentials
V = G\B';
%remapping back to i,j
Vmap = zeros(nx,ny);
for i = 1:nx
    for j = 1:ny
        n = j + (i-1)*ny;
```

```
        Vmap(i,j) = V(n);
    end
end
%plotting V(x,y)
 figure(2)
 surf(Vmap)
 title('Surface plot of potential (Matrix solution)')
 ylabel('X dimension (L=150)')
 xlabel('Y dimension (W=100)')

%analytical solution
[x, y] = meshgrid(-75:2:75, 0:2:100);
a = 100;
b = 75;
V_an = 0;

for N = 1:100
    if rem(N,2) == 1
        V_an = V_an + (4/pi)*(cosh(N*pi*x/a).*sin(N*pi*y/a)) ./
 (N*cosh(N*pi*b/a));
        figure(3)
        surf(y,x,V_an)
        title('Surface plot of potential (analytical solution)')
        pause(0.01)
    end
end
```





Published with MATLAB® R2018a

Part 3 - Bottleneck Analysis

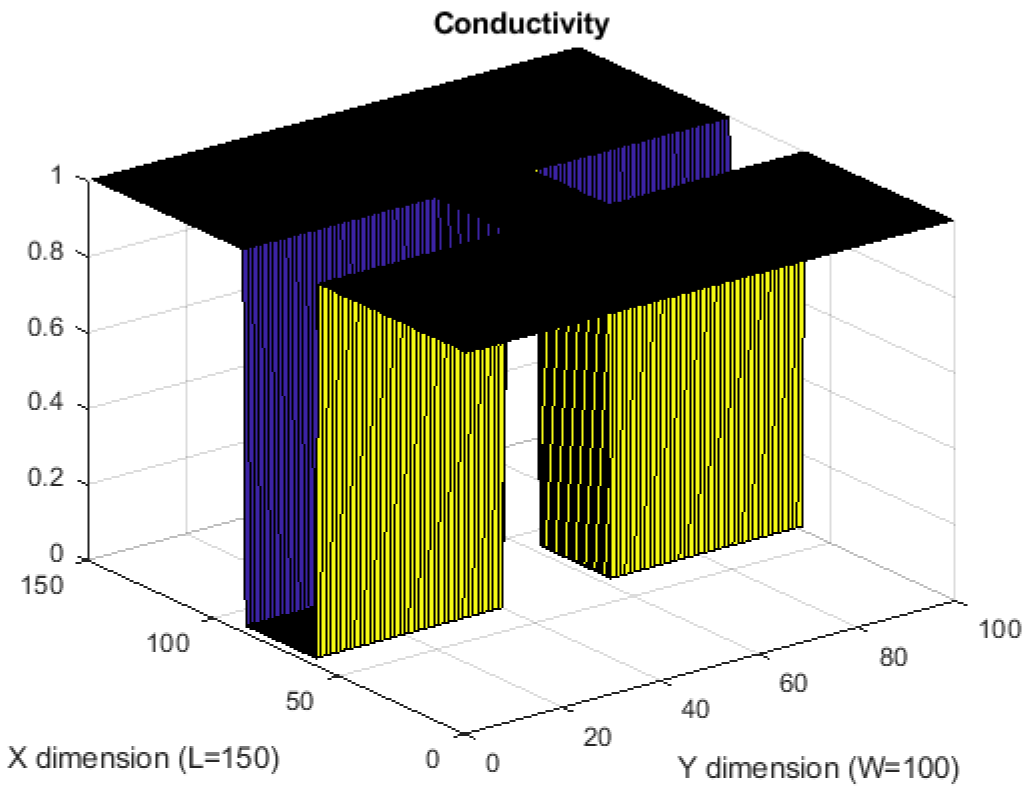
In this part, a bottleneck of a different material is introduced into the medium. The bottleneck has a significantly lower conductivity than the surrounding medium, so its expected that less current should flow through it. Graphs for conductivity, electric potential, electric field, and current density are produced. By altering the values of nx and ny, we can play around with the mesh density. normally, the program is run with nx = 150, and ny = 100. By reducing both by a factor of 10, the total current flow increased by over 50% from 0.314A to 0.452A. Naturally when the mesh is made more dense, the current is lower. Narrowing the bottleneck will also have an effect on the total current output. I reduced the width of the opening by a factor of 2, and the current went from 0.314A to 0.220A. The current continues to drop off as the bottleneck is made thinner, as expected. This essentially is increasing the resistance of the material. As the conductivity of the bottleneck is changed, then the total current will change as well. As the conductivity approaches 1, it essentially behaves as if there is no bottleneck whatsoever since the surrounding medium has a conductivity of 1. It follows naturally that as the conductivity of the bottleneck approaches zero, the current density will approach a steady state value where it can't really interact with the bottleneck at all.

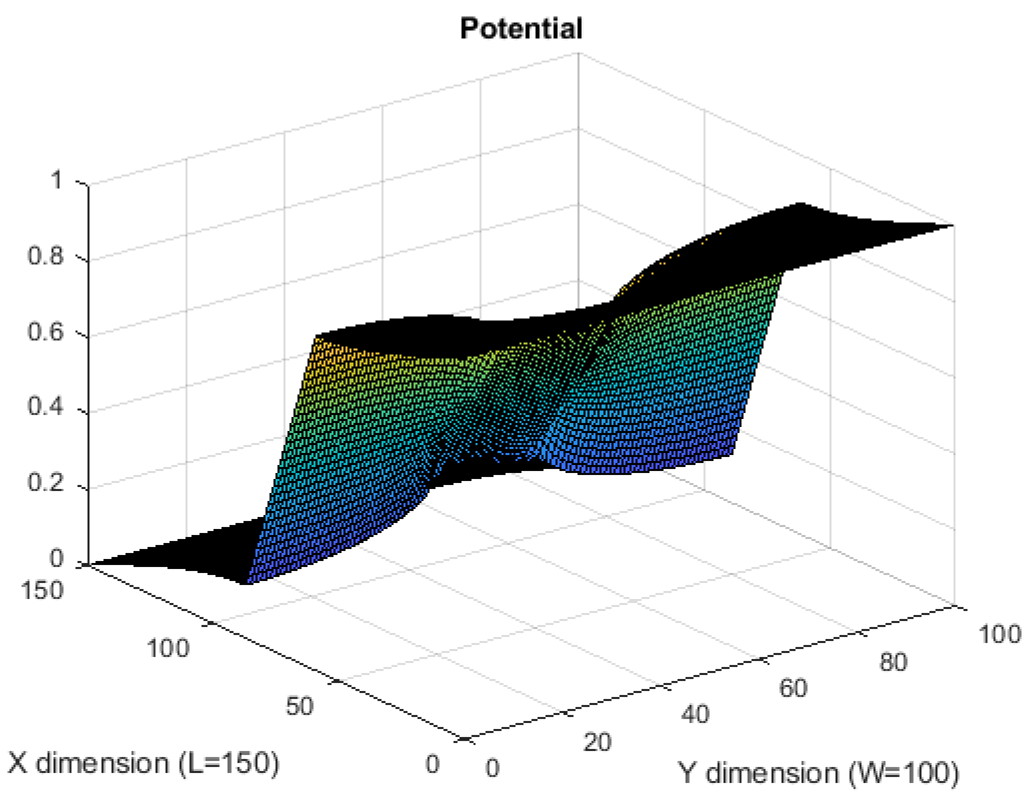
```
nx = 150;
ny = 100;
G = sparse(nx*ny, nx*ny);
B = zeros(1,nx*ny);
%setting conductivity map for 3rd part
cMap = ones(nx,ny);
 for q = 1:nx
     for w = 1:ny
         if ((q<(0.6*nx)&&q>(0.4*nx)&&w>(0.6*ny)) ||
 (q<(0.6*nx)&&q>(0.4*nx)&&w<(0.4*ny)))
             cMap(q,w) = 1e-2;
         end
     end
 end

for i = 1:nx
    for j = 1:ny
        %Setting up the G-Matrix
        n = j + (i-1)*ny;
        if i == 1
            G(n,n) = 1;
            B(n) = 1;
        elseif i == nx
            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 = (cMap(i,j) + cMap(i-1,j))/2.0;
            rxp = (cMap(i,j) + cMap(i+1,j))/2.0;
            ryp = (cMap(i,j) + cMap(i,j+1))/2.0;

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

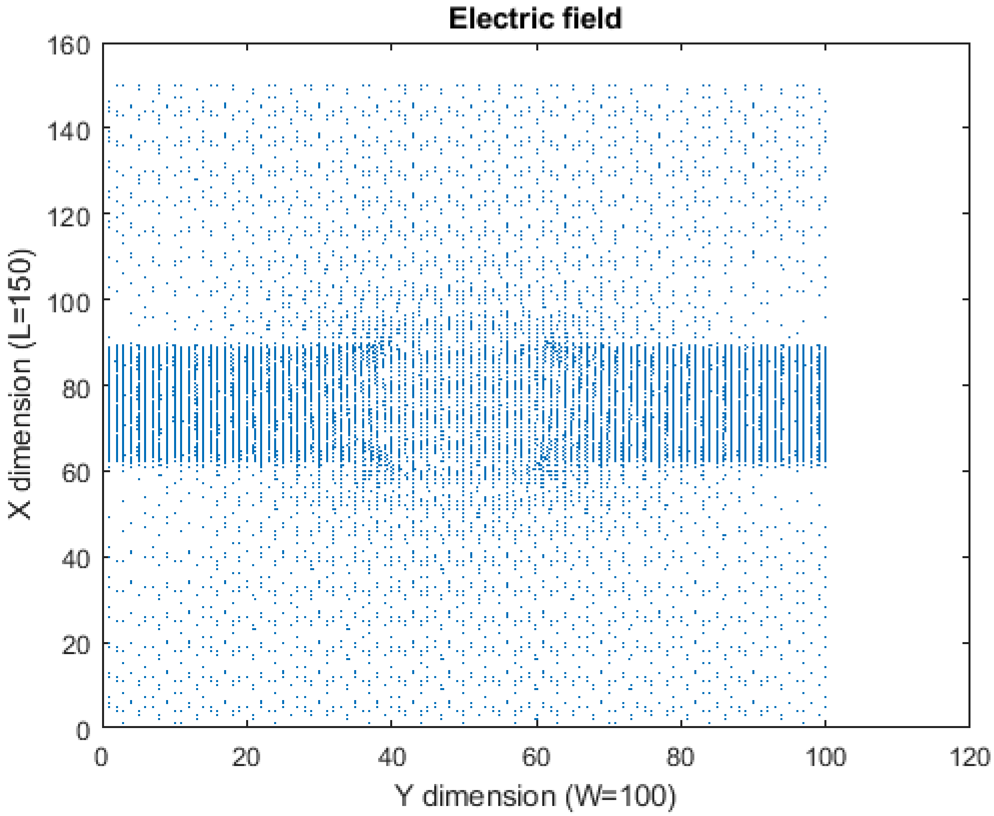
```
            G(n,nxp) = rxp;
            G(n,nyp) = ryp;
        elseif j == ny
            nxm = j + (i-2)*ny;
            nxp = j + (i)*ny;
            nym = j-1 + (i-1)*ny;

            rxm = (cMap(i,j) + cMap(i-1,j))/2.0;
            rxp = (cMap(i,j) + cMap(i+1,j))/2.0;
            rym = (cMap(i,j) + cMap(i,j-1))/2.0;

            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 = (cMap(i,j) + cMap(i-1,j))/2.0;
            rxp = (cMap(i,j) + cMap(i+1,j))/2.0;
            rym = (cMap(i,j) + cMap(i,j-1))/2.0;
            ryp = (cMap(i,j) + cMap(i,j+1))/2.0;

            G(n,nxp) = rxp;
            G(n,nyp) = ryp;
            G(n,nxm) = rxm;
            G(n,nym) = rym;
            G(n,n) = -(rxm + rxp + rym + ryp);
        end
    end
end
%solving for matrix of potentials
V = G\B';
Vmap = zeros(nx,ny);
for i = 1:nx
    for j = 1:ny
        n = j + (i-1)*ny;
        Vmap(i,j) = V(n);
    end
end

%conductivity map plot
figure(4)
surf(cMap)
title('Conductivity')
ylabel('X dimension (L=150)')
xlabel('Y dimension (W=100)')

%plotting potential
figure(5)
surf(Vmap)
```

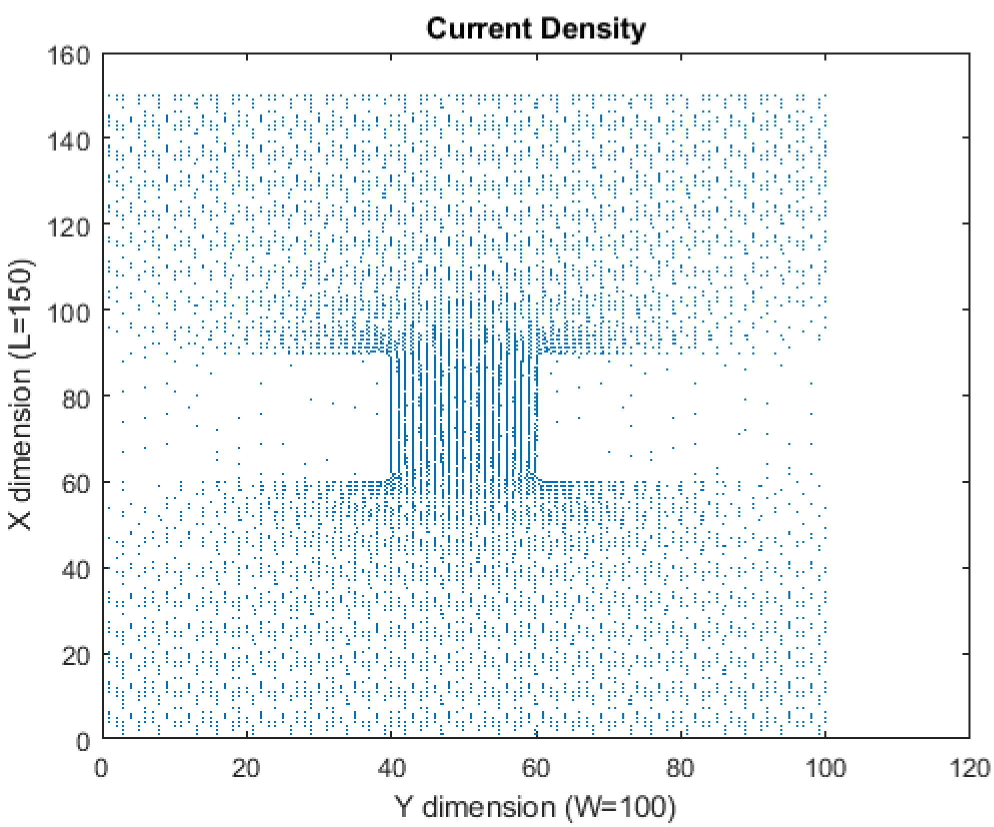
```
title('Potential')
ylabel('X dimension (L=150)')
xlabel('Y dimension (W=100)')

%plotting electric field
[Ex,Ey] = gradient(-Vmap);
figure(6)
quiver(Ex,Ey)
title('Electric field')
ylabel('X dimension (L=150)')
xlabel('Y dimension (W=100)')

%plotting current density
Jx = Ex .* cMap;
Jy = Ey .* cMap;
figure(7)
quiver(Jx,Jy)
title('Current Density')
ylabel('X dimension (L=150)')
xlabel('Y dimension (W=100)')

 %calculating current flow
 J = sqrt(Jx.*Jx+Jy.*Jy);
 C0 = sum(J(1,:));
 Cnx = sum(J(nx,:));
 Current = (C0 + Cnx)*0.5;
 fprintf('the total current flow is %f amps\n',Current);
```

the total current flow is 0.314268 amps


Conductivity
1
0.8
0.6
0.4
0.2
0
150
100
50
0
X dimension (L=150)
0
20
40
60
80
100
Y dimension (W=100)


Potential
1
0.8
0.6
0.4
0.2
0
150
100
50
0
X dimension (L=150)
0
20
40
60
80
100
Y dimension (W=100)


Electric field
160
140
120
100
80
60
40
20
0
X dimension (L=150)
0
20
40
60
80
100
120
Y dimension (W=100)


Current Density
160
140
120
100
80
60
40
20
0
X dimension (L=150)
0
20
40
60
80
100
120
Y dimension (W=100)

Published with MATLAB® R2018a