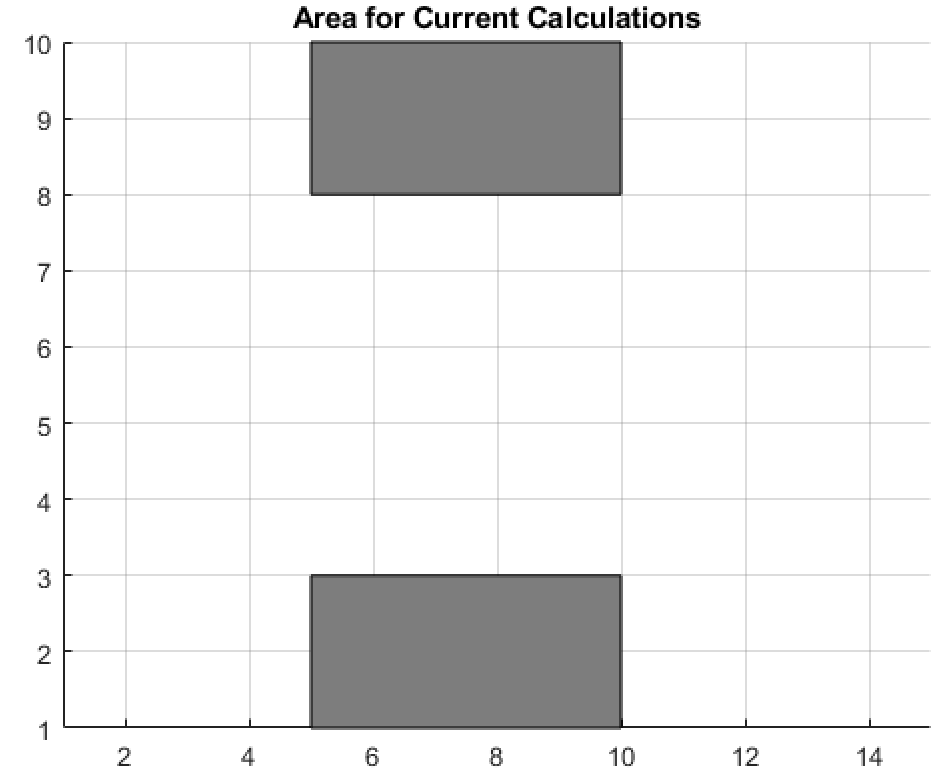
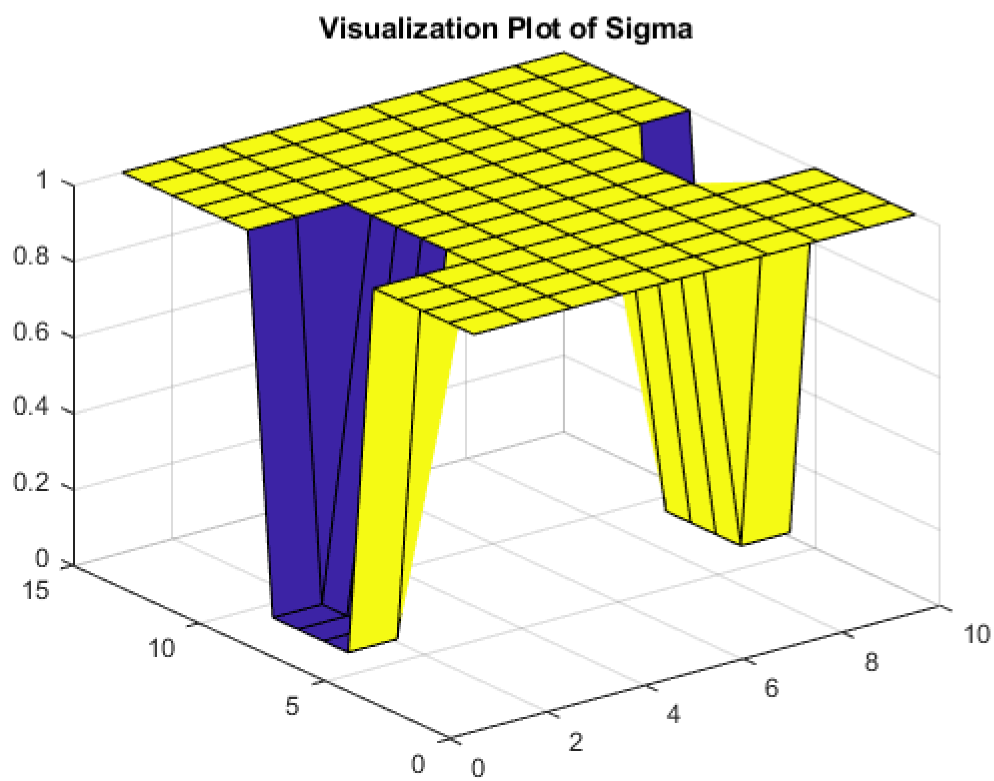
```
clear
clc
close all

global C

%Gotta include the constants just in case!
C.q_0 = 1.60217653e-19;             % electron charge
C.hb = 1.054571596e-34;             % Dirac constant
C.h = C.hb * 2 * pi;                % Planck constant
C.m_0 = 9.10938215e-31;             % electron mass
C.kb = 1.3806504e-23;               % Boltzmann constant
C.eps_0 = 8.854187817e-12;          % vacuum permittivity
C.mu_0 = 1.2566370614e-6;           % vacuum permeability
C.c = 299792458;                    % speed of light
C.g = 9.80665;                      %metres (32.1740 ft) per s


%Change to change area of calculation
nx = 15;       %length of boxed area
ny = 10;       %width of boxed area
lengthBox = 5;     %length of bottle neck box
heightBox = 2;     %height of bottle neck box



G = zeros(nx*ny, nx*ny);        %Create the G matrix to be used
B = zeros(nx*ny,1);             %Boundary condition vector
sigma = zeros(nx,ny);           %Conductivity Matrix
sigmaOutOfBox = 1;              %Free space conductivity
sigmaInBox = .01;               %Box conductivity
voltagePlot1 = zeros(nx,ny);    %Voltage matrix pre-initialization to
 save time

%Set the area for the calculations and plot a visualization of the
 region
figure(1)
%Bottom Box
rectangle ('position', [(nx/2-lengthBox/2)  1 lengthBox
 heightBox],'FaceColor',[.5 .5 .5],'EdgeColor','k');
%Top Box
rectangle ('position', [(nx/2-lengthBox/2)  (ny-heightBox) lengthBox
 heightBox],'FaceColor',[.5 .5 .5],'EdgeColor','k');
grid on;
xlim([1 nx]);
ylim ([1 ny]);
title ('Area for Current Calculations')


%Used to calculate sthe N value for the G matrix
fn = @(i, j) j + (i-1)*ny;
```

```
for w = 1:nx % x
    for s = 1:ny % y
        %Calculate the n values to be used for G matrix
        n = fn(w, s);
        nxm = fn(w-1, s);
        nxp = fn(w+1, s);
        nym = fn(w, s-1);
        nyp = fn(w, s+1);
        if w == 1                            %Check Left boundary
            sigma(w,s) = sigmaOutOfBox;
            G(n, n) = 1;
            B(n) = 1;
        elseif w == nx                       %Check right boundary
            sigma(w,s) = sigmaOutOfBox;
            G(n, n) = 1;
            B(n) = 0;
        elseif (s == 1 || s == ny)           %Check to see if top or
 bottom

            %Check to see if the x location on this axis is in a box
            if (w > (nx/2-lengthBox/2) && w < (nx/2+lengthBox/2))
                sigma(w,s) = sigmaInBox;
                G(n, n) = -3*sigmaInBox;
                G(n, nxp) = sigmaInBox;
                G(n, nxm) = sigmaInBox;
                G(n, nyp) = sigmaInBox;
            else        %Wasn't in a box!
                sigma(w,s) = sigmaOutOfBox;
                G(n, n) = -3*sigmaOutOfBox;
                G(n, nxp) = sigmaOutOfBox;
                G(n, nxm) = sigmaOutOfBox;
                G(n, nyp) = sigmaOutOfBox;
            end
        else
            G(n,n) = -4;
            %Check to see if in a box again (not on axis boundary)
           if (w > (nx/2-lengthBox/2) && w < (nx/2+lengthBox/2))
               %Check to see if y component is in box
               if ((s > 1 && (s < heightBox+1)) || (s > (ny-heightBox)
 && s < ny))
                   %We found a box!
                    sigma(w,s) = sigmaInBox;
                    G(n, nxp) = sigmaInBox;
                    G(n, nxm) = sigmaInBox;
                    G(n, nyp) = sigmaInBox;
                    G(n, nym) = sigmaInBox;
               else
                    %We didn't find a box...
                    sigma(w,s) = sigmaOutOfBox;
                    G(n, nxp) = sigmaOutOfBox;
                    G(n, nxm) = sigmaOutOfBox;
                    G(n, nyp) = sigmaOutOfBox;
                    G(n, nym) = sigmaOutOfBox;
               end
```

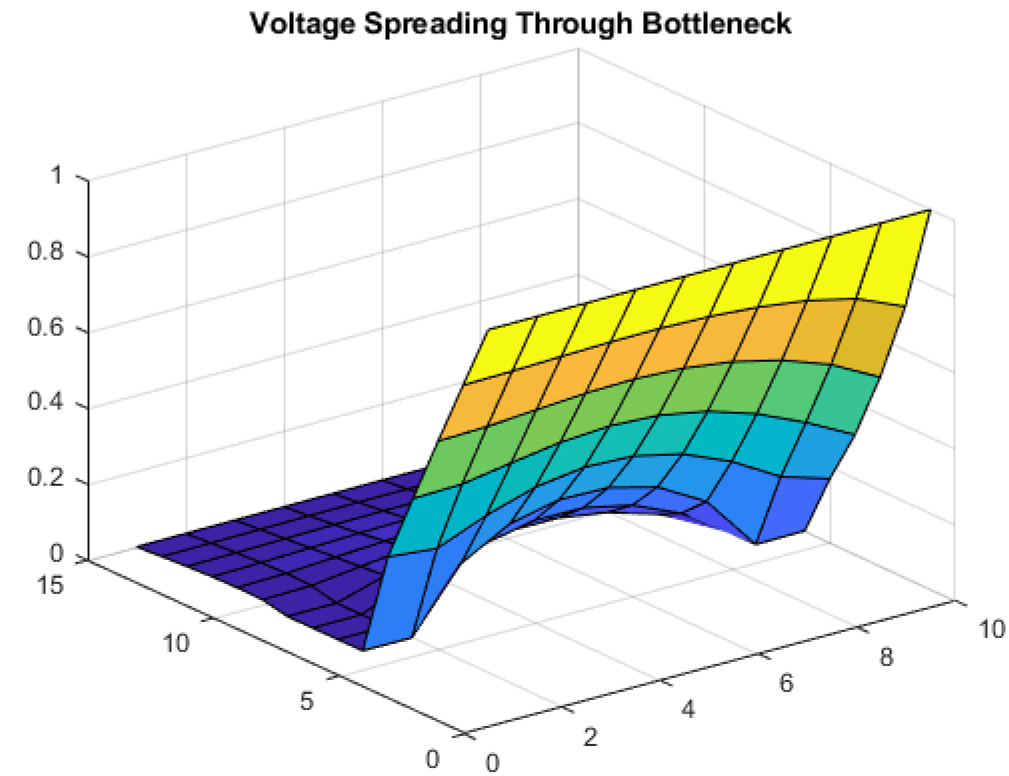
```
            else
                %We still haven't found a box.....
                 sigma(w,s) = sigmaOutOfBox;
                 G(n, nxp) = sigmaOutOfBox;
                 G(n, nxm) = sigmaOutOfBox;
                 G(n, nyp) = sigmaOutOfBox;
                 G(n, nym) = sigmaOutOfBox;
             end
        end
    end
end

figure(2)
surf(sigma)
title('Visualization Plot of Sigma')

%Calculate the voltage matrix from the boundary and G
V=G\B;

%Remap V back to a familiar matrix
for w = 1:nx % x
    for s = 1:ny % y
        n = s+(w-1)*ny;
        voltagePlot1(w, s) = V(n)*sigma(w,s);
    end
end

%Plot the voltage matrix
figure(3)
surf(voltagePlot1)
title('Voltage Spreading Through Bottleneck');

%Calculate and plot the EX electrix field
figure(4)
[ex,ey]=gradient(voltagePlot1);
surf(ex)
title('Electric Field Ex');

%Calculate and plot the EY electrix field
figure(5)
surf(ey)
title('Electric Field Ey');

%Calculate current density and plot
J = sigma .* gradient(voltagePlot1);
figure (6)
surf(J)
title('Current Density')
```



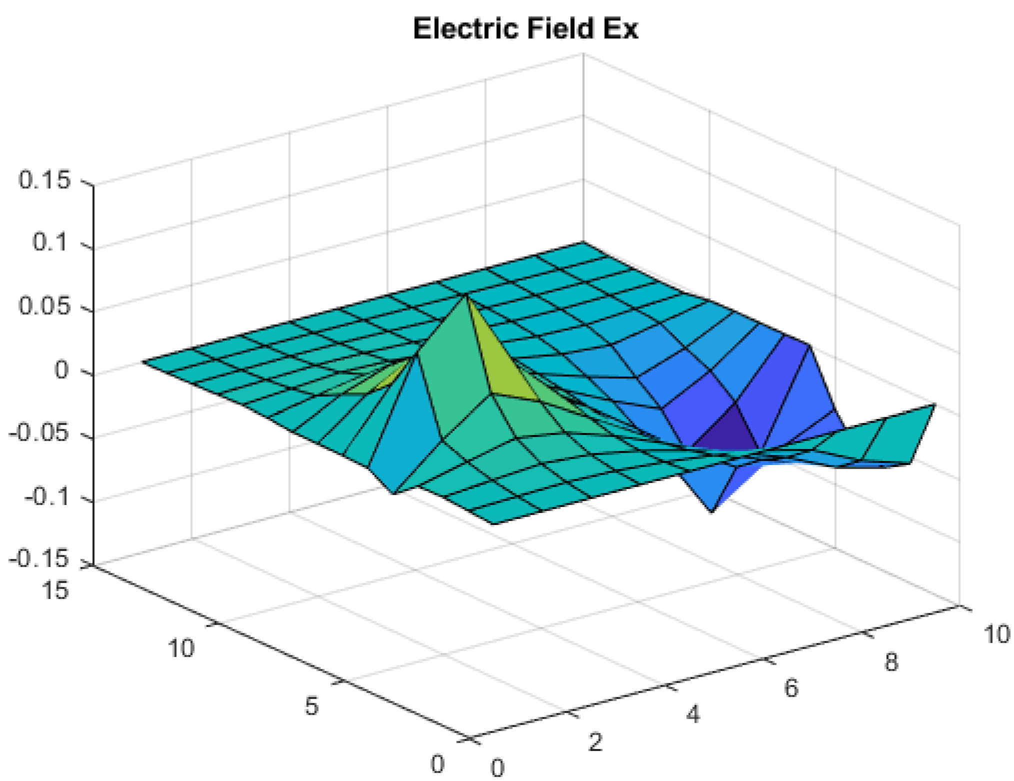
Area for Current Calculations
10
9
8
7
6
5
4
3
2
1
2
4
6
8
10
12
14



Visualization Plot of Sigma
1
0.8
0.6
0.4
0.2
0
15
10
5
0
0
2
4
6
8
10

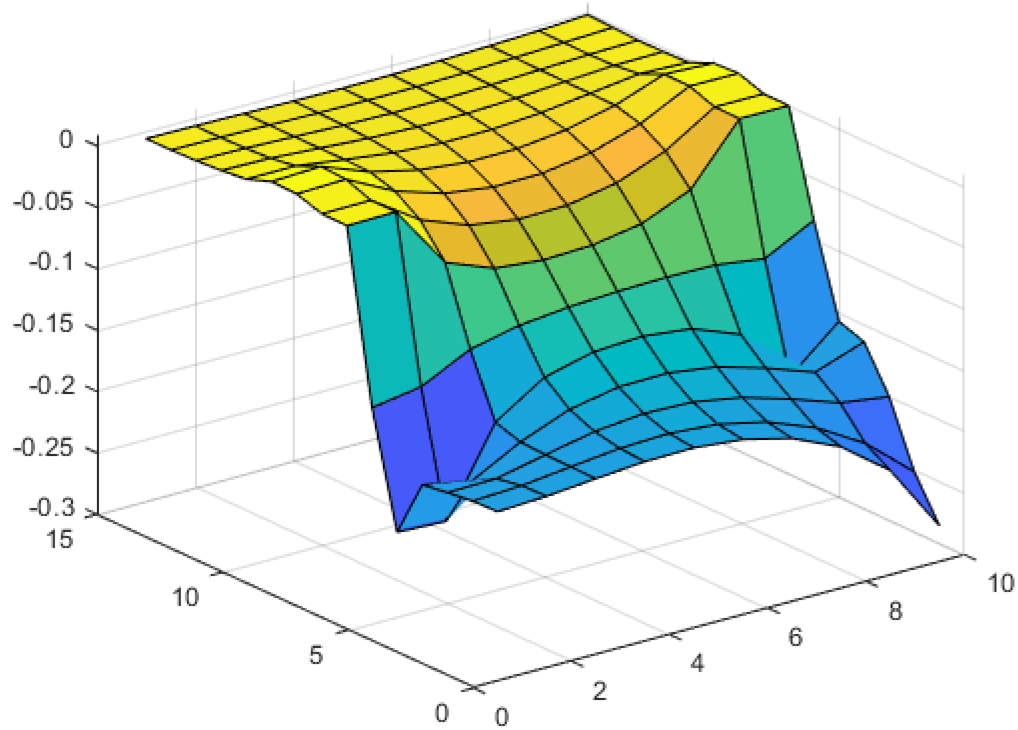


Voltage Spreading Through Bottleneck
1
0.8
0.6
0.4
0.2
0
15
10
5
0
0
2
4
6
8
10

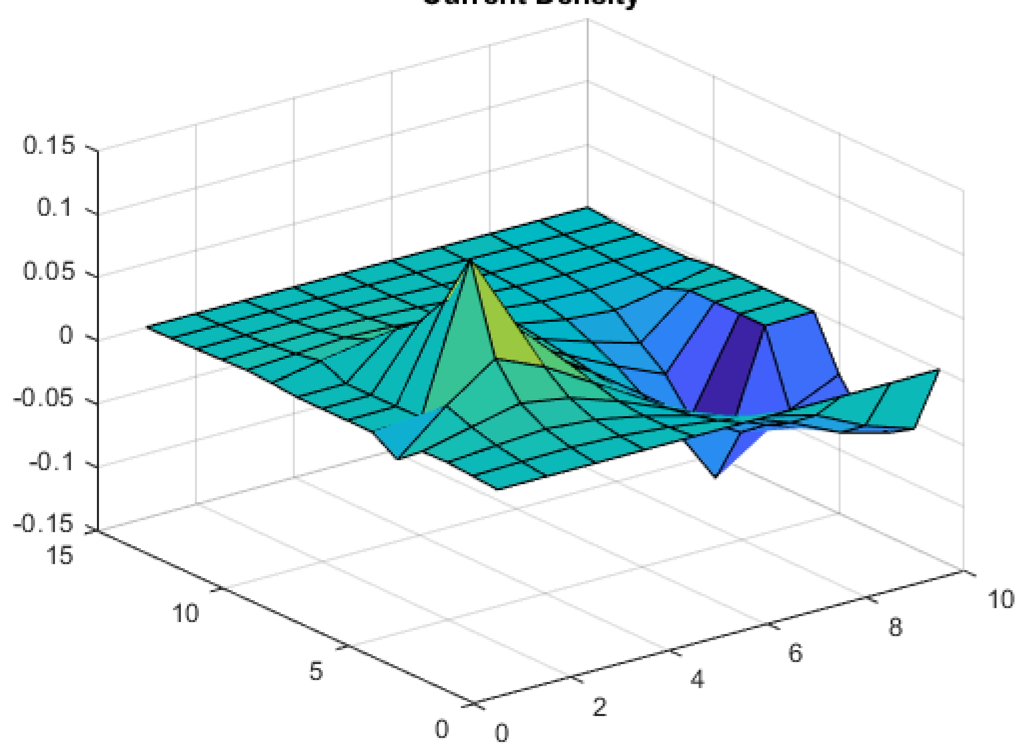


Electric Field Ex
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
15
10
5
0
0
2
4
6
8
10

Electric Field Ey



Current Density



Published with MATLAB® R2018a