```
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

TotalElectrons = 10000;
lastTemperature= 300;
Tau = 0.2e-12;
length = 200e-9;
height = 100e-9;
Px= zeros(1,TotalElectrons);
NewPx= zeros(1,TotalElectrons);
Py= zeros(1,TotalElectrons);
NewPy= zeros(1,TotalElectrons);
Vx= zeros(1,TotalElectrons);
Vy= zeros(1,TotalElectrons);
temperatures = zeros(11,11);
lastLoop=0;
lastDensity=0;


%Change to change area of calculation
nx = 200;       %length of boxed area
ny = 100;       %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 = ones(nx,ny);           %Conductivity Matrix
% sigmaOutOfBox = 1;              %Free space conductivity
sigmaInBox = .01;               %Box conductivity
voltagePlot1 = zeros(nx,ny);    %Voltage matrix pre-initialization to
 save time



%Old Stuff, don't need this
```

```
% %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')
box1 = [80 120  1  30];
box2 = [80 120 70 100];

sigma(box1(1):box1(2), box1(3):box1(4)) = sigmaInBox;
sigma(box2(1):box2(2), box2(3):box2(4)) = sigmaInBox;

%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
            G(n, n) = 1;
            B(n) = 0.1;
        elseif w == nx
            G(n, n) = 1;
            B(n) = 0;
        elseif s == 1
            s1 = (sigma(w,s)+sigma(w+1,s))/1.0;
            s2 = (sigma(w,s)+sigma(w-1,s))/1.0;
            s3 = (sigma(w,s)+sigma(w,s+1))/2.0;

            G(n, n) = -(s1+s2+s3);
            G(n, nxp) = s1;
            G(n, nxm) = s2;
            G(n, nyp) = s3;
        elseif s == ny
            s1 = (sigma(w,s)+sigma(w+1,s))/1.0;
            s2 = (sigma(w,s)+sigma(w-1,s))/1.0;
            s3 = (sigma(w,s)+sigma(w,s-1))/2.0;

            G(n, n) = -(s1 + s2 + sym);
            G(n, nxp) = s1;
```

```
            G(n, nxm) = s2;
            G(n, nym) = sym;
        else
            s2 = (sigma(w,s)+sigma(w-1,s))/2.0;
            s1 = (sigma(w,s)+sigma(w+1,s))/2.0;
            sym = (sigma(w,s)+sigma(w,s-1))/2.0;
            s3 = (sigma(w,s)+sigma(w,s+1))/2.0;

            G(n, n) = -(s2+s1+sym+s3);
            G(n, nxp) = s1;
            G(n, nxm) = s2;
            G(n, nyp) = s3;
            G(n, nym) = sym;
        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);
    end
end

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



%My old plots
%Calculate and plot the EX electrix field
figure(4)
[ex,ey]=gradient(voltagePlot1);
quiver(ex.', ey.');

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')




xForce = ex.'*C.q_0;
yForce = ey.'*C.q_0;

xAcc = xForce/(C.m_0*.26);
yAcc = yForce/(C.m_0*.26);

currentConsentration = 1e15;
currentConsentration2 = currentConsentration/1e-4;


Vth = sqrt((C.kb*lastTemperature)/(C.m_0*0.26));
time = 500;
dt = 15e-15;
lambda = Vth*Tau;
fprintf('The thermal velocity is equal to %d\n',Vth)
fprintf('The mean free path is equal to %d\n',lambda)


figure('Name','Electron Paths')

Px = rand(1, TotalElectrons)*200e-9;       %Generate the random x
 location
Py = rand(1, TotalElectrons)*100e-9;       %generate the random y
 location

RandAng = rand(1, TotalElectrons)*2*pi;
Vx = Vth*sin(RandAng);
Vy = Vth*cos(RandAng);

hold on

for loops=1:time
    NewPx = Vx*dt+Px;
    NewPy = Vy*dt+Py;

    Vx = Vx + (dt*xAcc);
    Vx = Vx + (dt*yAcc);

    ix = NewPx>length;
    NewPx(ix) = NewPx(ix)-length;
    Px(ix) = Px(ix)-length;

    ix = NewPx<0;
    NewPx(ix) = NewPx(ix)+length;
    Px(ix) = Px(ix)+length;
```

```
ix = NewPy<0;
Vy(ix) = -Vy(ix);

ix = NewPy>height;
Vy(ix) = -Vy(ix);

subplot (2,3,1)
plot([Px(1) NewPx(1)], [Py(1) NewPy(1)], 'b')
plot([Px(2) NewPx(2)], [Py(2) NewPy(2)], 'g')
plot([Px(3) NewPx(3)], [Py(3) NewPy(3)], 'r')
plot([Px(4) NewPx(4)], [Py(4) NewPy(4)], 'c')
plot([Px(5) NewPx(5)], [Py(5) NewPy(5)], 'm')
plot([Px(6) NewPx(6)], [Py(6) NewPy(6)], 'y')
hold on
title('Electron Paths');
xlim([0 200e-9]);
ylim ([0 100e-9]);
pause(0.000001)

Px=NewPx;
Py=NewPy;

% Drift Velocity to Temperature
V(1, :) = sqrt(Vx(1, :).^2 + Vy(1, :).^2);
V_mean = mean(V.^2);
temperature = V_mean*(C.m_0*.26)/C.kb;

% Plot Temperature Lines
subplot(2,3,2)
plot([lastLoop loops], [lastTemperature temperature], 'r');
title('Temperature');
xlabel('Time Step'); ylabel('Temperature (K)');
xlim([1 time]);
hold on;

subplot(2,3,5)
hist3([Px', Py'], 'CDataMode', 'auto', 'FaceColor', 'interp')
colormap('default');
colorbar;
xlabel('Length'); ylabel('Height');
title('Electron Density');
view(2);


%Map the temps
for y = 1:10
    ymax = y*10;
    ymin = ymax-10;
    for x = 1:10
        xmax = x*20;
        xmin = xmax-20;
        side1 = Px > (xmin*1e-9);
        side2 = Px < (xmax*1e-9);
```

```
            side3 = Py > (ymin*1e-9);
            side4 = Py < (ymax*1e-9);
            between1 = bitand(side1, side2);
            between2 = bitand(side3, side4);
            hit = bitand(between1, between2);
            velocity = sqrt((Vx(hit).^2) + (Vy(hit).^2));
            v_mean = mean(velocity.*velocity);
            temperature_value = (((v_mean)*(C.m_0*.26))/(C.kb));
            temperatures(x, y) = temperature_value;
        end
    end


    %Distribut the temps
    subplot(2,3,4)
    surf(transpose(temperatures));
    xlabel('Length');
    ylabel('Height');
    title('Temperature Distribution');
    colorbar;
    view(2);




     % Electron Drift Current Density
    drift = mean(Vx);
    density = C.q_0*currentConsentration*drift;

    % Plotting Current Density
    subplot(2,3,3)
    plot([lastLoop loops], [lastDensity density], 'r');
    title('Current Density');
    xlabel('Time Step'); ylabel('Current Density (A/cm2)');
    xlim([1 time]);
    hold on;

    pause(0.01);

    % Update electron coordinates

    lastLoop = loops;
    lastTemperature = temperature;
    lastDensity = density;
end

The thermal velocity is equal to 1.322426e+05
The mean free path is equal to 2.644852e-08

Matrix dimensions must agree.

Error in a4700Assignment3_part3 (line 199)
```

```
    Vx = Vx + (dt*xAcc);
```

Published with MATLAB® R2018b