Table of Contents

Assignment 3 1
Part 1 1
Part 2 7
Part 3 11

Assignment 3

```
%By: Chantel Lepage, 100999893
```

Part 1

```
k=1.38e-23;
T=300;
qe = -1.602e-19;
m0 = 9.109e-31;
m=0.26*m0;
t=0.2e-12;
Vx = 0.1;
Vy = 0;
density = 1e15*100^2;
L=200e-9;
W=100e-9;
plotPop=10;
popNum = 3e4;

Vth=sqrt((2*k*T)/m);
tStep=W/Vth/100;
iterations = 200;

pScat = 1 - exp(-tStep/0.2e-12);
vPDF = makedist('Normal', 'mu', 0, 'sigma', sqrt(k*T/m));
MFP = Vth*0.2e-12;


Ex = Vx/L
Ey = Vy/W

Fx = qe*Ex
Fy = qe*Ey

dVx = Fx*tStep/m;
dVy = Fy*tStep/m;
dVx = dVx.*ones(popNum,1);
dVy = dVy.*ones(popNum,1);

state = zeros(popNum,4);
traj=zeros(iterations, plotPop*2);
temp=zeros(iterations, 1);
J = zeros(iterations,2);
```

```
Ex =

   5.0000e+05


Ey =

     0


Fx =

  -8.0100e-14


Fy =

     0
```

The relationship between electron drift current density and average carrier velocity is derived as follows. Let v_x and v_y be the velocity components in x and y for each of the N particles in the simulation. The the average carrier velocties are $\bar{v_x} = 1/N \sum v_x$ and $\bar{v_y} = 1/N \sum v_y$. The electron concentration is $\rho = 10^{15}$ cm^2 and the charge is q. The electron drift current density components are

$$J_x = (q\rho)\left(\frac{1}{N}\right)\sum_{n=1}^{N} v_{x,n} = \frac{q\rho}{N}\sum_{n=1}^{N} v_{x,n}$$

$$J_y = (q\rho)\left(\frac{1}{N}\right)\sum_{n=1}^{N} v_{y,n} = \frac{q\rho}{N}\sum_{n=1}^{N} v_{y,n}$$

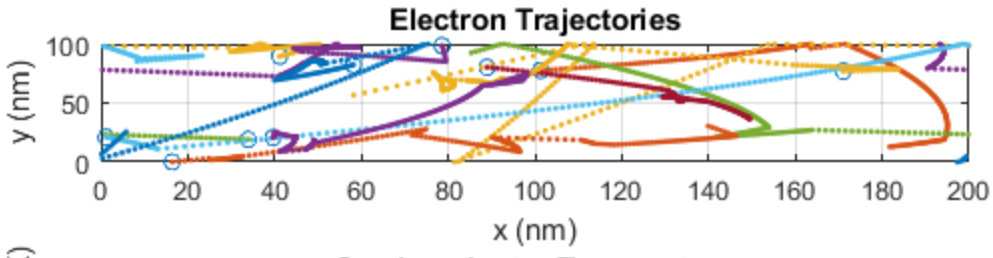
These equations are used to plot the current density over time.

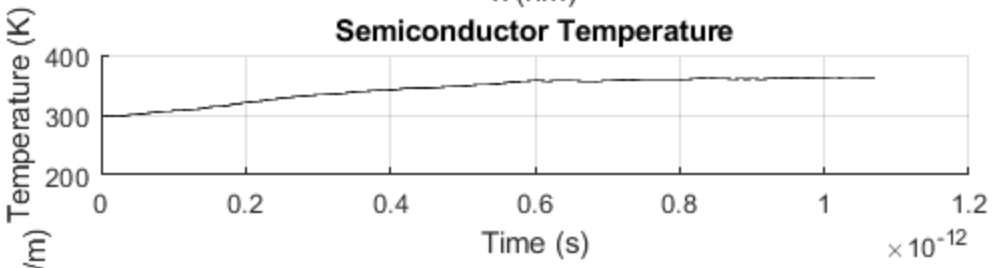
```
top_specular = 0;
bottom_specular = 0;

for i=1:popNum
    theta=rand*2*pi;
    state(i,:)= [L*rand W*rand random(vPDF) random(vPDF)];
end

figure(1);
subplot(3,1,1);
plot([],[]);
axis([0 L/1e-9 0 W/1e-9]);
title(sprintf('Trajectories of Electrons'));
xlabel('x (nm)');
ylabel('y (nm)');

figure(1);
subplot(3,1,2);
```

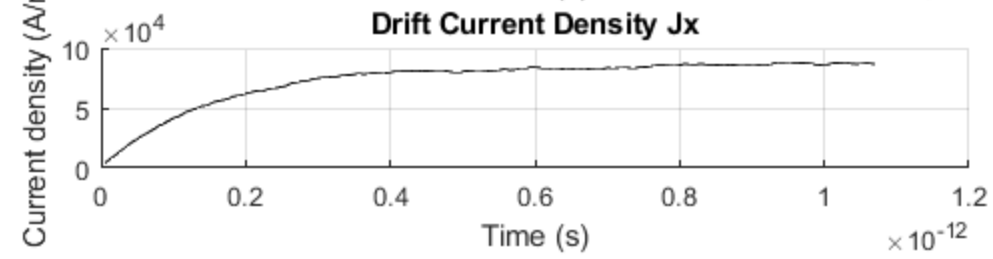
```
tPlot = animatedline;
title('Semiconductor Temperature');
xlabel('Time (s)');
ylabel('Temperature (K)');
grid on;

figure(1);
subplot(3,1,3);
currentPlot = animatedline ;
title('Drift Current Density Jx');
xlabel('Time (s)');
ylabel('Current density (A/m)');
grid on;


for i = 1:iterations

    state(:,3) = state(:,3) + dVx;
    state(:,4) = state(:,4) + dVy;
    state(:,1:2) = state(:,1:2) + tStep.*state(:,3:4);

    j = state(:,1) > L;
    state(j,1) = state(j,1) - L;

    j = state(:,1) < 0;
    state(j,1) = state(j,1) + L;

    j = state(:,2) > W;

    if(top_specular)
        state(j,2) = 2*W - state(j,2);
        state(j,4) = -state(j,4);
    else
        state(j,2) = W;
        v = sqrt(state(j,3).^2 + state(j,4).^2);
        theta = rand([sum(j),1])*2*pi;
        state(j,3) = v.*cos(theta);
        state(j,4) = -abs(v.*sin(theta));
    end

    j = state(:,2) < 0;

    if(bottom_specular)
        state(j,2) = -state(j,2);
        state(j,4) = -state(j,4);
    else
        state(j,2) = 0;
        v = sqrt(state(j,3).^2 + state(j,4).^2);
        theta = rand([sum(j),1])*2*pi;
        state(j,3) = v.*cos(theta);
        state(j,4) = abs(v.*sin(theta));
    end

    j = rand(popNum, 1) < pScat;
```

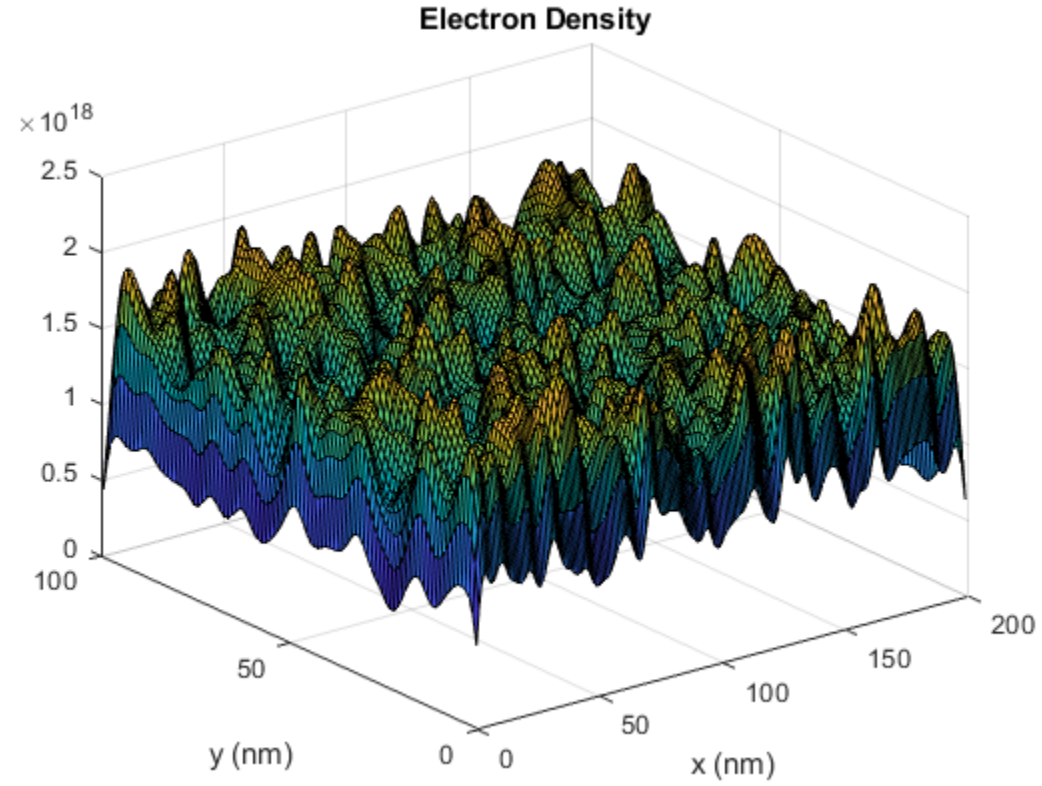
```
        state(j,3:4) = random(vPDF, [sum(j),2]);


        temp(i) = (sum(state(:,3).^2) + sum(state(:,4).^2))*m/k/2/popNum;


        for j=1:plotPop
            traj(i, (2*j):(2*j+1)) = state(j, 1:2);
        end

        J(i, 1) = qe.*density.*mean(state(:,3));
        J(i, 2) = qe.*density.*mean(state(:,4));

        addpoints(tPlot, tStep.*i, temp(i));
        addpoints(currentPlot, tStep.*i, J(i,1));


        if  mod(i,5) == 0
            figure(1);
            subplot(3,1,1);
            hold off;
            plot(state(1:plotPop,1)./1e-9, state(1:plotPop,2)./1e-9, 'o');
            axis([0 L/1e-9 0 W/1e-9]);
            hold on;
            title(sprintf('Trajectories of Electrons'));
            xlabel('x (nm)');
            ylabel('y (nm)');
            pause(0.05);
        end
    end


    figure(1);
    subplot(3,1,1);
    title(sprintf('Electron Trajectories'));
    xlabel('x (nm)');
    ylabel('y (nm)');
    axis([0 L/1e-9 0 W/1e-9]);
    grid on;
    hold on;
    for i=1:plotPop
        plot(traj(:,i*2)./1e-9, traj(:,i*2+1)./1e-9, '.');

    end


    density = hist3(state(:,1:2),[200 100])';

    N = 20;
    sigma = 1.5;
    [x y]=meshgrid(round(-N/2):round(N/2), round(-N/2):round(N/2));
    f=exp(-x.^2/(2*sigma^2)-y.^2/(2*sigma^2));
    f=f./sum(f(:));
    figure(2);
```

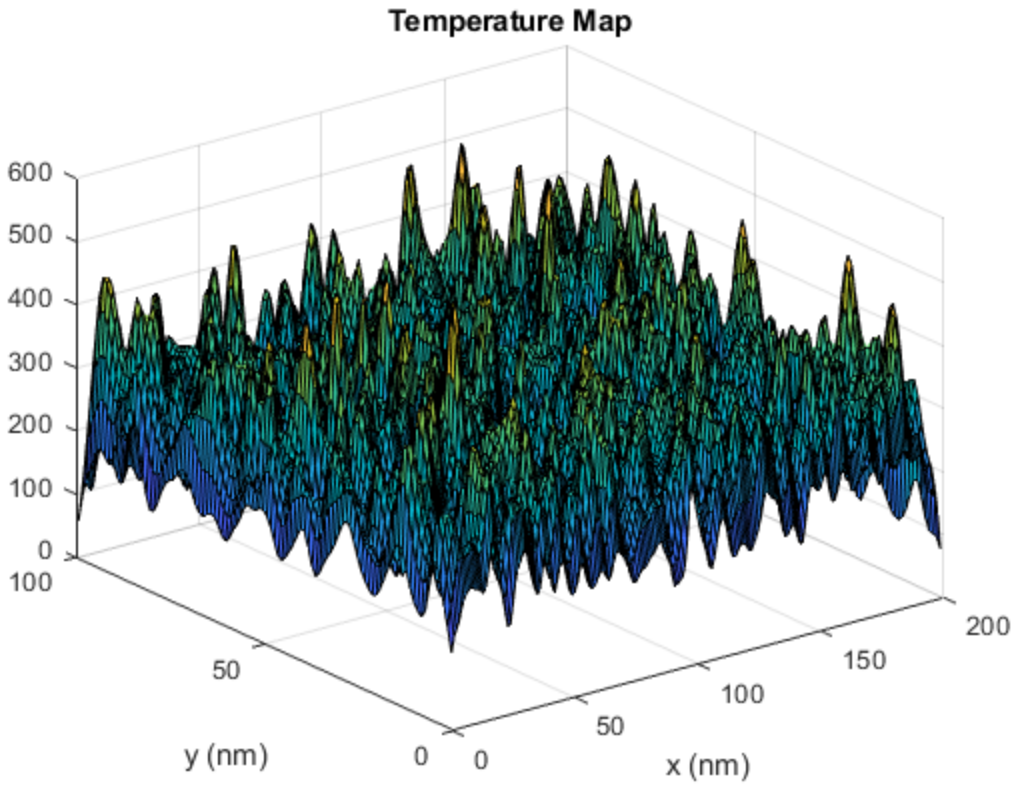
```
density = conv2(density,f,'same');
density = density/(W./size(density,1)*L./size(density,2));
surf(conv2(density,f,'same'));
title('Electron Density');
xlabel('x (nm)');
ylabel('y (nm)');


tempSumX = zeros(ceil(L/1e-9),ceil(W/1e-9));
tempSumY = zeros(ceil(L/1e-9),ceil(W/1e-9));
tempSum = zeros(ceil(L/1e-9),ceil(W/1e-9));


for i=1:popNum

    x = floor(state(i,1)/1e-9);
    y = floor(state(i,2)/1e-9);
    if(x==0)
        x = 1;
    end
    if(y==0)
        y= 1;
    end


    tempSumY(x,y) = tempSumY(x,y) + state(i,3)^2;
    tempSumX(x,y) = tempSumX(x,y) + state(i,4)^2;
    tempSum(x,y) = tempSum(x,y) + 1;
end



temp = (tempSumX + tempSumY).*m./k./2./tempSum;
temp(isnan(temp)) = 0;
temp = temp';



N = 20;
sigma = 1.5;
[x y]=meshgrid(round(-N/2):round(N/2), round(-N/2):round(N/2));
f=exp(-x.^2/(2*sigma^2)-y.^2/(2*sigma^2));
f=f./sum(f(:));
figure(3);
surf(conv2(temp,f,'same'));
title('Temperature Map');
xlabel('x (nm)');
ylabel('y (nm)');
```


Electron Trajectories
y (nm)
x (nm)


Semiconductor Temperature
Temperature (K)
Time (s)
×10^-12


Drift Current Density Jx
Current density (A/m)
×10^4
Time (s)
×10^-12


Electron Density
×10^18
y (nm)
x (nm)

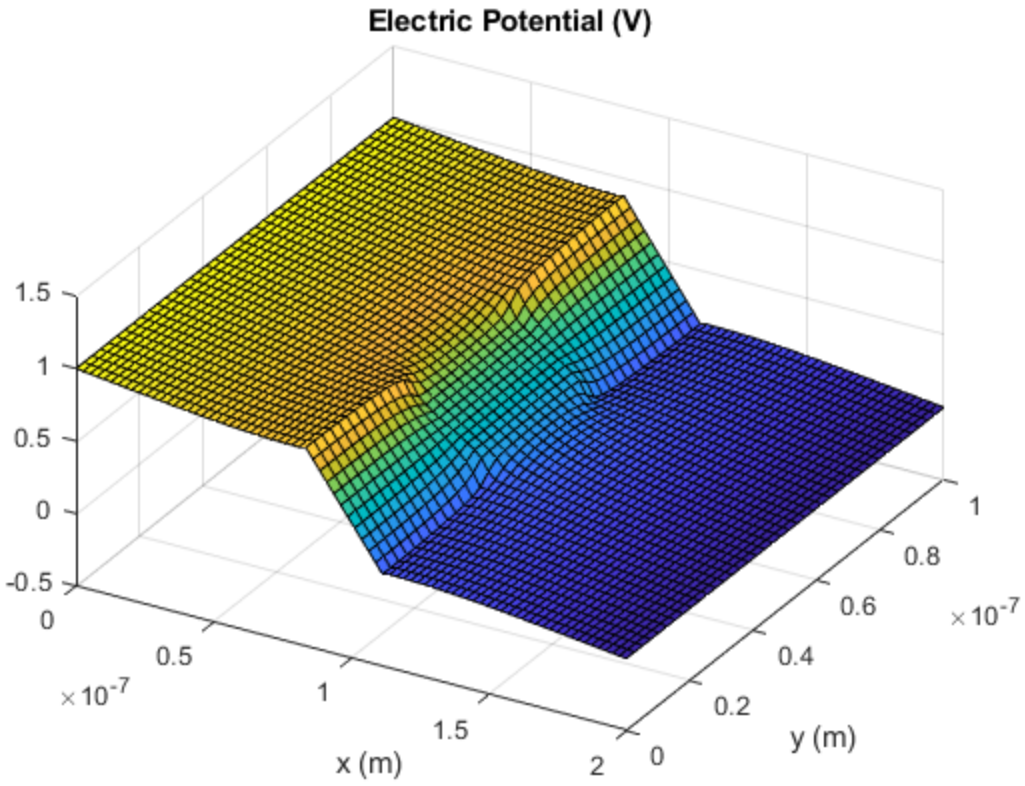


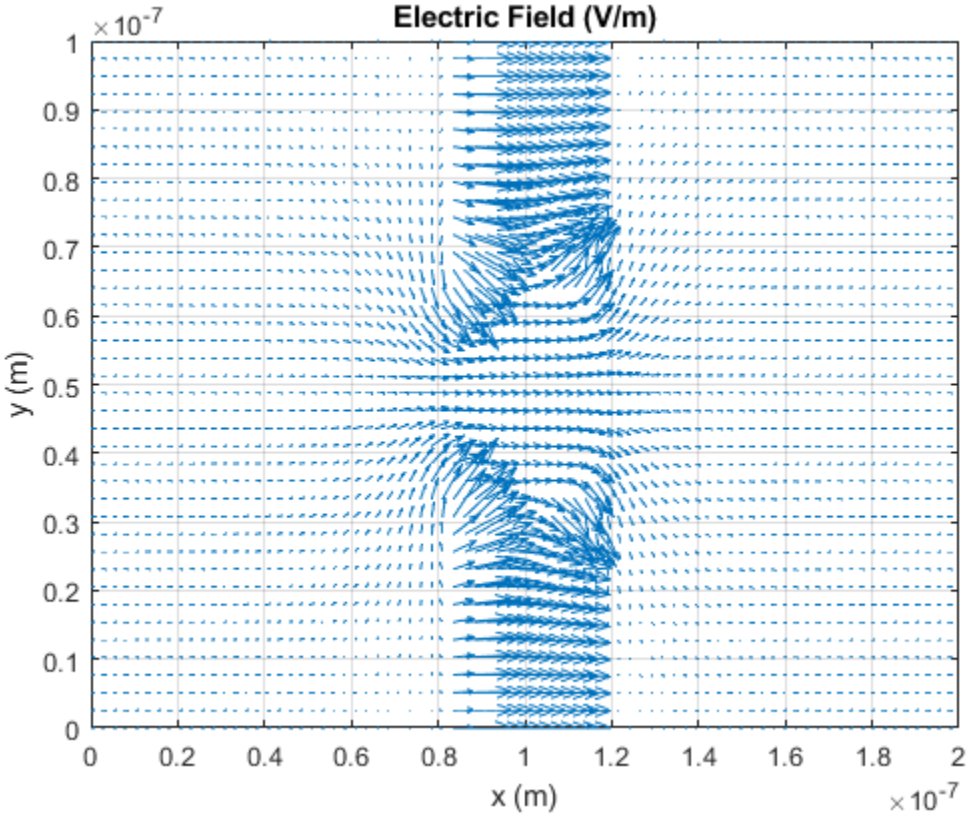
Part 2

```
W = 1;
L = 2;
scale = 100e-9;
V0 = 1;

dx = 0.025;
dy = 0.025;
nx = ceil(L/dx);
ny = ceil(W/dy);
dx = L/nx;
dy = W/ny;

Lb = 0.4;
Wb = 0.4;
sigma1 = 1;
sigma2 = 1e-2;

C = sigma1.*ones(ny,nx);
cSubtract = zeros(ny,nx);

for x=1:nx
    for y=1:ny
        xx = x*dx;
```

```
        yy = y*dy;

        if(xx <= (L+Lb)/2 && xx >= (L-Lb)/2 && (yy >= W-Wb || yy <=
 Wb))
            cSubtract(y,x) = sigma1-sigma2;
        end
    end
end

cSubtract = imgaussfilt(cSubtract, 1);
C = C - cSubtract;



G = zeros(nx*ny,nx*ny);
V = zeros(nx*ny,1);

dx2 = 1./(dx.^2);
dy2 = 1./(dy.^2);

for x=2:(nx-1)
    for y=2:(ny-1)
        index = mapCoordinate(x,y,nx);

        G(index,index) = -2.*C(y,x).*(dx2 + dy2);
        G(index, mapCoordinate(x+1,y,nx)) = dx2.*(0.25.*(C(y,x+1) -
 C(y,x-1)) + C(y,x));
        G(index, mapCoordinate(x-1,y,nx)) = dx2.*(-0.25.*(C(y,x+1) -
 C(y,x-1)) + C(y,x));

        G(index, mapCoordinate(x,y+1,nx)) = dy2.*(0.25.*(C(y+1,x) -
 C(y-1,x)) + C(y,x));
        G(index, mapCoordinate(x,y-1,nx)) = dy2.*(-0.25.*(C(y+1,x) -
 C(y-1,x)) + C(y,x));
    end
end

for x=2:(nx-1)
    index = mapCoordinate(x,1,nx);
    G(index,index) = 1;
    G(index,mapCoordinate(x,2,nx)) = -1;
    V(index) = 0;

    index = mapCoordinate(x,ny,nx);
    G(index,index) = 1;
    G(index,mapCoordinate(x,ny-1,nx)) = -1;
    V(index) = 0;
end

for y=1:ny
    index = mapCoordinate(1,y,nx);
    G(index,index) = 1;
    V(index) = V0;
```

```
    index = mapCoordinate(nx,y,nx);
    G(index,index) = 1;
    V(index) = 0;
end


V = G\V;
V = reshape(V,[],ny)';

figure(4);
surf(linspace(0,L.*scale,nx),linspace(0,W.*scale,ny),V);
view(30,45);
xlabel('x (m)');
ylabel('y (m)');
title('Electric Potential (V)');
grid on;

figure(5);
[Ex,Ey] = gradient(V,dx.*scale,dy.*scale);
Ex = -1.*Ex;
Ey = -1.*Ey;
quiver(linspace(0,L.*scale,nx),linspace(0,W.*scale,ny),Ex,Ey,4);
xlabel('x (m)');
ylabel('y (m)');
title('Electric Field (V/m)');
axis([0 L.*scale 0 W.*scale]);
grid on;
```


Electric Potential (V)
1.5
1
0.5
0
-0.5
0
0.5
1
1.5
2
× 10^-7
x (m)
0
0.2
0.4
0.6
0.8
1
× 10^-7
y (m)


Electric Field (V/m)
× 10^-7
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
y (m)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
x (m)
× 10^-7

Part 3

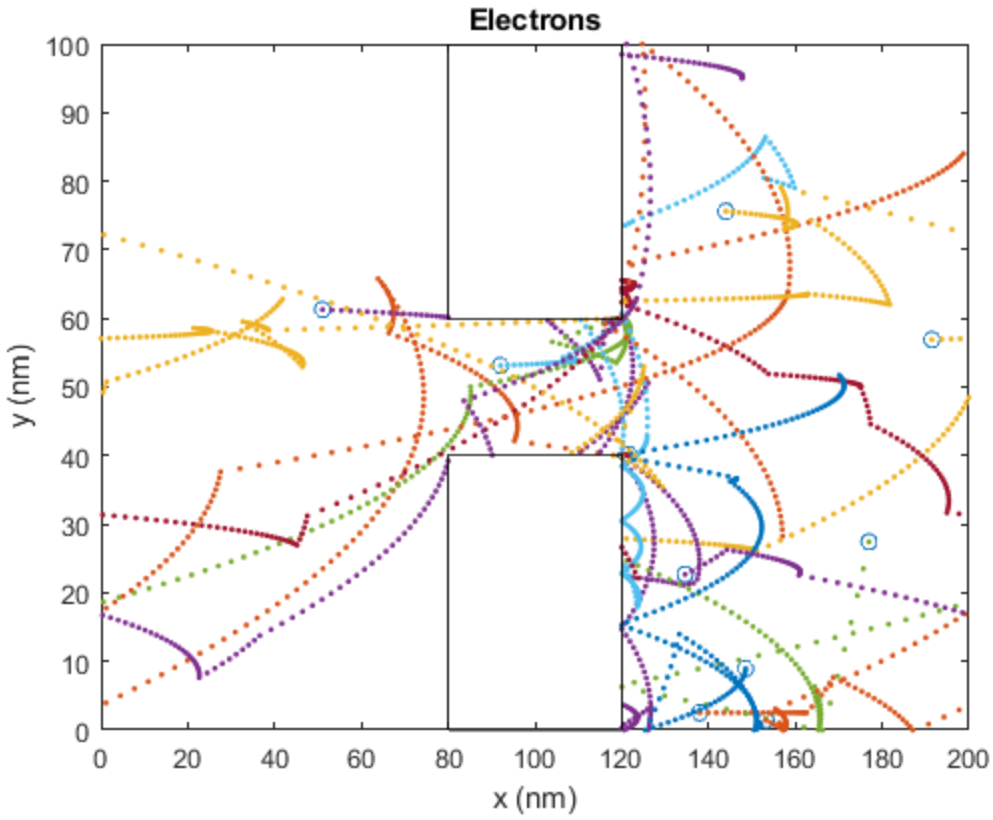
```
k=1.38e-23;
T=300;
qe = -1.602e-19;
m0 = 9.109e-31;
m=0.26*m0;
t=0.2e-12;
Vx = 0.8;
Vy = 0;
density = 1e15*100^2;
L=200e-9;
W=100e-9;
plotPop=10;
popNum = 3e4;

Vth=sqrt((2*k*T)/m);
tStep=W/Vth/100;
iterations = 200;

pScat = 1 - exp(-tStep/0.2e-12);
vPDF = makedist('Normal', 'mu', 0, 'sigma', sqrt(k*T/m));
MFP = Vth*0.2e-12;


Ex = Vx/L;
Ey = Vy/W;

Fx = qe*Ex;
Fy = qe*Ey;

dVx = Fx*tStep/m;
dVy = Fy*tStep/m;
dVx = dVx.*ones(popNum,1);
dVy = dVy.*ones(popNum,1);

state = zeros(popNum,4);
traj=zeros(iterations, plotPop*2);
temp=zeros(iterations, 1);
J = zeros(iterations,2);

top_specular = 0;
bottom_specular = 0;


boxes = 1e-9.*[80 120 0 40; 80 120 60 100];
boxes_specular = [0 1];


for i = 1:popNum
    theta = rand*2*pi;
    state(i,:) = [L*rand W*rand random(vPDF) random(vPDF)];
```

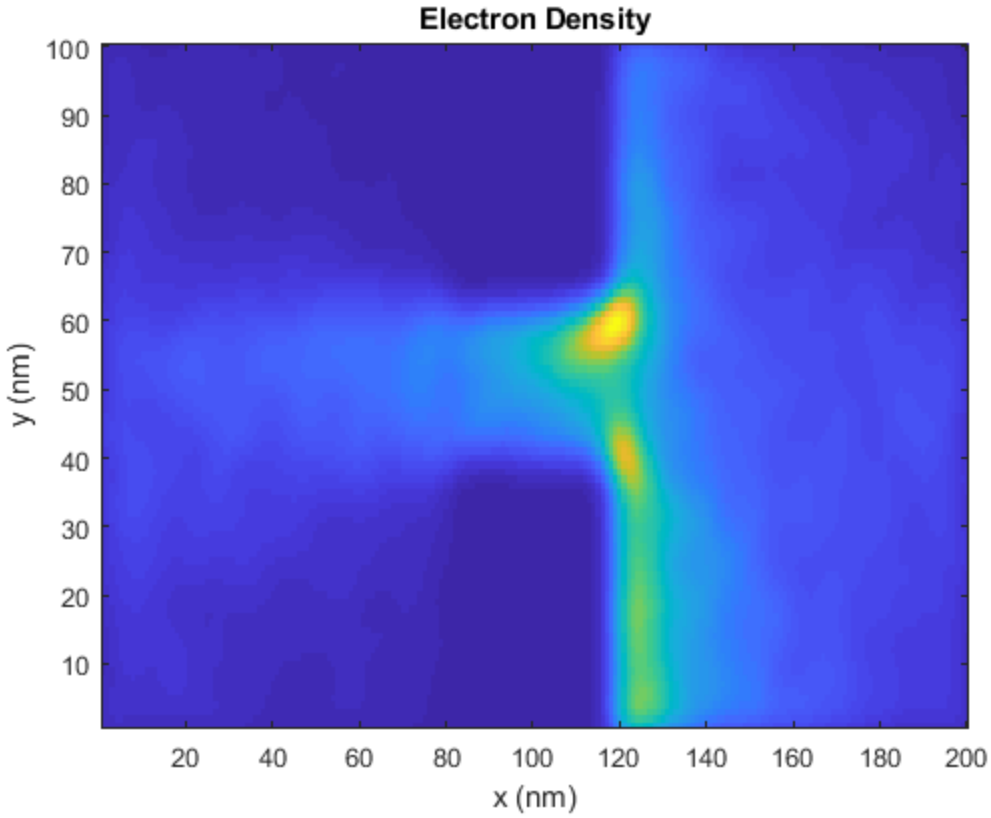
```
    if (state(i,2)>60e-9 &(state(i,1)>80e-9 & state(i,1)<120e-9))  |
 (state(i,2)< 40e-9 &(state(i,1)>80e-9 & state(i,1)<120e-9))
        state(i,1:2) = [L*rand W*rand];
    end
end


for i = 1:iterations
    state(:,3) = state(:,3) + dVx;
    state(:,4) = state(:,4) + dVy;
    state(:,1:2) = state(:,1:2) + tStep.*state(:,3:4);

    j = state(:,1) > L;
    state(j,1) = state(j,1) - L;

    j = state(:,1) < 0;
    state(j,1) = state(j,1) + L;

    j = state(:,2) > W;

    if(top_specular)
        state(j,2) = 2*W - state(j,2);
        state(j,4) = -state(j,4);
    else
        state(j,2) = W;
        v = sqrt(state(j,3).^2 + state(j,4).^2);
        theta = rand([sum(j),1])*2*pi;
        state(j,3) = v.*cos(theta);
        state(j,4) = -abs(v.*sin(theta));
    end

    j = state(:,2) < 0;

    if(bottom_specular)
        state(j,2) = -state(j,2);
        state(j,4) = -state(j,4);
    else
        state(j,2) = 0;
        v = sqrt(state(j,3).^2 + state(j,4).^2);
        theta = rand([sum(j),1])*2*pi;
        state(j,3) = v.*cos(theta);
        state(j,4) = abs(v.*sin(theta));
    end

    for j=1:popNum
        if (state(j,2)>60e-9 &(state(j,1)>80e-9 & state(j,1)<120e-9))
            boxNum = 1;
        elseif (state(j,2)< 40e-9 &(state(j,1)>80e-9 &
 state(j,1)<120e-9))
                boxNum = 2;
        else
            boxNum = 0;
        end
        while(boxNum ~= 0)
```

```
            xDist = 0;
            newX = 0;
            if(state(j,3) > 0)
                xDist = state(j,1) - boxes(boxNum,1);
                newX = boxes(boxNum,1);
            else
                xDist = boxes(boxNum,2) - state(j,1);
                newX = boxes(boxNum,2);
            end

            y_dist = 0;
            new_y = 0;
            if(state(j,4) > 0)
                y_dist = state(j,2) - boxes(boxNum, 3);
                new_y = boxes(boxNum, 3);
            else
                y_dist = boxes(boxNum, 4) - state(j,2);
                new_y = boxes(boxNum, 4);
            end

            if(xDist < y_dist)
                state(j,1) = newX;
                if(~boxes_specular(boxNum))
                    sgn = -sign(state(j,3));
                    v = sqrt(state(j,3).^2 + state(j,4).^2);
                    theta = rand()*2*pi;
                    state(j,3) = sgn.*abs(v.*cos(theta));
                    state(j,4) = v.*sin(theta);
                else
                    state(j,3) = -state(j,3);
                end
            else
                state(j,2) = new_y;
                if(~boxes_specular(boxNum))
                    sgn = -sign(state(j,4));
                    v = sqrt(state(j,3).^2 + state(j,4).^2);
                    theta = rand()*2*pi;
                    state(j,3) = v.*cos(theta);
                    state(j,4) = sgn.*abs(v.*sin(theta));
                else
                    state(j,4) = -state(j,4);
                end
            end
             boxNum = 0;
        end
    end


    j = rand(popNum, 1) < pScat;
    state(j,3:4) = random(vPDF, [sum(j),2]);


    temp(i) = (sum(state(:,3).^2) + sum(state(:,4).^2))*m/k/2/popNum;
```

```
    for j=1:plotPop
        traj(i, (2*j):(2*j+1)) = state(j, 1:2);
    end

    J(i, 1) = qe.*density.*mean(state(:,3));
    J(i, 2) = qe.*density.*mean(state(:,4));


    if  mod(i,5) == 0
        figure(6);
        hold off;
        plot(state(1:plotPop,1)./1e-9, state(1:plotPop,2)./1e-9, 'o');
        hold on;

        for j=1:size(boxes,1)
           plot([boxes(j, 1) boxes(j, 1) boxes(j, 2) boxes(j, 2)
 boxes(j, 1)]./1e-9,[boxes(j, 3) boxes(j, 4) boxes(j, 4) boxes(j, 3)
 boxes(j, 3)]./1e-9, 'k-');
        end

        axis([0 L/1e-9 0 W/1e-9]);
        title(sprintf('Electrons',plotPop, popNum));
        xlabel('x (nm)');
        ylabel('y (nm)');
        pause(0.05);
    end
end


figure(6);
title(sprintf('Electrons',plotPop, popNum));
xlabel('x (nm)');
ylabel('y (nm)');
axis([0 L/1e-9 0 W/1e-9]);
hold on;
for i=1:plotPop
    plot(traj(:,i*2)./1e-9, traj(:,i*2+1)./1e-9, '.');

end


for j=1:size(boxes,1)
   plot([boxes(j, 1) boxes(j, 1) boxes(j, 2) boxes(j, 2) boxes(j,
 1)]./1e-9,...
       [boxes(j, 3) boxes(j, 4) boxes(j, 4) boxes(j, 3) boxes(j,
 3)]./1e-9, 'k-');
end


density = hist3(state(:,1:2),[200 100])';
N = 20;
sigma = 3;
```

```
[x y]=meshgrid(round(-N/2):round(N/2), round(-N/2):round(N/2));
f=exp(-x.^2/(2*sigma^2)-y.^2/(2*sigma^2));
f=f./sum(f(:));
figure(7);
imagesc(conv2(density,f,'same'));
set(gca,'YDir','normal');
title('Electron Density');
xlabel('x (nm)');
ylabel('y (nm)');
```





Both of the density plot shows that most electrons populate on the right edge of the boxes of the bottle-neck because the voltage applied across the field forces the electrons to constantly try to move toward the left side of the region but they cannot pass through the bottle-neck most of time hence creating a higher temperature on the right side.

The next step to improve the simulation is to increase the mesh size of the G-matrix to get more accurate electric field strength and acceleration on each electron and its respective location without rounding its location within the region.

Published with MATLAB® R2017b