Contents

```
% Student Name: Patrobas Adewumi
% Student Number: 100963608
% ELEC 4700: The Physics and Modeling of Advanced Devices and Technologies
% Assignment#1: Monte-Carlo Modeling of Electron Transport
clearvars
clearvars -GLOBAL
close all

global C
global X Y
    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 rest 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 s²

mn = 0.26*C.m_0;                        % effective masss of electron
Temp = 300;                             % In kelvin
runTime = 10000;                        % run time in timesteps
Tmn = 0.2e-12;                          % mean time between collisions
sizeX = 200e-9;                         % normal size of the region in x dir (length)
sizeY = 100e-9;                         % normal size of the region in y dir (width)

Vth = sqrt(2*C.kb*Temp/mn);

size = 1000;
dispSize = 10;
```

Publishing Documents with MATLAB

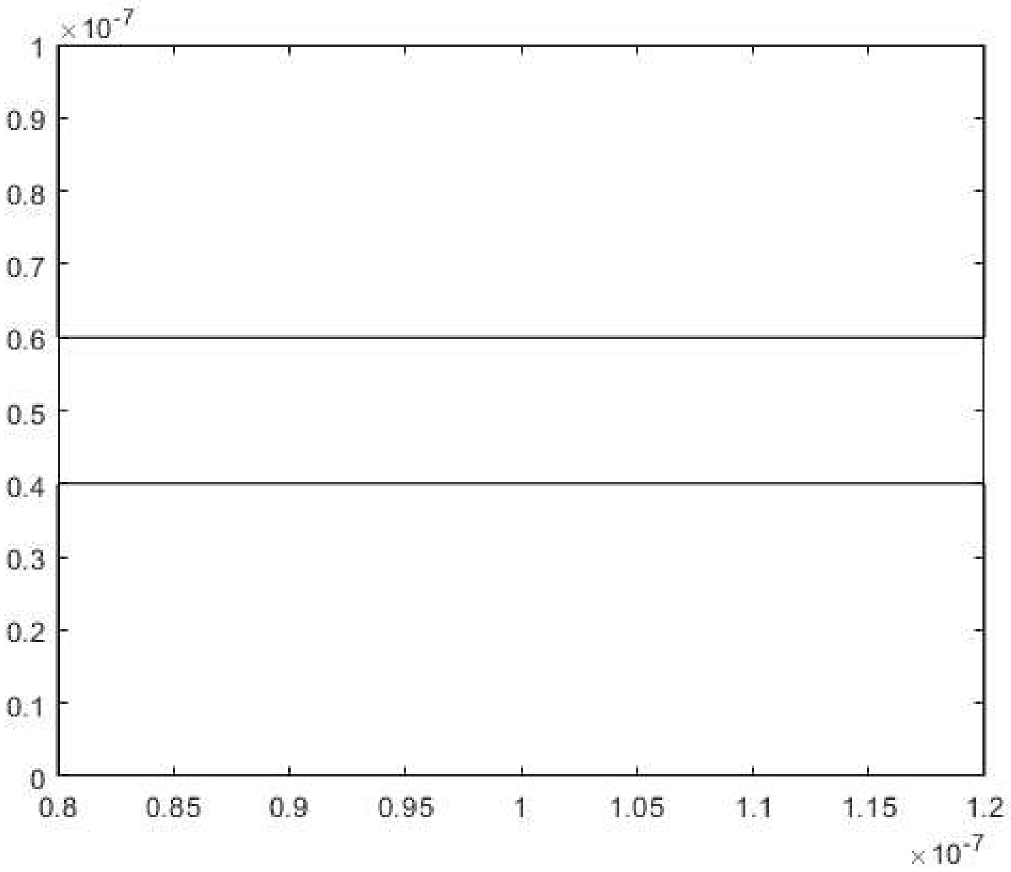
Part 3

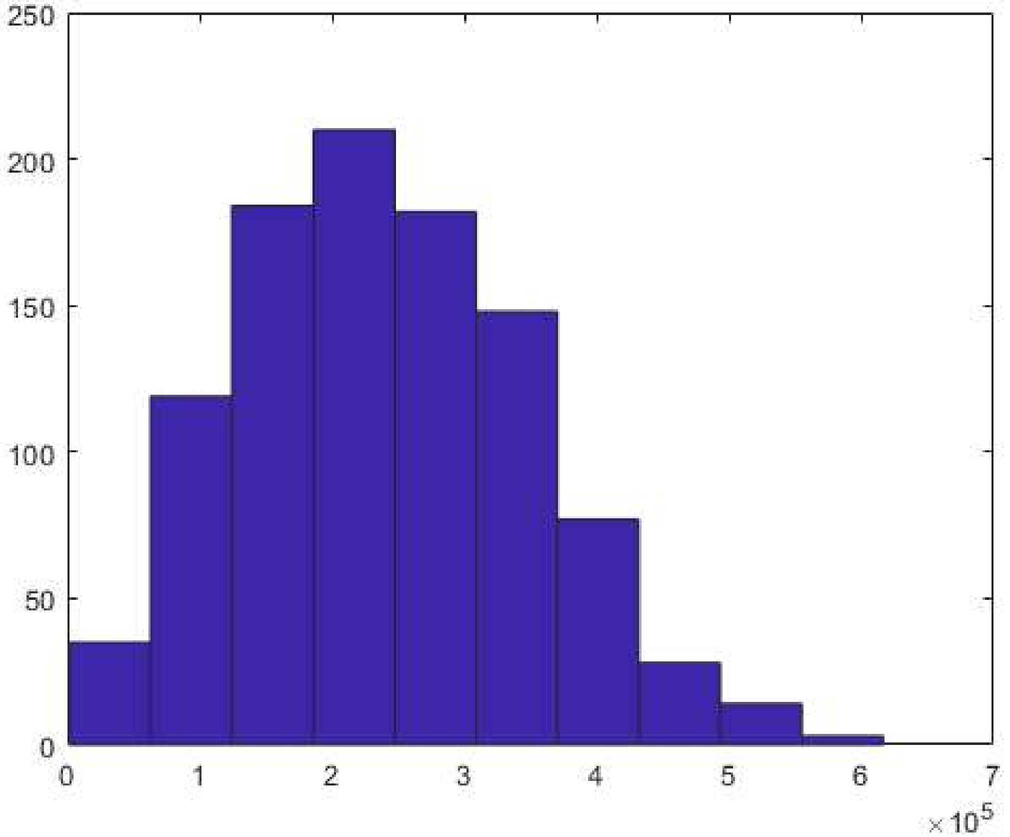
```
%%%%%%%% Assigning each particle a random location in the x?y plane %%%%%%%
%%%%%%%% within the region defined by the extent of the Silicon %%%%%%%%%%%
X = rand(2,size);
Y = rand(2,size);

Pos_X(1,:)= X(1,:)*sizeX;
Pos_Y(1,:)= Y(1,:)*sizeY;

checkXboxLHS = Pos_X > 0.8e-7;
checkXboxRHS = Pos_X < 1.2e-7;
checkXbox = checkXboxLHS & checkXboxRHS;
checkYBoxbot = Pos_Y < 0.4e-7;
```

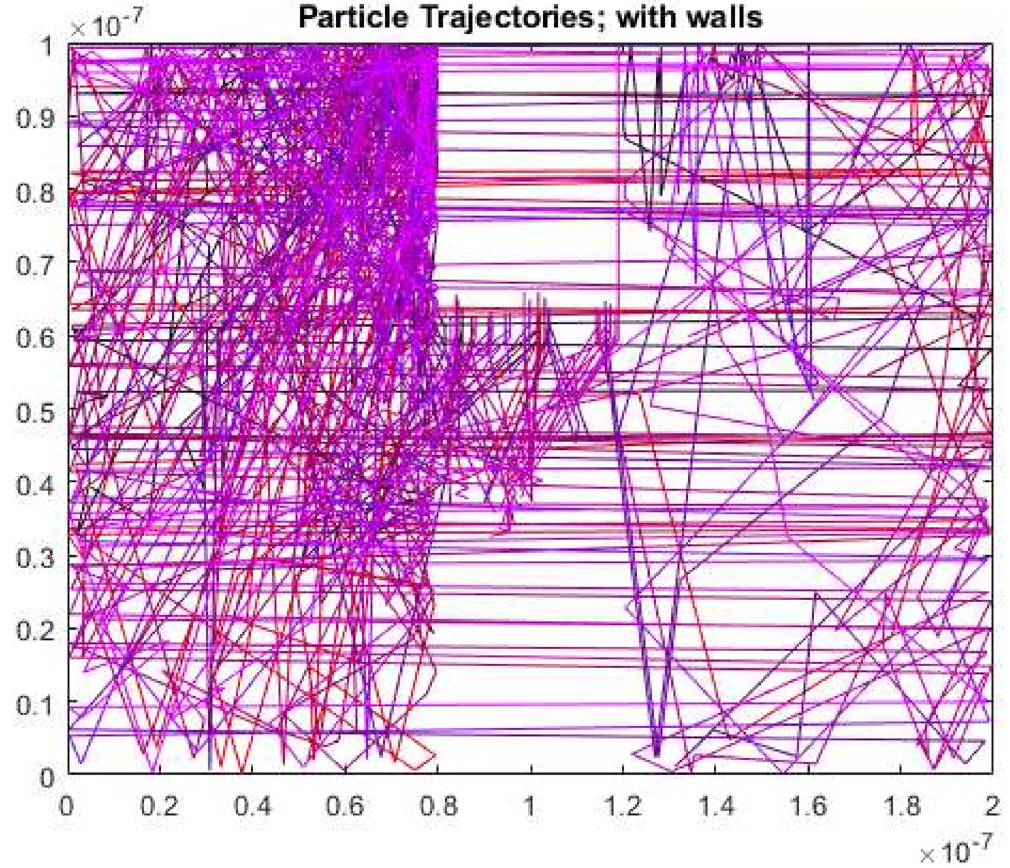
```
checkBoxbot = checkYBoxbot & checkXbox;
checkYBoxtop = Pos_Y > 0.6e-7;
checkBoxtop = checkYBoxtop & checkXbox;
checkboxes = checkBoxtop | checkBoxbot;

while(sum(checkboxes) > 0)

    Pos_X(checkboxes) = rand*sizeX;
    Pos_Y(checkboxes) = rand*sizeY;

    checkXboxLHS = Pos_X > 0.8e-7;
    checkXboxRHS = Pos_X < 1.2e-7;
    checkXbox = checkXboxLHS & checkXboxRHS;
    checkYBoxbot = Pos_Y < 0.4e-7;
    checkBoxbot = checkYBoxbot & checkXbox;
    checkYBoxtop = Pos_Y > 0.6e-7;
    checkBoxtop = checkYBoxtop & checkXbox;

    checkboxes = checkBoxtop | checkBoxbot;
end
colour = rand(1,dispSize);

%%%%%%%% Normal distribution of velocity %%%%%%%%%%%%%%%%
sigma = sqrt(C.kb*Temp/mn);
mu = Vth/sqrt(2);
MB_dist = makedist('Normal',mu,sigma);
Vel_X = zeros(1,size);
Vel_Y = zeros(1,size);
for i=1:1:size
    Vel_X(1,i) = random(MB_dist);
    Vel_Y(1,i) = random(MB_dist);
end

%%%%%%%%%%% Set timestep of function %%%%%%%%%%%%%%%%%%%%%%%%
spacStep = 0.01*sizeX;
dt = spacStep/Vth;
steps = 1000;

%%%%%% Variable change %%%%%%%%%%%%%%%%%%%%%

Vel_X(1,:) = Vel_X(1,:)*dt;
Vel_Y(1,:) = Vel_Y(1,:)*dt;

%%%%%%%%%%%%%%%%%%%% Percent scatter %%%%%%%%%%%%%%%%%%%%%
Pscat = 1 - exp(-(dt/Tmn));

calcTemp = zeros(1,size);

figure(6)
boxplotX = [0.8e-7 0.8e-7 1.2e-7 1.2e-7];
boxplotY = [0 0.4e-7 0.4e-7 0];
plot(boxplotX,boxplotY,'color',[0 0 0]);
hold on
boxplotY = [1e-7 0.6e-7 0.6e-7 1e-7];
plot(boxplotX,boxplotY,'color',[0 0 0]);
```

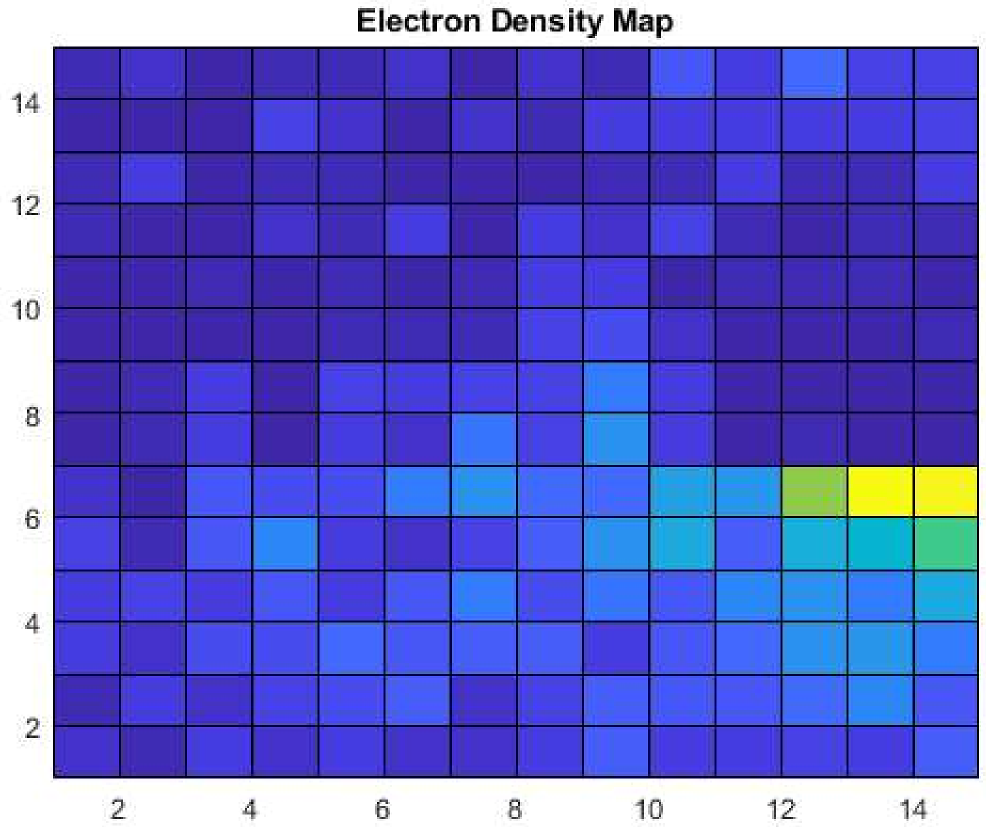
```
for i = 1:1:steps
    scattered = rand(1,size);
    scatterCheck = scattered <= Pscat;
    velocity = random(MB_dist,1,size);
    Vel_X(scatterCheck) = velocity(scatterCheck).*dt;
    velocity = random(MB_dist,1,size);
    Vel_Y(scatterCheck) = velocity(scatterCheck).*dt;


    checkXright = Pos_X + Vel_X > 2e-7;
    Pos_X(checkXright) = Pos_X(checkXright)+Vel_X(checkXright)- sizeX;
    checkXleft = Pos_X +Vel_X<0;
    Pos_X(checkXleft) = Pos_X(checkXleft) +Vel_X(checkXleft)+ sizeX;

    %%%%%%%%%%%% Bottom box %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%% For x reflection %%%%%%%%%%%%%%%%%%%%%%%%%%
    checkXboxLHS = (Pos_X + Vel_X) > 0.8e-7;
    checkXboxRHS = (Pos_X + Vel_X) < 1.2e-7;
    checkXbox = checkXboxLHS & checkXboxRHS;

    checkYBoxbot =(Pos_Y + Vel_Y) < 0.4e-7;

    checkBoxBottom = checkYBoxbot & checkXbox;

    Vel_X(checkBoxBottom) = Vel_X(checkBoxBottom).*(-1);

    %%%%%%%%%%% For y reflection %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    checkXboxLHS = (Pos_X + Vel_X) > 0.8e-7 + spacStep;
    checkXboxRHS = (Pos_X + Vel_X) < 1.2e-7 - spacStep;
    checkXbox = checkXboxLHS & checkXboxRHS;
    checkYabove =Pos_Y < 0.4e-7 - spacStep;
    changeY = checkYabove & checkXbox;

    Vel_Y(changeY) = Vel_Y(changeY).*(-1);

    %%%%%%%%%%%%%% Top box %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%% For x reflection %%%%%%%%%%%%%%%%%
    checkXboxLHS = (Pos_X + Vel_X) > 0.8e-7;
    checkXboxRHS = (Pos_X + Vel_X) < 1.2e-7;
    checkXbox = checkXboxLHS & checkXboxRHS;

    checkYBoxbot =(Pos_Y + Vel_Y) > 0.6e-7;

    checkBoxBottom = checkYBoxbot & checkXbox;

    Vel_X(checkBoxBottom) = Vel_X(checkBoxBottom).*(-1);

    %%%%%$$$$ For y reflection %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    checkXboxLHS = (Pos_X + Vel_X) > 0.8e-7 + spacStep;
    checkXboxRHS = (Pos_X + Vel_X) < 1.2e-7 - spacStep;
    checkXbox = checkXboxLHS & checkXboxRHS;
    checkYabove =Pos_Y > 0.6e-7 + spacStep;
    changeY = checkYabove & checkXbox;

    Vel_Y(changeY) = Vel_Y(changeY).*(-1);

    leftover = ~(checkXright | checkXleft | checkBoxBottom);
    Pos_X(leftover) = Pos_X(leftover) + Vel_X(leftover);

    %%%%%%%%%%%%%%%% Reflect Y boundary %%%%%%%%%%%%%%%%%%%%%%%%%%%%
    checkY = (Pos_Y+Vel_Y > 1e-7 | Pos_Y + Vel_Y < 0);
    Vel_Y(checkY) = Vel_Y(checkY).*(-1);
    Pos_Y(1,:) = Pos_Y(1,:) + Vel_Y(1,:);

    %%%%%%%%%%%%%% Temperature calculations %%%%%%%%%%%%%%%%%%
    Ydt = Vel_Y./dt;
    Xdt = Vel_X./dt;
    Ysum = sum(Ydt);
    Xsum = sum(Xdt);
    Ysquare = Ysum.^2;
    Xsquare = Xsum.^2;
    vel = sqrt(Ysquare + Xsquare)/size;
    calcTemp = (mn*(vel)^2)/(2*C.kb);
    averageTemp = calcTemp;

    mean_vel = sum(velocity)/size;
    calcTemp(1,i) = mn*(mean_vel)^2/(2*C.kb);

    prevX(i,:) = Pos_X(1,:);
    prevY(i,:) = Pos_Y(1,:);
    figure(2)
    hist(sqrt((Vel_X/dt).^2 + (Vel_Y/dt).^2));
end

for j = 1:1:dispSize

    figure (35)
    plot(prevX(:,j),prevY(:,j),'color',[colour(1,j) 0 j/dispSize]);

    title('Particle Trajectories; with walls')

    rectangle('Position',[1 2 5 6])
    axis([0 10 0 10])

    xlim([0 sizeX]);
    ylim([0 sizeY]);
    drawnow
    hold on
end



figure(7)
n = hist3([Pos_X',Pos_Y'],[15,15]);
pcolor(n)
title('Electron Density Map')

%%%%%%%%%%%%%% For temperature areas %%%%%%%%%%%%%%%%%%%%
X_edges = linspace(0,sizeX,10);
Y_edges = linspace(0,sizeY,10);

X_bins = discretize(Pos_X,X_edges);
Y_bins = discretize(Pos_Y,Y_edges);
```

```
binTemp = zeros(10,10);

for k = 1:1:10 % x
    for L = 1:1:10 % y
        logicX = X_bins == k;
        logicY = Y_bins == L;
        logic = logicX & logicY;
        sumX = sum(Vel_X(logic))/dt;
        sumY = sum(Vel_Y(logic))/dt;
        mean_vel = sqrt((sumX)^2 + (sumY)^2);
        binTemp(k,L) = mn*(mean_vel)^2/(2*C.kb);
    end
end

figure(8)
title('Temperature Map')
surf(binTemp)
colorbar;
```

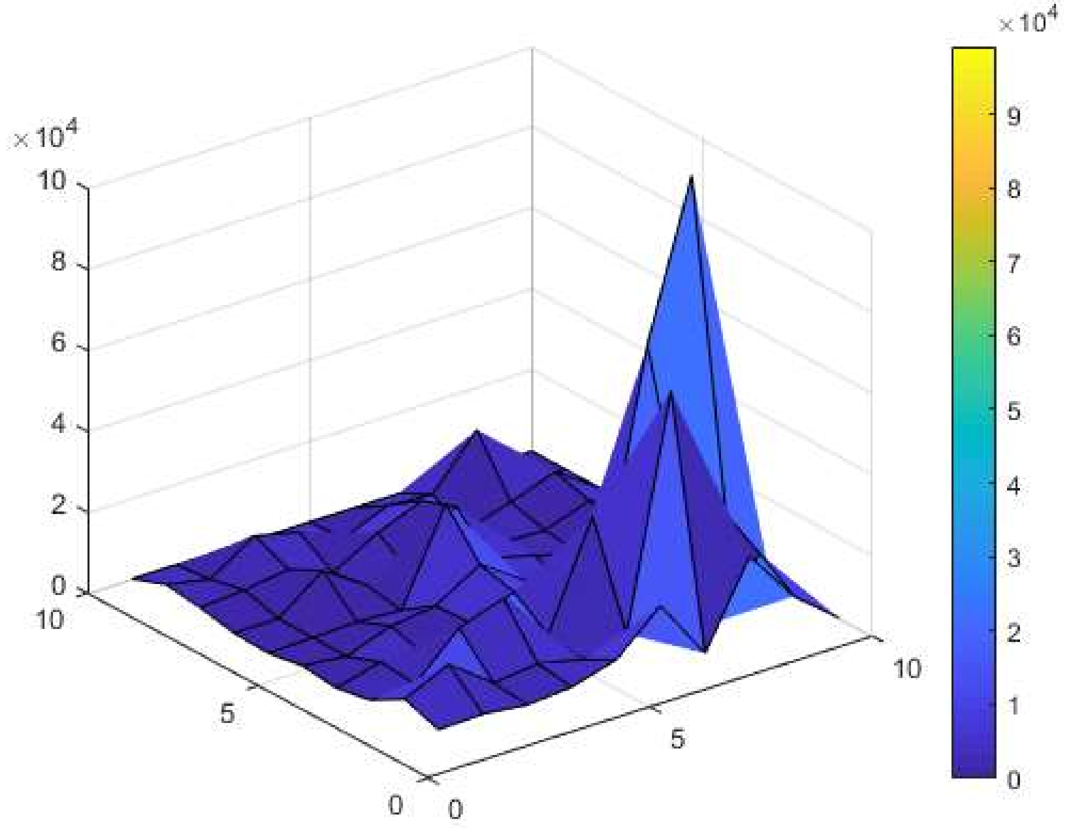



250
200
150
100
50
0
0
1
2
3
4
5
6
7
×10^5


Particle Trajectories; with walls
×10^-7
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
×10^-7



Electron Density Map



Published with MATLAB® R2018a