ELEC 4700 Assignment 3 Part 1

Umeh Ifeanyi 101045786

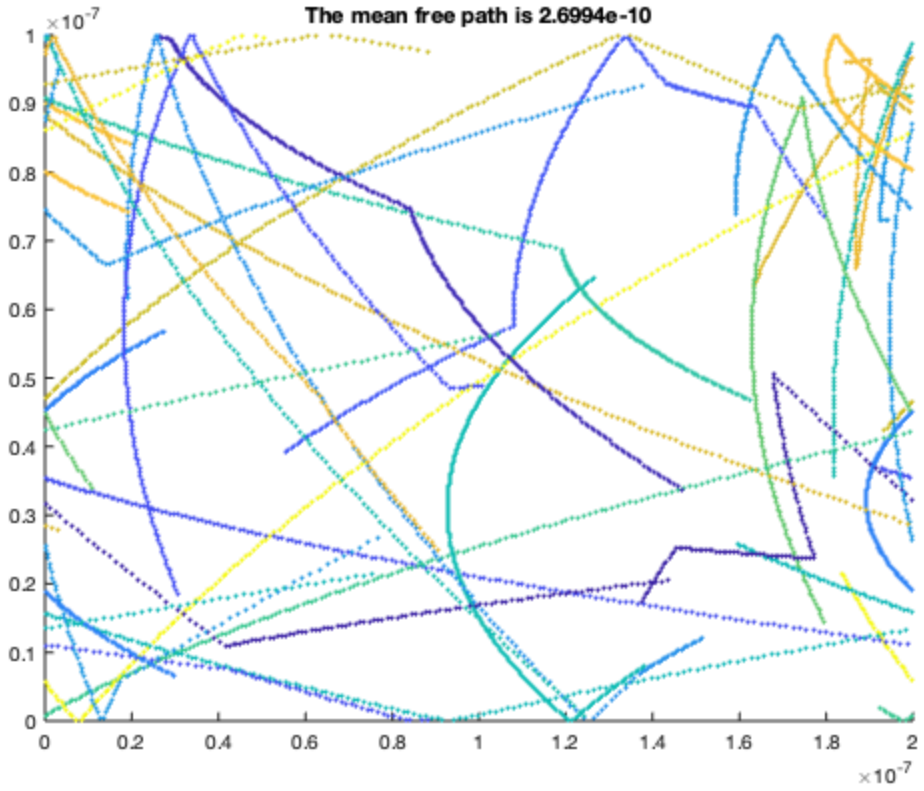
```
% In this part of the assignment, we applied a 0.1 electric field in
 the x
% direction. The motion of the electrons were mapped to view the
 effects
% this field had on the movement. As before, density and temperature
 maps
% were created.

close all
clear
%Constants

q_0 = 1.60217653e-19;             % electron charge
m_0 = 9.10938215e-31;             % electron mass
kB = 1.3806504e-23;               % Boltzmann constant
deltat = 0.2e-12;                 % mean time between collisions
mn = 0.26*m_0;                    % effective mass of electrons



%variables

numofelec = 1000;              %current numbers of electrons t be
 simulated
T = 300;                     %temperature in kelvin

%NEW ADDITION for assignment 3%
Voltage = 0.1;               %the one dimensional voltage applied
 across the
                             %x dimension of the semiconductor (PART A)
%%%%%%%%%%%%%

dt = 1;
%Assign each particle with the fixed velocity given by vth but give
 each one a
%random direction.

vth = sqrt((kB*T)/mn);
%Spatial Boundaries

Length = 200;
Width = 100;


    %I am going to represent the location of each electron using
 vectors

x = randi([0 Length], 1, numofelec)*1e-9;       %initializing x
```

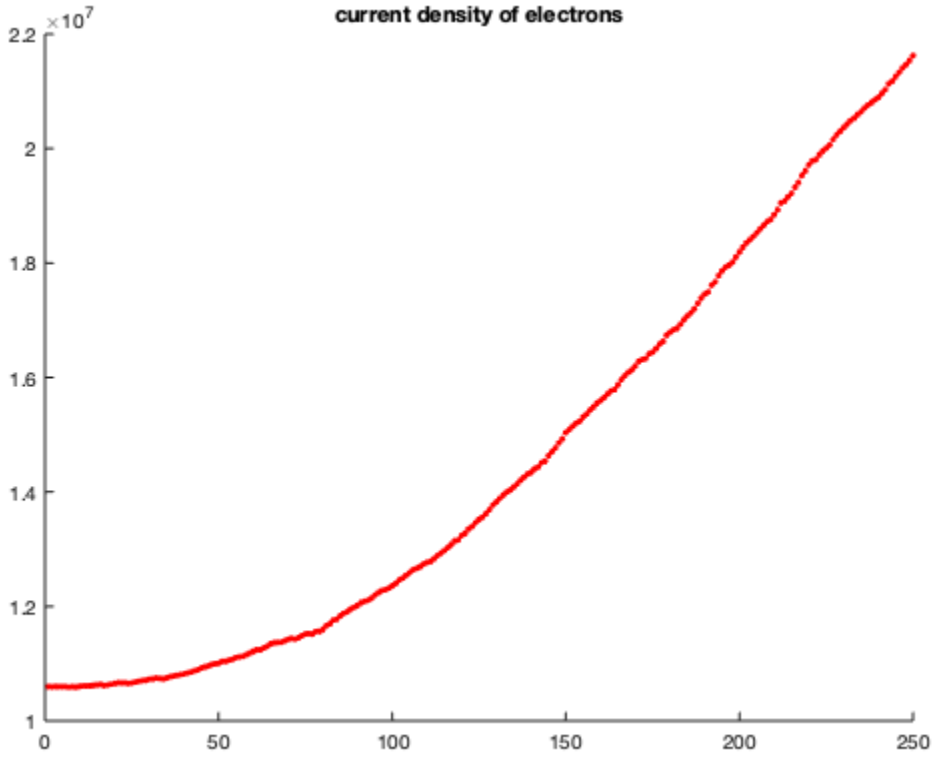
```
y = randi([0 Width], 1, numofelec)*1e-9;        %initializing y

    %top side of lower rectangle
    for it=1:1:numofelec

        %moving spawned electrons outside of rectangles

        if x(1,it) >=(80e-9) && x(1,it) <= (120e-9) && y(1,it)<=
 (40e-9)
                x(1,it) = x(1,it) + randi([45 80], 1,1)*1e-9;
        end

        if x(1,it) >=(80e-9) && x(1,it) <= (120e-9) && y(1,it)>=
 (60e-9)
                x(1,it) = x(1,it) - randi([45 80], 1,1)*1e-9;
        end

    end

    %now we have position vectors for the x and y positions of each
    %electron. Need to create vectors for vy and vx. Remember that
 each
    %electron has a rand angle to start with, but same velocity vth.

angles = randi([0 360], 1, numofelec);
v_x = zeros(1, numofelec);
v_y = zeros(1, numofelec);

v_x = vth*cos(angles);
v_y = vth*sin(angles);

%NEW ADDITION for assignment 3%

elec_x = Voltage/(Length*1e-9);     %value of electric field across x
 (PART A)

fprintf('Part a) The value of the electric field on the electrons is
 %i\n',elec_x);

Force = elec_x*q_0;         %creates a vector containing forces of all
 electrons
                            %(PART B)
fprintf('Part b) The value of force on each electron is %i\n',Force);

a_elec = Force/mn;        %creates a vector containing all
 acceleration of electrons
                            %will be used to modify the plot. stay
 tuned
                            %for more (PART C)
fprintf('Part c) The value of acceletation on each electron is %i
\n',a_elec);
%%%%%%%%%%%%%%
    %scatter
    pscat = 1 - exp(-1e-14/(1e-11*0.2));
```

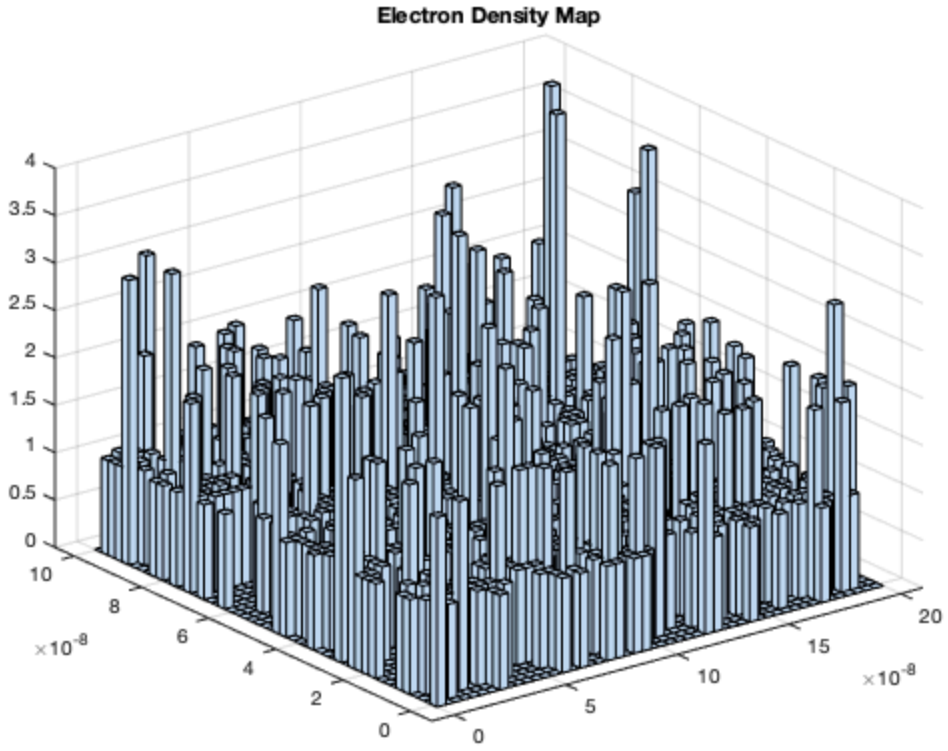
```
    pscatvector = ones(1,numofelec)*pscat;

    colorarray= rand(1,20);
for time= 1:dt:250

    %movement of the electrons is now impacted by acceleration in the
 x
    %direction

    v_x = v_x + a_elec*(dt*3e-15);


    random = rand(1,numofelec);

    %all electrons with higher probabilities
    new = random < pscat;

    %all electrons with lower probabilities
    new2 = random >= pscat;


    rand_v_x = zeros(1,numofelec);
    rand_v_y = zeros(1,numofelec);

   for i = 1:1:numofelec
     r1 = randi([1 numofelec], 1,1);
     r2 = randi([1 numofelec], 1,1);
        rand_v_x(1,i) = v_x(1,r1);
        rand_v_y(1,i) = v_y(1,r2);
   end
        %all electrons with lower probabilities will stay the same
   v_x = v_x.*new2;
   v_y = v_y.*new2;

   rand_v_x=rand_v_x.*new;
   rand_v_y=rand_v_y.*new;

   v_x = v_x+rand_v_x;
   v_y = v_y+rand_v_y;

     dx = v_x*dt*1e-15*5;
     dy = v_y*dt*1e-15*5;
%
     x = x + dx;
     y = y + dy;
%
    %if y is greater than 200
    temp = y>=Width*1e-9;
    temp1 = y<Width*1e-9;

    temp = temp*(-1);

    temphigher = temp + temp1;
```

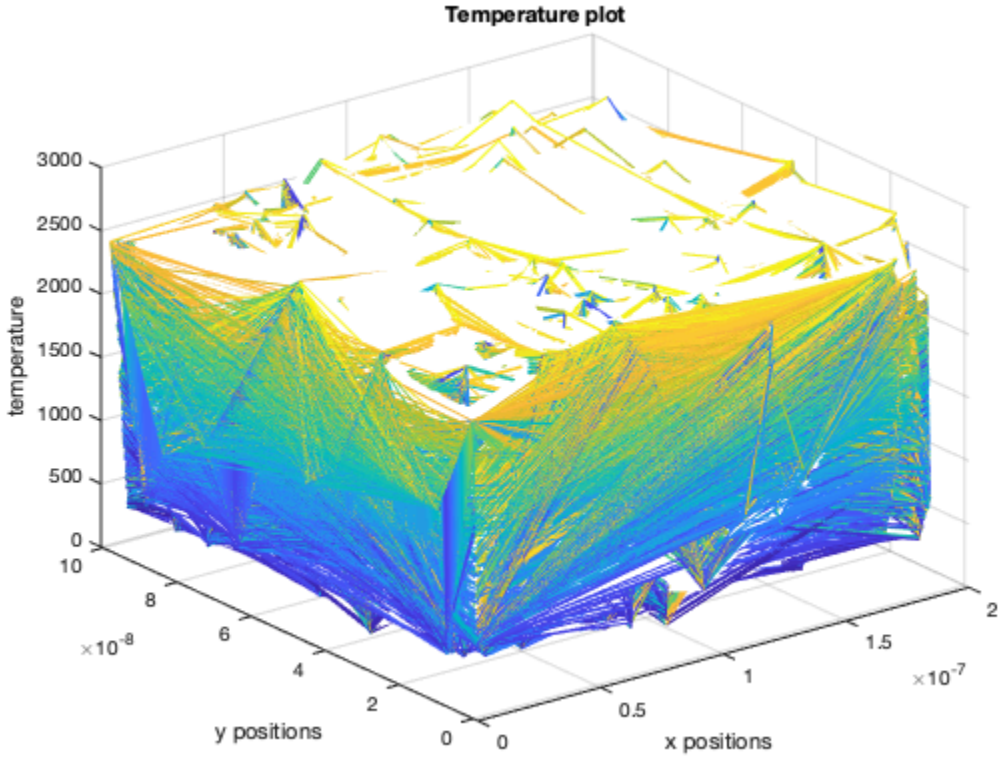
```
    v_y = temphigher.*v_y;

     %if y is less than 100
    temp2 = y>=0;
    temp3 = y<0;

    temp3 = temp3*(-1);
    templower = temp3 + temp2;
    v_y = templower.*v_y;



 %if x greater than 200
   temp5 = x<200*1e-9;

   x = x .* temp5;

   %if x is less than 0
   temp4 = x< 0;
   temp4 = temp4*200*1e-9;

   %temp4 = temp4*200*1e-9;
   x = x + temp4;
%
    %average thermal velocity
    v_avg = mean(sqrt((v_x.^2)+(v_y.^2)));
    v_matrix = sqrt((v_x.^2)+(v_y.^2));
    T_avg = (mn*(v_avg^2))/kB;

    T_matrix = (mn*(v_matrix.*v_matrix))/kB;

    %mean free path

    mfp = (10^-15)*(v_avg);

        %setting up plot for 20 electrons
    for q =1:1:20
        plotx(q) = x(q);
        ploty(q) = y(q);
    end

    figure(1)
    scatter(plotx,ploty,3,colorarray);
    axis([0 200*10^-9 0 100*10^-9])
    title(['The mean free path is ', num2str(mfp)]);
   % pause(0.00000000000000000000000000000000000000000001)
    hold on

    %calculation of drift current of electron

    elec_conc = 10^15;          %electron concentration given in
 outline
    I_d = v_avg*elec_conc*elec_x*q_0;
```

```
 %PART D setting up the plot for
    figure(2)
    scatter(time,I_d,'r.')
    title('current density of electrons')
    hold on

end

%PART E electron density map and temperature map
dens_mat = [x(:) y(:)];
figure(4)
hist3(dens_mat(:,1:2) ,[50 50]);
title("Electron Density Map")

%temperature plot
    [X,Y] = meshgrid (x' , y');
    f1 = scatteredInterpolant(x',y',T_matrix');
    Z = f1(X,Y);
    figure (5);
    mesh(X,Y,Z);
    title('Temperature plot')
    xlabel('x positions')
    ylabel('y positions')
    zlabel('temperature')

Part a) The value of the electric field on the electrons is 500000
Part b) The value of force on each electron is 8.010883e-14
Part c) The value of acceletation on each electron is
 338234653108549376
```


The mean free path is 2.6994e-10
×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


current density of electrons
×10^7
2.2
2
1.8
1.6
1.4
1.2
1
0
50
100
150
200
250


Electron Density Map
4
3.5
3
2.5
2
1.5
1
0.5
0
10
8
6
4
2
0
×10^-8
0
5
10
15
20
×10^-8


Temperature plot
temperature
3000
2500
2000
1500
1000
500
0
10
8
6
4
2
0
×10^-8
y positions
0
0.5
1
1.5
2
×10^-7
x positions

Published with MATLAB® R2019b