ARJUN BHATTI – Assignment 1 – 100915440 – ELEC 4700

Part 1: Electron Modelling

```
clc
clear all
%% constants

m0 = 9.10938356e-31;
m = 0.26*m0;
T = 300;
k = 1.38064852e-23;
VTH = sqrt(2*k*T/m)*.0010 % formula
disp("187 km/hr");
MFP = VTH*0.2;
disp(MFP);  %% nm
```

1. What is the thermal velocity vth? Assume T = 300K. `VTH = 187.0191 km/hr`
2. If the mean time between collisions is τmn = 0.2ps what is the mean free path? `MFP=37.4038 nm`

Variables used :

```
H = 100e-9;  %Height
L = 200e-9; %length
population_size = 3000;
Population_toPlot = 20;   % values from pdf
time  = H/VTH/100;
iterations = 1000;    % from pdf
Movie_ONOFF = 0; % movie  =1 or 0 for no movie

% generating matrix with zeroes.incudes velocity and position
Matrix1 = zeros(population_size, 4);
Zeroes = zeros(iterations, Population_toPlot*2);
Zeroes1 = zeros(iterations,1);   %Zeroes1
```

Q3 • Assign each particle a random location in the x − y plane within the region defined by the extent of the Silicon. For simplicity you may use a small number of particles (1000-10000 works well) but you can start much smaller initially if you like.
• Assign each particle with the fixed velocity given by vth but give each one a random direction.
• At a fixed time interval of 4t, update the particle location using Newton’s laws of motion. You will need to pick a time step size that takes into account the velocity of your particles and the size of the region. Typically the spacial step should be smaller than 1/100 of the region size. Simulate for nominally 1000 timesteps.This should allow each particle to bounce around quite a bit inside the region.
• For a few of the particles trace out their trajectories using the ’plot’ command in Matlab. To plot the trajectories you should keep the previous x and y positions.
use the ’pause’ command in Matlab to have the plot update in a loop.
• Show a 2-D plot of all (or a subset) of the particles that updates with each time
step. Hint: use the ’pause’ command in Matlab to have the plot update in a loop.

• For the y direction use a boundary condition where the particle reflects at the same angle (specular) and retains its velocity.

• For the x direction use a periodic boundary condition where the particle jumps
to the opposite edge. i.e. if it reaches the right side it appears at the left with
the same velocity.

• Calculate and display the semiconductor temperature on the plot at a fixed time
interval and verify that it stays constant.

• Your program should plot trace trajectories producing something like Figure

```
%  initial population - for figure 1

 for i = 1:population_size
 angle = 2*pi*rand;
 Matrix1(i,:) = [L*rand H*rand VTH*cos(angle) VTH*sin(angle)];
end
```

```
 %itertion over time - changes matrix1
for i = 1:iterations
Matrix1(:,1:2) = Matrix1(:,1:2) + time.*Matrix1(:,3:4);


j = Matrix1(:,1) > L;
%greter than length
Matrix1(j,1) = Matrix1(j,1) - L;

j = Matrix1(:,1) < 0;
Matrix1(j,1) = Matrix1(j,1) + L;

j = Matrix1(:,2) > H;  %greater than height
Matrix1(j,2) = 2*H - Matrix1(j,2);
Matrix1(j,4) = -Matrix1(j,4);

j = Matrix1(:,2) < 0;
Matrix1(j,2) = -Matrix1(j,2);
Matrix1(j,4) = -Matrix1(j,4);

Zeroes1(i) = (sum(Matrix1(:,3).^2) +
sum(Matrix1(:,4).^2))*m/k/2/population_size;
```

```
% Record the Zeroes
 for j=1:Population_toPlot
 Zeroes(i, (2*j):(2*j+1)) = Matrix1(j, 1:2);
end

% create movie every 5 changes

 if Movie_ONOFF && mod(i,5) == 0
 figure(1);
 subplot(2,1,1);
 hold off;
 plot(Matrix1(1:Population_toPlot,1)./1e-9,
Matrix1(1:Population_toPlot,2)./1e-9, 'O');
 axis([0 L/1e-9 0 H/1e-9]);
 title(sprintf('Zeroes for %d of %d Electrons with Fixed
Velocity (Part 1)',...
 Population_toPlot, population_size));
 xlabel(' x -(nm)');
 ylabel(' y - (nm)');

if i > 1
     subplot(2,1,2);
     hold off;
     plot(time *(0:i-1), Zeroes1(1:i));
     axis([0 time *iterations min(Zeroes1)*0.98
max(Zeroes1)*1.02]);
     title('Semiconductor');
     xlabel('Time (s)');
     ylabel('Temp  (K)');
 end
 pause(0.02);
 end
end

% Figure 1 shows movement of electrons with constant
velocity,along with Time vs temperature graph.
figure(1);
subplot(2,1,1);
title(sprintf('Electron Zeroes for %d of %d Electrons with Fixed
Velocity (Part 1)',...
Population_toPlot, population_size));
xlabel('x (nm)');
ylabel('y (nm)');
axis([0 L/1e-9 0 H/1e-9]);
hold on;
```

```
for i=1:Population_toPlot
 plot(Zeroes(:,i*2)./1e-9, Zeroes(:,i*2+1)./1e-9, '.');
end

if(~Movie_ONOFF)
subplot(2,1,2);
hold off;
plot(time *(0:iterations-1), Zeroes1);
axis([0 time *iterations min(Zeroes1)*0.98 max(Zeroes1)*1.02]);
title(' N-type Si semiconductor crystal@ 300K');
 xlabel('Time (s)');
 ylabel(' (K)');
end



%% Part 2: Collisions with Mean Free Path
% Maxwell-Boltzmann distribution

p_scat = 1 - exp(-time /0.2e-12)
```

p_scat =

0.0264

avg_v =

1.8699e+05

Histogram :

```
% Show histogram of speeds

subplot(3,1,3);

b = sqrt(Matrix1(:,3).^2 + Matrix1(:,4).^2);

title('Histogram of Electron Speeds');

histogram(b);

xlabel('Speed (m/s)');

ylabel('Number of particles');
```

```
pause(0.05);
```

Part 3: **Enchancements**

```
% create box as obstacles.
% create simulation for boxes.
%  each box values
boxes = 1e-9.*[80 120 0 40; 80 120 60 100];
boxes_specular = [0 1];
for i = 1:iterations
Matrix1(:,1:2) = Matrix1(:,1:2) + time .*Matrix1(:,3:4);
j = Matrix1(:,1) > L;
Matrix1(j,1) = Matrix1(j,1) - L;
j = Matrix1(:,1) < 0;
Matrix1(j,1) = Matrix1(j,1) + L;
j = Matrix1(:,2) > H;
top_Value =0;
bottom_val =0;
if(top_Value)
Matrix1(j,2) = 2*H - Matrix1(j,2);
Matrix1(j,4) = -Matrix1(j,4);
else % Diffusive
% The electron bounces off at a random angle
Matrix1(j,2) = H;
```

```
b = sqrt(Matrix1(j,3).^2 + Matrix1(j,4).^2);

angle = rand([sum(j),1])*2*pi;

Matrix1(j,3) = b.*cos(angle);

Matrix1(j,4) = -abs(b.*sin(angle));

End


% Scatter particles

j = rand(population_size, 1) < p_scat;

Matrix1(j,3:4) = random(v_pdf, [sum(j),2]);


Zeroes1(i) = (sum(Matrix1(:,3).^2) +
sum(Matrix1(:,4).^2))*m/k/2/population_size;

------------------------------------------------------------

subplot(3,1,3);

b = sqrt(Matrix1(:,3).^2 + Matrix1(:,4).^2);

title('Histogram of Electron Speeds');

histogram(b);

xlabel('Speed (m/s)');

ylabel('# of particles');

pause(0.03);
```

References

https://github.com/villetiukuvaara/elec-4700-assignment-1. (n.d.). Retrieved from github.com.