Assignment 1

Table of Contents

Question 1: Electron Modelling 1
Thermal Velocity 1
Mean Free Path 1

Matthew Lazarus 100962142

Question 1: Electron Modelling

In this question, a number of electrons are randomly positioned within a set grid. With the system set to 300K, each electron moves at the thermal velocity in a random direction. When an electron hits the top of the grid, it bounces back, and when it hits the side of the grid, it continues its trajectory from the opposite side of the grid.

```
% Clear all previous variables, figures, etc, to ensure that the
 workspace
% is clean.
clear all
clearvars
clearvars -GLOBAL
close all

%Define constants that may need to be used later in the code.
global C
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 sÂ²
```

Thermal Velocity

The Thermal Velocity at 300K can be found knowing that $v_{Th} = k_B * T/m$

```
vth = sqrt(C.kb*300/(0.26*C.m_0));
```

Therefore the thermal velocity is $1.3224 * 10^5 m/s$.

Mean Free Path

Additionally, the Mean Free Path can be found knowing the time between collisions. The equation $v = d/t$ can be rearranged to sovle for the mean free path.

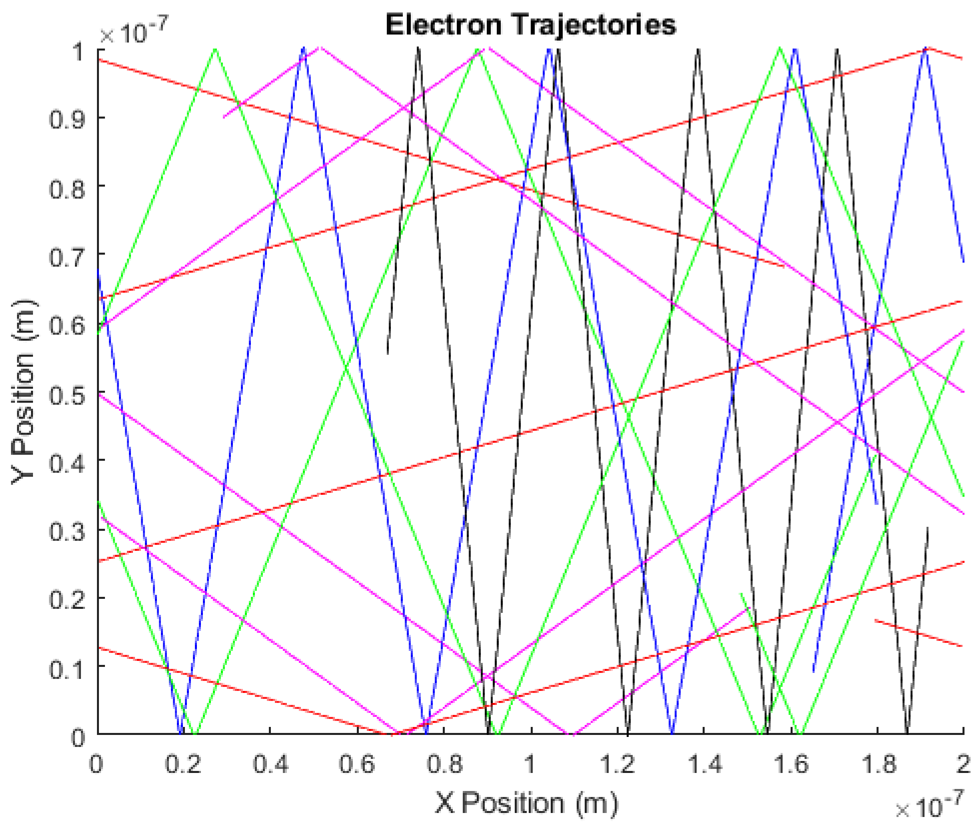
```
tMN = 0.2*10^-12;
freePath = tMN * vth;
```

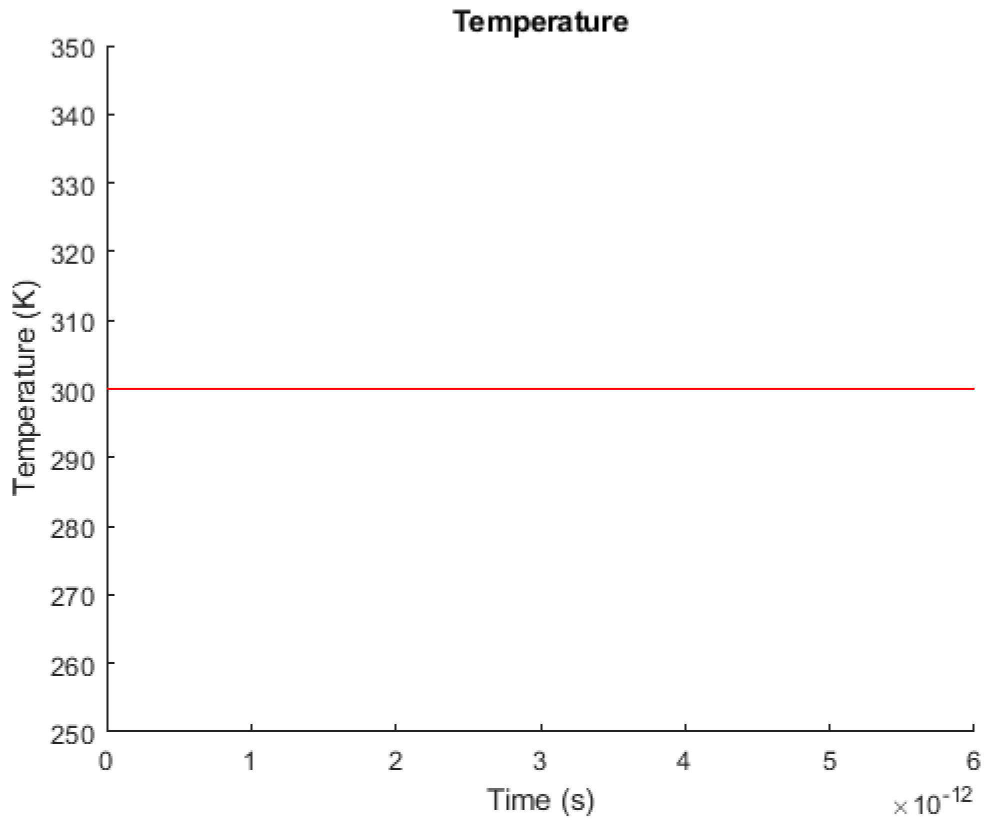
Therefore the mean free path is $2.6449 * 10^{-8}m$.

```
% Set the number of electrons, time step and total time. Initialize
% matrices for the x and y positions, the x and y components of the
% velocity, and the temperature of the system. Column 1 of each
 matrice is
% the previous value, while column 2 is the current value.
numElectrons=10000;
dt = 6e-15; %seconds
nTime = 6e-12; %Simulation length
x = zeros(numElectrons,2); %Position (x)
y = zeros(numElectrons, 2); %Position (y)
vx = zeros(numElectrons, 2); %Velocity (x)
vy = zeros(numElectrons, 2); %Velocity (y)
temperature = zeros(numElectrons,2);

% Now, randomly assign initial positions & directions.
for electronCount = 1:numElectrons
    x(electronCount,2)=rand()*200e-9;
    y(electronCount,2)=rand()*100e-9;

    startAngle = rand()*2*pi;
    vx(electronCount,2) = vth * cos(startAngle);
    vy(electronCount,2) = vth * sin(startAngle);
end

% Create a figure for the electron trajectories and the temperature of
 the
% system.
figure(1)
title('Electron Trajectories')
xlabel('X Position (m) ')
ylabel('Y Position (m)')
axis([0 200e-9 0 100e-9]);

figure(2)
title('Temperature')
xlabel('Time (s)')
ylabel('Temperature (K)')
axis([0 (nTime) 250 350]);

% Define a vector that will indicate whether an electron crosses a
 horizontal
% boundary. As only 5 electrons will be plotted, if the electron cross
 the
% boundary, a 1 will be set in the position of the vector that
 corresponds
% to the number of the electron (1-5).
xBreakpoint = zeros(5);

% Run simulation over time.
```

```
for count =  1:ceil((nTime)/dt)
    % Run through each electron.
    for c = 1:numElectrons
        if(count~=1)
            % Update the previous positions and velocities.
            vx(c,1)=vx(c,2);
            vy(c,1)=vy(c,2);
            x(c,1)=x(c,2);
            y(c,1)=y(c,2);

            % Update the current position of the electron.
            x(c,2) = x(c,1) + vx(c, 2)*dt;
            y(c,2) = y(c,1) + vy(c, 2)*dt;

            % Check to see if an electron hit a boundary. If it hit a
            % horizontal boundary, move it to the other side of the
 grid
            % (with the same velocity). If it hit a vertical boundary,
 it
            % should bounce off.
            if(x(c,2)>200e-9)
                x(c,2) = x(c,2)-200e-9;
                if(c<6 && c>0)
                    xBreakpoint(c)=1;
                end
            elseif(x(c,2)<0)
                x(c,2)=x(c,2)+200e-9;
                if(c<6 && c>0)
                   xBreakpoint(c)=1;
                end
            end
            if(y(c,2)>=100e-9)
                vy(c,2) = -vy(c,2);
            elseif(y(c,2)<=0)
                vy(c,2)=-vy(c,2);
            end
        end
    end

    if(count>1)
        % Plot the displacement of the electrons in different colours.
        figure(1)
        hold on
        if(xBreakpoint(1)~=1)
            plot(x(1,1:2),y(1,1:2),'b')
        end
        if(xBreakpoint(2)~=1)
            plot(x(2,1:2),y(2,1:2),'r')
        end
        if(xBreakpoint(3)~=1)
            plot(x(3,1:2),y(3,1:2),'g')
        end
        if(xBreakpoint(4)~=1)
            plot(x(4,1:2),y(4,1:2),'k')
```

```
        end
        if(xBreakpoint(5)~=1)
            plot(x(5,1:2),y(5,1:2),'m')
        end
        hold off
    end
    % Reset xBreakpoint.
    xBreakpoint(:)=0;

    % Update the previous and current temperature values. Plot the
 change
    % in temeperature over the step in time.
    temperature(:,1)=temperature(:,2);
    temperature(:,2) = (vx(:,2).^2 + vy(:,2).^2).*((0.26*C.m_0))./
C.kb;
    if(count>1)
        figure(2)
        hold on
        plot([(count-1)*dt,count*dt],
[mean(temperature(:,1)),mean(temperature(:,2))],'r');
        hold off
    end

    % Take a short pause to allow the screen to update.
    pause(0.000001)
end
```





Published with MATLAB® R2018b