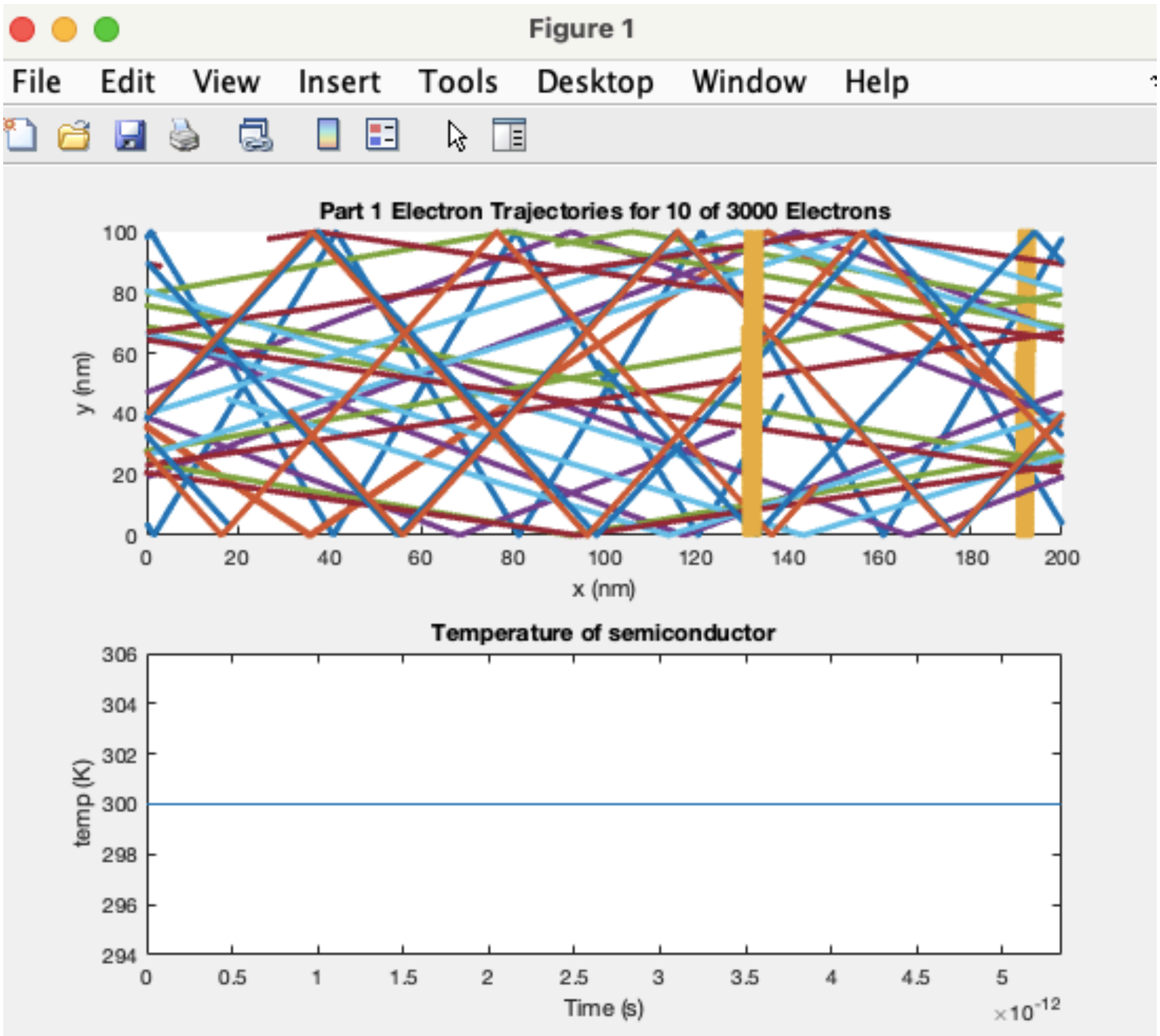
Elec4700A
Assignment #1

Muhammad Shabeeh
Raza Abbas
101092004

Date: 7th Feb 2021

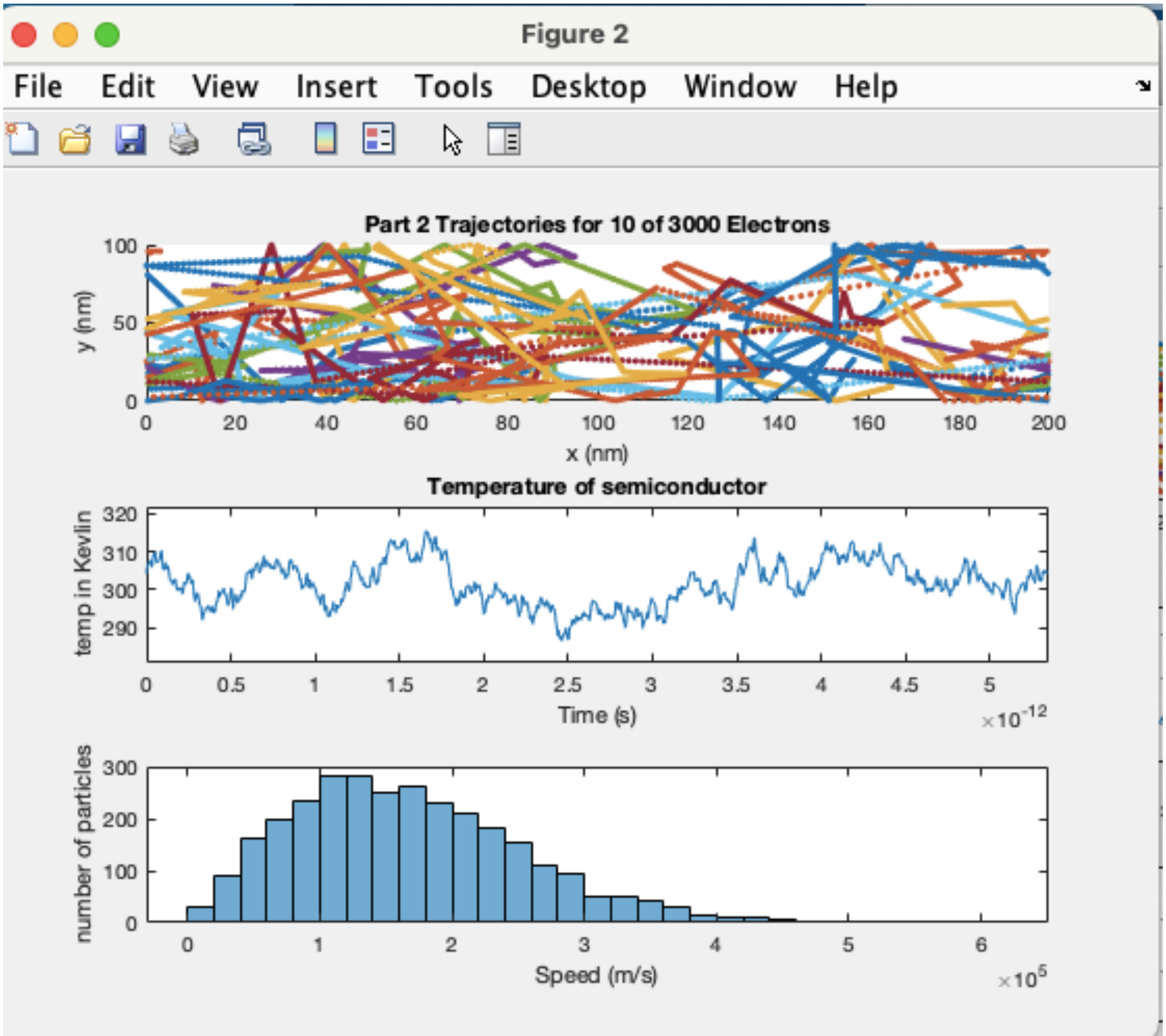
Part 1: Electron Modelling,

Q2) Calculating the mean free path,
vthev = 1.8702e+05

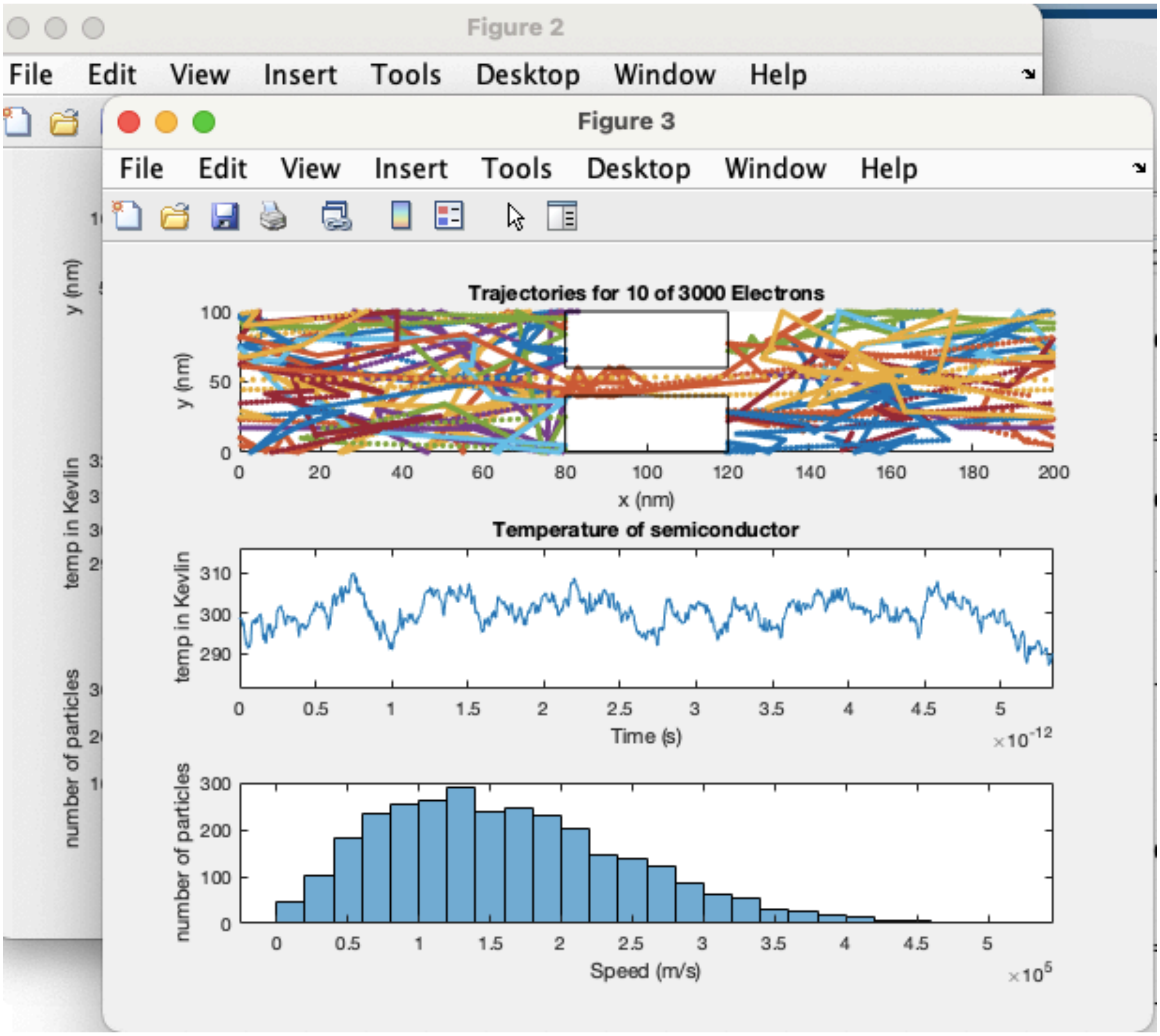


Part 2: Collisions with Mean Free Path (MFP)

Q3) The average temperature fluctuates over time due to the scattering, but it maintains an average of 300 K- 310 K over time as instructed in the lab manual and shown below.



Part 3: Enhancements,



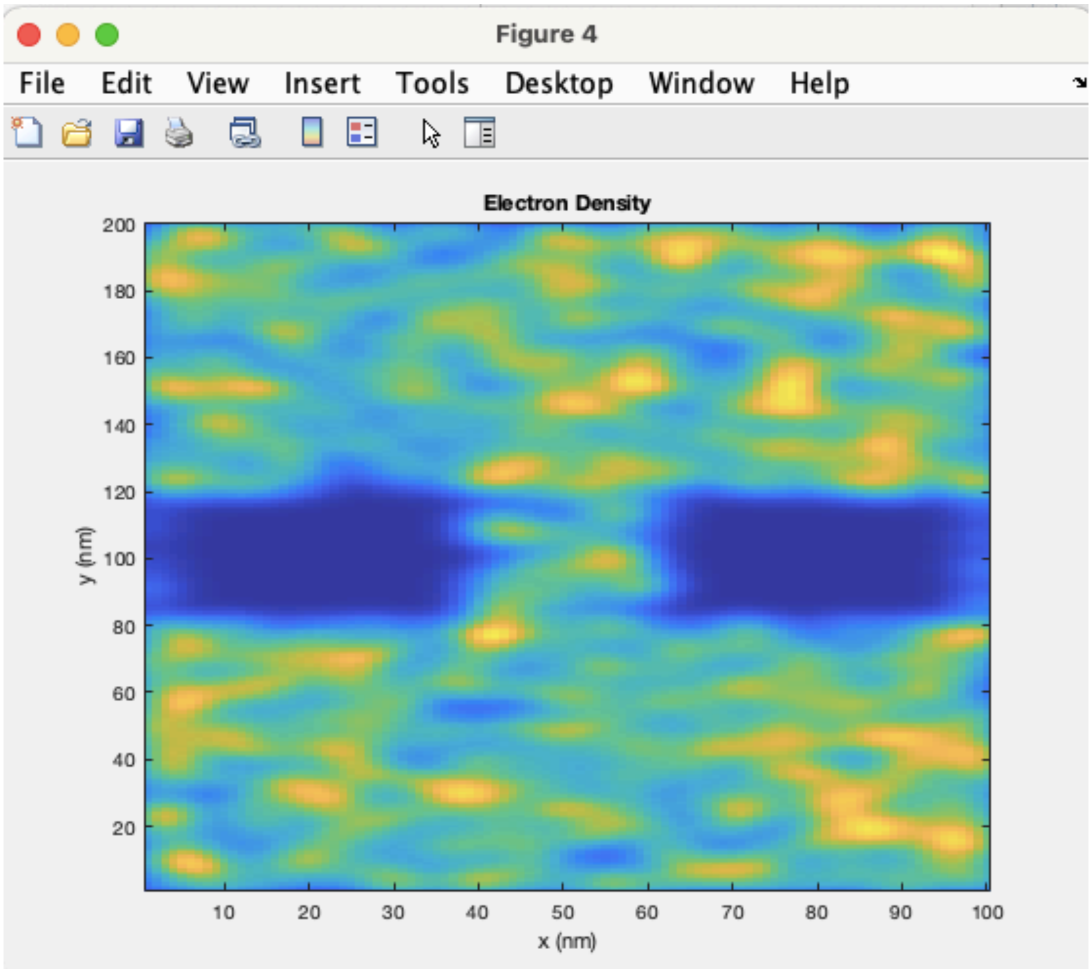


Figure 4
File Edit View Insert Tools Desktop Window Help
Electron Density
200
180
160
140
120
100
80
60
40
20
y (nm)
10 20 30 40 50 60 70 80 90 100
x (nm)

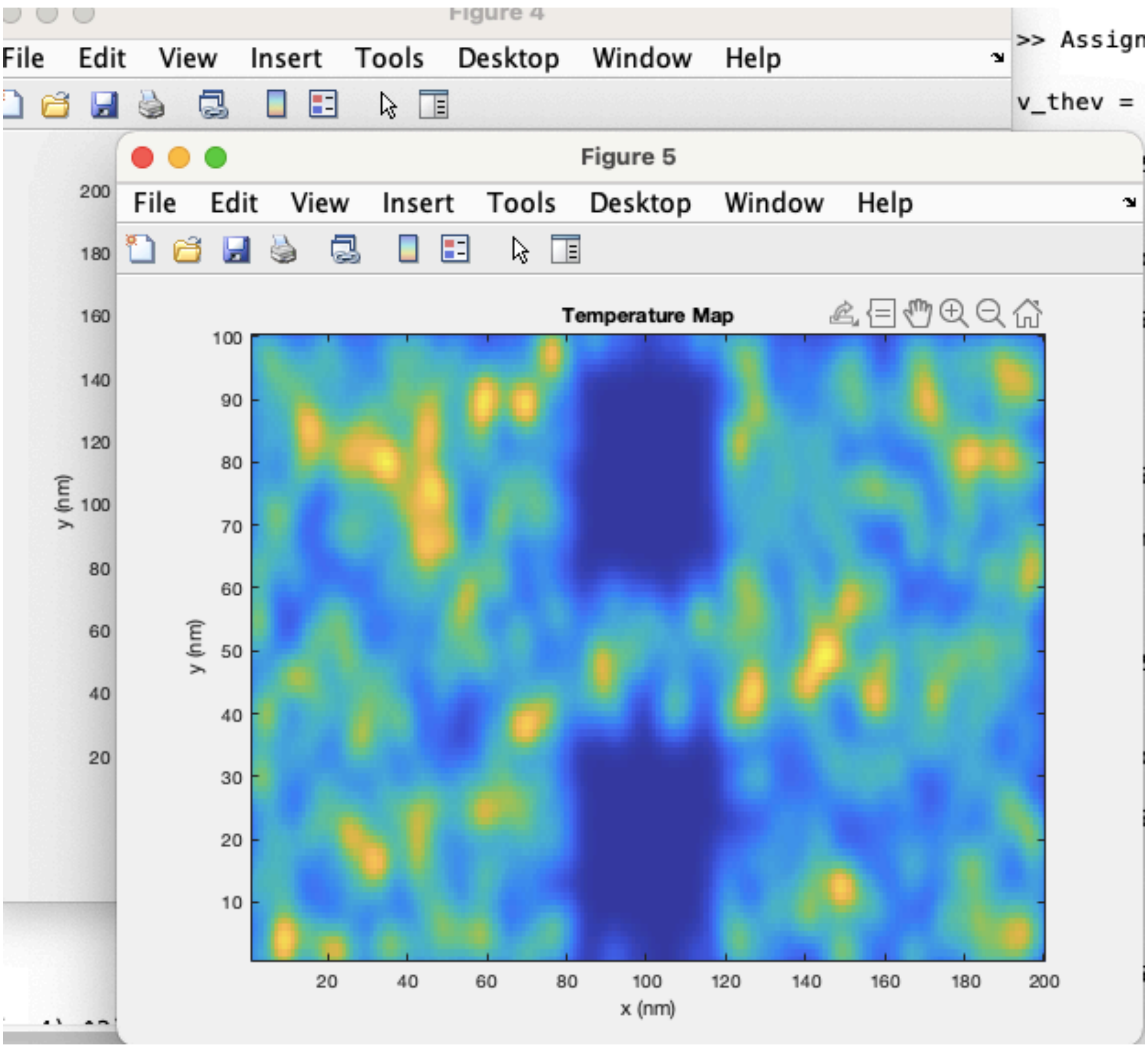


Figure 4
File Edit View Insert Tools Desktop Window Help
>> Assign
v_thev =
Figure 5
File Edit View Insert Tools Desktop Window Help
Temperature Map
y (nm)
x (nm)

```
%Muhammad Shabeeh Raza Abbas 101092004
%Part 1 Electron Modelling
clear all
close all
m = 9.10938356e-31; %electron mass
K = 1.38064852e-23; %Boltzmann's constant
Temp = 300; %Temp in Kelvin
m = 0.26*m; % Given in the mamaual
v_thev = sqrt(2*K*Temp/m) %eqyation given in manual
width = 100e-9;
lenght = 200e-9;
numElec = 3000;
numElecPlot = 10;
time_step = width/v_thev/100;
iterations = 1000;
particalemovement= 0;

% Each row maps electron with positions/velocities [x y vx vy]
state = zeros(numElec, 4);
trajectories = zeros(iterations, numElecPlot*2);
temp = zeros(iterations,1);
for i = 1:numElec
    angle = rand*2*pi;
    state(i,:) = [lenght*rand width*rand v_thev*cos(angle) v_thev*sin(angle)];
end

for i = 1:iterations
    state(:,1:2) = state(:,1:2) + time_step.*state(:,3:4);
    % Search for collisions along boundaries
    j = state(:,1) > lenght;
    state(j,1) = state(j,1) - lenght;
    j = state(:,1) < 0;
    state(j,1) = state(j,1) + lenght;
    j = state(:,2) > width;
    state(j,2) = 2*width - state(j,2);
    state(j,4) = -state(j,4);
    j = state(:,2) < 0;
    state(j,2) = -state(j,2);
    state(j,4) = -state(j,4);
    temp(i) = (sum(state(:,3).^2) + sum(state(:,4).^2))*m/K/2/numElec;

    % Record trajectories
    for j=1:numElecPlot
        trajectories(i, (2*j):(2*j+1)) = state(j, 1:2);
    end

    % Updating motion every 5 iters
    if particalemovement && mod(i,5) == 0
        figure(1);
        subplot(2,1,1);
        hold off;
        plot(state(1:numElecPlot,1)./1e-9,state(1:numElecPlot,2)./1e-9, 'o');
        axis([0 lenght/1e-9 0 width/1e-9]);
        title(sprintf('Part 1 Electron Trajectories for %d of %d
Electrons',numElecPlot, numElec));
        xlabel('x(nm)');
        ylabel('y(nm)');
        if i > 1
            subplot(2,1,2);
            hold off;
            plot(time_step*(0:i-1), temp(1:i));
```

```
            axis([0 time_step*iterations min(temp)*0.98
                max(temp)*1.02]);
            title('Temperature of semiconductor');
            xlabel('Time(s)');
            ylabel('Temperature(K)');
        end
        pause(0.05);
    end
end


    % Plotting trajectories and subplot of temp
figure(1);
subplot(2,1,1);
title(sprintf('Part 1 Electron Trajectories for %d of %d Electrons',numElecPlot,
numElec));
xlabel('x (nm)');
ylabel('y (nm)');
axis([0 lenght/1e-9 0 width/1e-9]);
hold on;
for i=1:numElecPlot
    plot(trajectories(:,i*2)./1e-9, trajectories(:,i*2+1)./1e-9, '.');
end
if(~particalemovement)
    subplot(2,1,2);
    hold off;
    plot(time_step*(0:iterations-1), temp);
    axis([0 time_step*iterations min(temp)*0.98 max(temp)*1.02]);
    title('Temperature of semiconductor');
    xlabel('Time (s)');
    ylabel('temp (K)');
end




%PART 2 Collisions with Mean Free Path (MFP)
%
scatter_p = 1 - exp(-time_step/0.2e-12)

pdf_v = makedist('Normal', 'mu', 0, 'sigma', sqrt(K*Temp/m));
for i = 1:numElec
    angle = rand*2*pi;
    state(i,:) = [lenght*rand width*rand random(pdf_v) random(pdf_v)];
end
avg_v = sqrt(sum(state(:,3).^2)/numElec + sum(state(:,4).^2)/numElec)
%the for loop used below was frmm the aid of an online forum i was unable
%to sucessfully do it on my own and looked through online forum and other
%websites and modified my code to suite this assignment.
for i = 1:iterations
    state(:,1:2) = state(:,1:2) + time_step.*state(:,3:4);
    j = state(:,1) > lenght;
    state(j,1) = state(j,1) - lenght;
    j = state(:,1) < 0;
    state(j,1) = state(j,1) + lenght;
    j = state(:,2) > width;
    state(j,2) = 2*width - state(j,2);
    state(j,4) = -state(j,4);
    j = state(:,2) < 0;
    state(j,2) = -state(j,2);
    state(j,4) = -state(j,4);
```

```
    j = rand(numElec, 1) < scatter_p;
    state(j,3:4) = random(pdf_v, [sum(j),2]);

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

    for j=1:numElecPlot
        trajectories(i, (2*j):(2*j+1)) = state(j, 1:2);
    end

    if particalemovement && mod(i,5) == 0
        figure(2);
        subplot(3,1,1);
        hold off;
        plot(state(1:numElecPlot,1)./1e-9, state(1:numElecPlot,2)./1e-9, 'o');
        axis([0 lenght/1e-9 0 width/1e-9]);
        title(sprintf('Part 2 Trajectories for %d of %d Electrons', numElecPlot,
numElec));
        xlabel('x (nm)');
        ylabel('y (nm)');
        if i > 1
            subplot(3,1,2);
            hold off;
            plot(time_step*(0:i-1), temp(1:i));
            axis([0 time_step*iterations min(temp)*0.98
                max(temp)*1.02]);
            title('Temperature of semiconductor');
            xlabel('Time (s)');
            ylabel('temp (K)');
        end
        % Show histogram of speeds
        subplot(3,1,3);
        v = sqrt(state(:,3).^2 + state(:,4).^2);
        title('Histogram of Electron Speeds');
        histogram(v);
        xlabel('Speed (m/s)');
        ylabel('# of particles');
        pause(0.05);
    end
end

figure(2);
subplot(3,1,1);
title(sprintf('Part 2 Trajectories for %d of %d Electrons',numElecPlot,
numElec));
xlabel('x (nm)');
ylabel('y (nm)');
axis([0 lenght/1e-9 0 width/1e-9]);
hold on;
for i=1:numElecPlot
    plot(trajectories(:,i*2)./1e-9, trajectories(:,i*2+1)./1e-9, '.');
end

% Show temp plot/time
if(~particalemovement)
    subplot(3,1,2);
    hold off;
    plot(time_step*(0:iterations-1), temp);
    axis([0 time_step*iterations min(temp)*0.98 max(temp)*1.02]);
    title('Temperature of semiconductor');
    xlabel('Time (s)');
    ylabel('temp (K)');
end
% Speed histogram
```

```
subplot(3,1,3);
v = sqrt(state(:,3).^2 + state(:,4).^2);
title('Histogram of Electron Speeds');
histogram(v);
xlabel('Speed (m/s)');
ylabel('# of particles');




%Part 3 Enhancments


specular_upper = 0;
specular_lower = 0;
boxes = 1e-9.*[80 120 0 40; 80 120 60 100];
boxes_specular = [0 1];
% Generating the initial popul
for i = 1:numElec
    angle = rand*2*pi;
    state(i,:) = [lenght*rand width*rand random(pdf_v) random(pdf_v)];
    % Check for Box Contents
    while(for_box(state(i,1:2), boxes))
        state(i,1:2) = [lenght*rand width*rand];
    end
end

for i = 1:iterations
    state(:,1:2) = state(:,1:2) + time_step.*state(:,3:4);
    j = state(:,1) > lenght;
    state(j,1) = state(j,1) - lenght;
    j = state(:,1) < 0;
    state(j,1) = state(j,1) + lenght;
    j = state(:,2) > width;
    if(specular_upper)
        state(j,2) = 2*width - state(j,2);
        state(j,4) = -state(j,4);
    else
        % The electrons bouncing off at a random angle
        state(j,2) = width;
        v = sqrt(state(j,3).^2 + state(j,4).^2);
        angle = rand([sum(j),1])*2*pi;
        state(j,3) = v.*cos(angle);
        state(j,4) = -abs(v.*sin(angle));
    end
    j = state(:,2) < 0;
    if(specular_lower)
        state(j,2) = -state(j,2);
        state(j,4) = -state(j,4);
    else % Diffusive
        % The electron bounces off at a random angle
        state(j,2) = 0;
        v = sqrt(state(j,3).^2 + state(j,4).^2);
        angle = rand([sum(j),1])*2*pi;
        state(j,3) = v.*cos(angle);
        state(j,4) = abs(v.*sin(angle));
    end

    % Moving electrons to their positions after entering box
    for j=1:numElec
        box_num = for_box(state(j,1:2), boxes);
        while(box_num ~= 0)
```

```
        % Finding the electron collision side
        distance_x = 0;
        updated_x = 0;
        if(state(j,3) > 0)
            distance_x = state(j,1) - boxes(box_num,1);
            updated_x = boxes(box_num,1);
        else
            distance_x = boxes(box_num,2) - state(j,1);
            updated_x = boxes(box_num,2);
        end
        distance_y = 0;
        updated_y = 0;
        if(state(j,4) > 0)
            distance_y = state(j,2) - boxes(box_num, 3);
            updated_y = boxes(box_num, 3);
        else
            distance_y = boxes(box_num, 4) - state(j,2);
            updated_y = boxes(box_num, 4);
        end

        if(distance_x < distance_y)
            state(j,1) = updated_x;
            if(~boxes_specular(box_num))
                sgn = -sign(state(j,3));
                v = sqrt(state(j,3).^2 + state(j,4).^2);
                angle = rand()*2*pi;
                state(j,3) = sgn.*abs(v.*cos(angle));
                state(j,4) = v.*sin(angle);
            else
                state(j,3) = -state(j,3);
            end

        else
            state(j,2) = updated_y;
            if(~boxes_specular(box_num))
                sgn = -sign(state(j,4));
                v = sqrt(state(j,3).^2 + state(j,4).^2);
                angle = rand()*2*pi;
                state(j,3) = v.*cos(angle);
                state(j,4) = sgn.*abs(v.*sin(angle));
            else
                state(j,4) = -state(j,4);
            end
        end
        box_num = for_box(state(j,1:2), boxes);
    end
end

% Scatter particles
j = rand(numElec, 1) < scatter_p;
state(j,3:4) = random(pdf_v, [sum(j),2]);

% Record the temp
temp(i) = (sum(state(:,3).^2) + sum(state(:,4).^2))*m/K/2/numElec;

% Record positions
for j=1:numElecPlot
    trajectories(i, (2*j):(2*j+1)) = state(j, 1:2);
end

% Update the motion for 5 iters
```

```
    if particalemovement && mod(i,5) == 0
        figure(3);
        subplot(3,1,1);
        hold off;
        plot(state(1:numElecPlot,1)./1e-9,state(1:numElecPlot,2)./1e-9, 'o');
        hold on;

        % Plotting the boxes
        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 lenght/1e-9 0 width/1e-9]);
        title(sprintf('Part 3 Trajectories for %d of %d Electrons',numElecPlot,
numElec));
        xlabel('x(nm)');
        ylabel('y(nm)');
        if i > 1
            subplot(3,1,2);
            hold off;
            plot(time_step*(0:i-1), temp(1:i));
            axis([0 time_step*iterations min(temp(1:i))*0.98
                max(temp)*1.02]);
            title('Temperature of semiconductor');
            xlabel('Time(s)');
            ylabel('Temperature(K)');
        end

        subplot(3,1,3);
        v = sqrt(state(:,3).^2 + state(:,4).^2);
        title('Electron Speeds Histogram');
        histogram(v);
        xlabel('Speed(m/s)');
        ylabel('# of particles');
        pause(0.05);
    end
end

% Showing trajectories after the motion is finished
figure(3);
subplot(3,1,1);
title(sprintf('Part 3 Trajectories for %d of %d Electrons', numElecPlot,
numElec));
xlabel('x (nm)');
ylabel('y (nm)');
axis([0 lenght/1e-9 0 width/1e-9]);
hold on;
for i=1:numElecPlot
    plot(trajectories(:,i*2)./1e-9, trajectories(:,i*2+1)./1e-9, '.');
end

% Plotting boxes
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

% Plotting temp
if(~particalemovement)
```

```
    subplot(3,1,2);
    hold off;
    plot(time_step*(0:iterations-1), temp);
    axis([0 time_step*iterations min(temp)*0.98 max(temp)*1.02]);
    title('Temperature of semiconductor');
    xlabel('Time (s)');
    ylabel('temp (K)');
end
subplot(3,1,3);
v = sqrt(state(:,3).^2 + state(:,4).^2);
title('Histogram of Electron Speeds');
histogram(v);
xlabel('Speed (m/s)');
ylabel('# of particles');

%Temp map
density = hist3(state(:,1:2),[200 100]);

% Smoothes out the density map
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(4);
imagesc(conv2(density,f,'same'));
set(gca,'YDir','normal');
title('Electron Density');
xlabel('x (nm)');
ylabel('y (nm)');
temp_sum_x = zeros(ceil(lenght/1e-9),ceil(width/1e-9));
temp_sum_y = zeros(ceil(lenght/1e-9),ceil(width/1e-9));
temp_num = zeros(ceil(lenght/1e-9),ceil(width/1e-9));

% velocities
for i=1:numElec
    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

    % Adds velocity to count
    temp_sum_y(x,y) = temp_sum_y(x,y) + state(i,3)^2;
    temp_sum_x(x,y) = temp_sum_x(x,y) + state(i,4)^2;
    temp_num(x,y) = temp_num(x,y) + 1;
end
temp = (temp_sum_x + temp_sum_y).*m./K./2./temp_num;
temp(isnan(temp)) = 0;
temp = temp';
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(5);
imagesc(conv2(temp,f,'same'));
set(gca,'YDir','normal');
title('Temperature Map');
xlabel('x (nm)');
```

```
ylabel('y (nm)');
```