Carleton University

Elec 4700 – Modelling of integrated Device

Assignment 1 - Monte-Carlo Modeling of Electron Transport

Done By: Yousef Nour

ID: 101046991

Question 1: Electron Modelling

The purpose of this code is to demonstrate the use of MATLAB to determine certain values and graphs. As you can see from the code, this question displays a solution to how electrons are modelled in a silicon as particles with a given effective mass, temperature, and other constants & variables. Using these givens, we were able to define the *thermal velocity* "Vt" (assuming temperature was 300 K), the *mean free path* "MFP", while the rest of this section shows the code to print and model the random motion of electrons given some specific requirements.

Q) What is the thermal velocity v_{th}? Assume T = 300K?

Ans) The thermal velocity, assuming T = 300K, is 132.24 km/s.

Q) If the mean time between collisions is Tmn = 0:2 ps what is the mean free path?

Ans) The mean free path, assuming a mean time of 0.2ps, is 26.45 nm.

Program code:

```
%% Part 1:Electron Modelling

Mo   = 9.1093837e-31; %electron rest mass
Mn   = 0.26*Mo;       %effective electron mass
Temp = 300;           %temp in kelvin
K    = 1.38064853e-23;% boltzmann's constant

%q1:What is the thermal velocity vth? Assume T = 300K.
V_th = sqrt(K*Temp/Mn);%thermal velocity calculation
V_display = V_th*1e-3;
%answer for thermal velocity
fprintf('The thermal velocity, assuming T = 300K, is %.2f km/s. \n',
V_display);

%q2:If the mean time between collisions is Tmn = 0.2 ps what is the mean free
path?
T_min = 0.2e-12; %Mean time
MFP   = V_th*T_min; % mean free path
MFP_display = MFP*1e9;
%answer for the mean free path
fprintf('The mean free path, assuming a mean time of 0.2ps, is %.2f nm.\n',
MFP_display)

%q3:Write a program that will model the random motion of the electrons
L = 200e-9;%length
W = 100e-9;%Width
popSize = 20; %Electron population size
popPlot = 5; % number of subset particles on plot
time = W/V_th/100;
iter = 1000;
sim_OnOff = 1;% intialize simulation to 1 (ON), if its 0 (OFF)

%intiallize matrices and vectors
```

```
matrix1 = zeros (popSize, 4);
col = 2*popPlot;
ZE = zeros (iter, col);
ZE1 = zeros (iter,1); %1000*1 vector of zeros

%populate the figure randomly
for i = 1:popSize
    theta = rand*2*pi; % angle
    a =  L*rand; b = W*rand; c = V_th*cos(theta); d = V_th*sin(theta);
    matrix1(i,:) = [a b c d];% recal: matrix1(row, column),so matrix1 (i,:)
entire thing of row i
end

%changes/iteration in matrix 1 over time
for i = 1:iter
    nextMat = time.*matrix1(:,3:4);
    matrix1(:,1:2) = matrix1(:,1:2) + nextMat;
    h = matrix1(:,1) > L;

    %calculation for matrix if greater than length
    matrix1(h,1) = matrix1(h,1) - L;
    h = matrix1(:,1) < 0;
    matrix1(h,1) = matrix1(h,1) + L;
    % calculation for matrix if greater than width
    h = matrix1(:,2) > W;
    matrix1(h,2) = 2*W - matrix1(h,2);
    matrix1(h,4) = -1*matrix1(h,4);

    h = matrix1(:,2) < 0;
    matrix1(h,2) = -1 * matrix1(h,2);
    matrix1(h,4) = -1 * matrix1(h,4);

    semiCon = Mn/K/popSize; ab = sum(matrix1(:,3).^2); cd =
sum(matrix1(:,4).^2);
    ZE1(i) = (ab + cd)*semiCon;

    %track the zeros (ZE)
    for h=1:popPlot
        s=2*h;
        ZE(i,(s):(s+1)) = matrix1(h, 1:2);
    end

%create a simulation for every 5 changes
if  sim_OnOff  && mod(i,5)==0

    figure(1);
    hold on;
    plot(matrix1(1:popPlot,1)./1e-9, matrix1(1:popPlot,2)./1e-9, 'X');
    axis([0 L/1e-9 0 W/1e-9]);
    title(sprintf('Zeros for %d particles of %d Electrons with Fixed Velocity
(Part 1)',popPlot, popSize));
    xlabel(' x -(nm)');
    ylabel(' y - (nm)');

    if i > 1
        figure (2);
        hold on;
```

```
        plot(time *(0:i-1), ZE1(1:i));
        axis([0 time *iter min(ZE1)*0.98 max(ZE1)*1.02]);
        title('Semiconductor at fixed Temperature (Part 1)');
        xlabel('Time (s)');
        ylabel('Temp  (K)');
    end
    pause(0.05);
 end
end

figure(1);
title(sprintf('Zeros for %d particles of %d Electrons with Fixed Velocity
(Part 1)',...
popPlot, popSize));
xlabel('x (nm)');
ylabel('y (nm)');
axis([0 L/1e-9 0 W/1e-9]);
hold on;

for i=1:popPlot
    f=i*2;
 plot(ZE(:,f)./1e-9, ZE(:,f+1)./1e-9, '-');
end

if(~sim_OnOff)
    figure (2);
    hold off;
    plot(time *(0:iter-1), ZE1);
    axis([0 time *iter min(ZE1)*0.98 max(ZE1)*1.02]);
    title(' N-type Si semiconductor at fixed Temperature');
    xlabel('Time (s)');
    ylabel('Temp (K)');
end
```

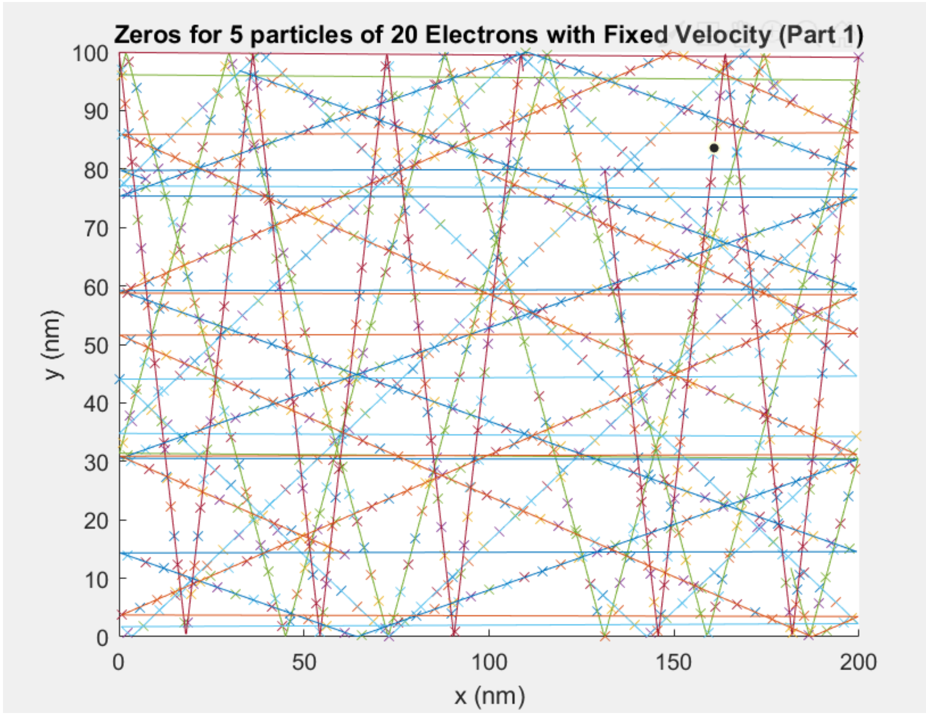

Figure 1: 2D plot of Particle Trajectory

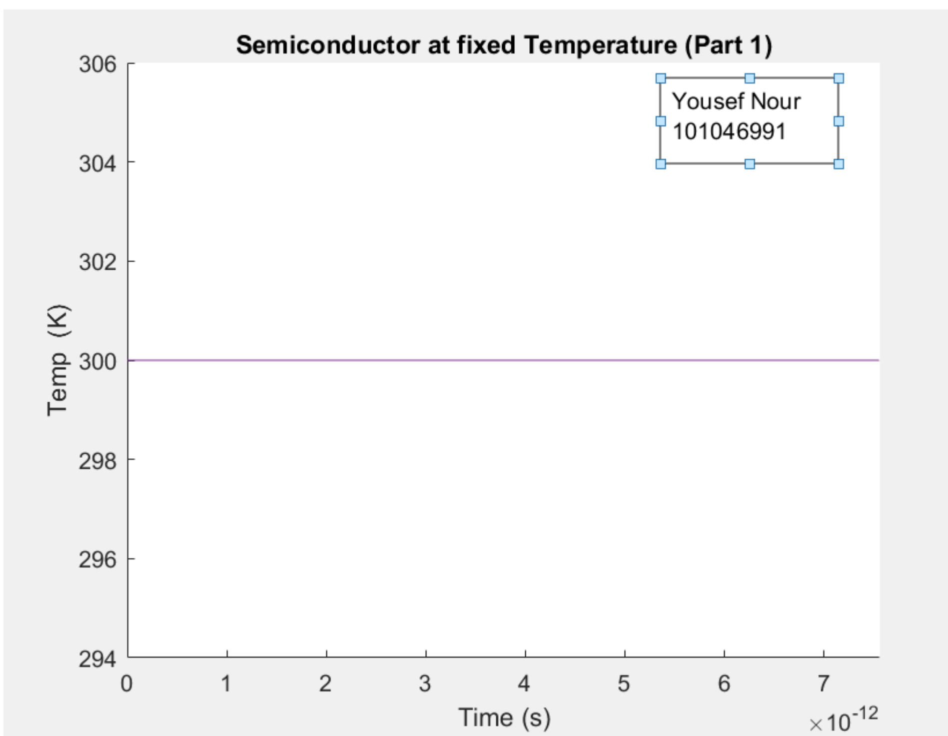

Figure 2: Temperature Plot

Question 2: Collisions with Mean Free Path (MFP)

This part of the assignment foceuses on assigning a random velocity to each of the particles at the start. Moreover, it models the scattering of the electrons using an exponential scattering probability.

Program code:

```
%% Part 2: Collisions with Mean Free Path

%Q2:Model the scattering of the electrons using an exponential scattering
probability
pScat = 1-exp(-1*time/0.2e-12);
%Create a Probability Distribution
distVal = sqrt(K*Temp/Mn);
pdfVol = makedist('Normal', 'mu', 0, 'sigma',distVal);% recal
makedist(DISTNAME,,PNAME1,PVAL1)

for i = 1:popSize
    theta = rand*2*pi;
    c = random(pdfVol);
    matrix1(i,:) = [a b c c];% recal: matrix1(row, column),so matrix1 (i,:)
entire column of row i
end

%Q3:What happens to the average temperature over time
avgTemp = sqrt((ab/popSize) + (cd/popSize));
fprintf('over time the average temp becomes: %f K.\n',avgTemp);

%Q4:Measure the actual Mean Free Path and mean time between collisions to
verify your model.

for i = 1:iter
    %record electron positions
    nextMat = time .*matrix1(:,3:4);
    matrix1(:,1:2) = matrix1(:,1:2) + nextMat;

    h = matrix1(:,1) > L;
    matrix1(h,1) = matrix1(h,1) - L;

    h = matrix1(:,1) < 0;
    matrix1(h,1) = matrix1(h,1) + L;

    h = matrix1(:,2) > W;
    matrix1(h,2) = 2 * W - matrix1(h,2);
    matrix1(h,4) = -1 * matrix1(h,4);

    h = matrix1(:,2) < 0;
    matrix1(h,2) = -1 * matrix1(h,2);
    matrix1(h,4) = -1 * matrix1(h,4);

    % spread out electrons
```

```
    h = rand(popSize, 1) < pScat;
    matrix1(h,3:4) = random(pdfVol, [sum(h),2]);

    % Record the ZE1
    semiCon = Mn/K/2/popSize; ab = sum(matrix1(:,3).^2); cd =
sum(matrix1(:,4).^2);
    ZE1(i) = (ab + cd)*semiCon;

    % Record positions of each electron movement thorughout the simulation
    for h=1:popPlot
        ZE(i, (2*h):(2*h+1)) = matrix1(h, 1:2);
    end

    % Update the simulation every 5 iterations
    if sim_OnOff && mod(i,5) == 0
        figure(3);
        hold on;
        plot(matrix1(1:popPlot,1)./1e-9, matrix1(1:popPlot,2)./1e-9, 'X');
        axis([0 L/1e-9 0 W/1e-9]);
        title(sprintf('Zeros for %d of %d Electrons (Part 2)',...
        popPlot, popSize));
        xlabel(' x-(nm)');
        ylabel(' y-(nm)');

    if i > 1
        figure (4);
        hold off;
        plot(time *(0:i-1), ZE1(1:i));
        axis([0 time *iter min(ZE1)*0.98 max(ZE1)*1.02]);
        title('Semiconductor Temperature changes for every zero (Part 2)');
        xlabel('Time (s)');
        ylabel('ZE1 (K)');
    end

    % histogram speeds
    figure (5);
    w = sqrt(matrix1(:,3).^2 + matrix1(:,4).^2);
    title('Histogram of Electron Speeds');
    histogram(w);
    xlabel('Speed (m/s)');
    ylabel('particles');
    pause(0.05);
    end
end

% Show Zeroes after the simulation is over
figure(3);
title(sprintf('Zeros for %d of %d Electrons (Part 2)',...
popPlot, popSize));
xlabel('x-(nm)');
ylabel('y-(nm)');
axis([0 L/1e-9 0 W/1e-9]);
hold on;

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

```
if(~sim_OnOff)
figure (4);
hold off;
plot(time *(0:iter-1), ZE1);
axis([0 time *iter min(ZE1)*0.98 max(ZE1)*1.02]);
title('Semiconductor Temperature changes for every zero (Part 2)');
xlabel('Time (s)');
ylabel('ZE1 (K)');
end

% Histogram intial electron speed
figure (5);
hold on;
title('Electron Speeds Histogram');
w = sqrt(matrix1(:,3).^2 + matrix1(:,4).^2);
histogram(w);
xlabel('Speed (m/s)');
ylabel('Number of particles');
```

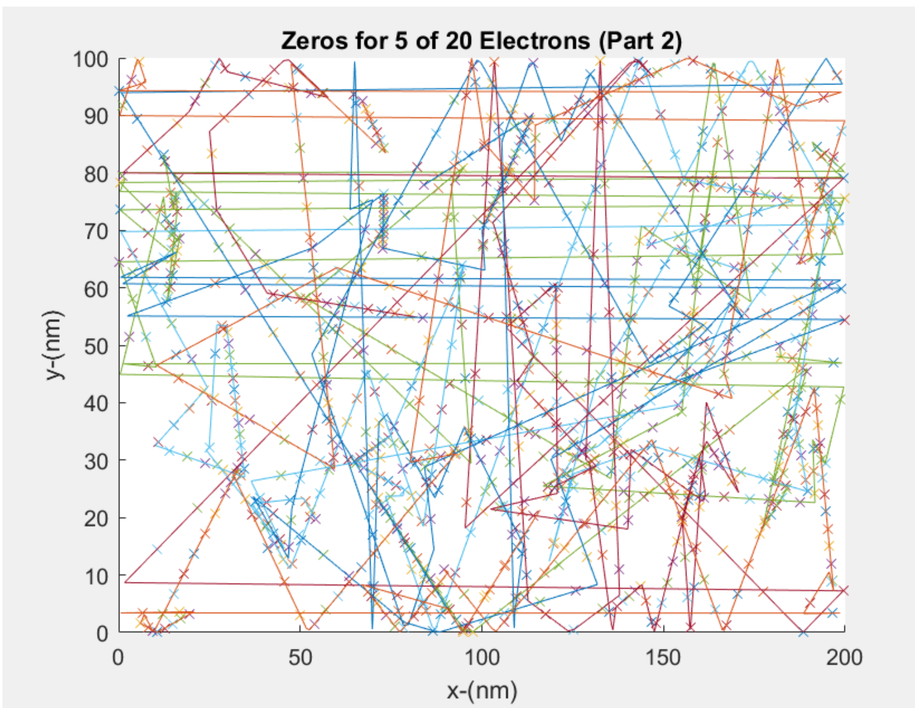

Figure 3: 2D plot of Particle Trajectory

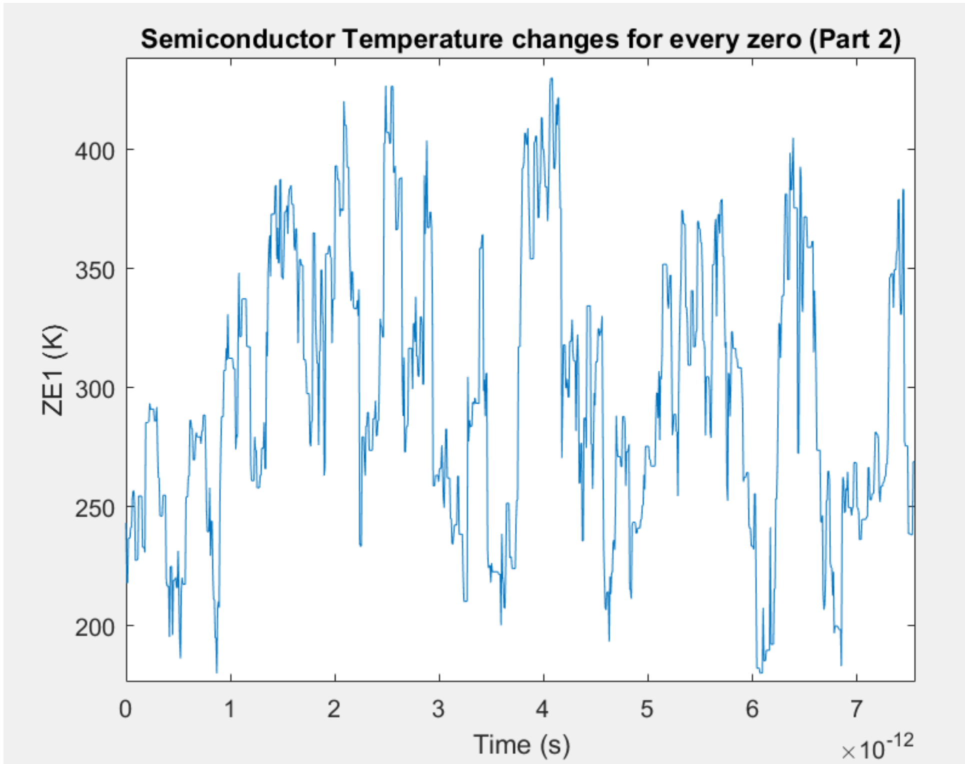

Figure 4: Temperature Plot

Q) What happens to the average temperature over time?

As seen in figure 4, the temperature varies widely but does not surpass 500K or goes below 100k. Over time the average temp becomes: 132.242 K.

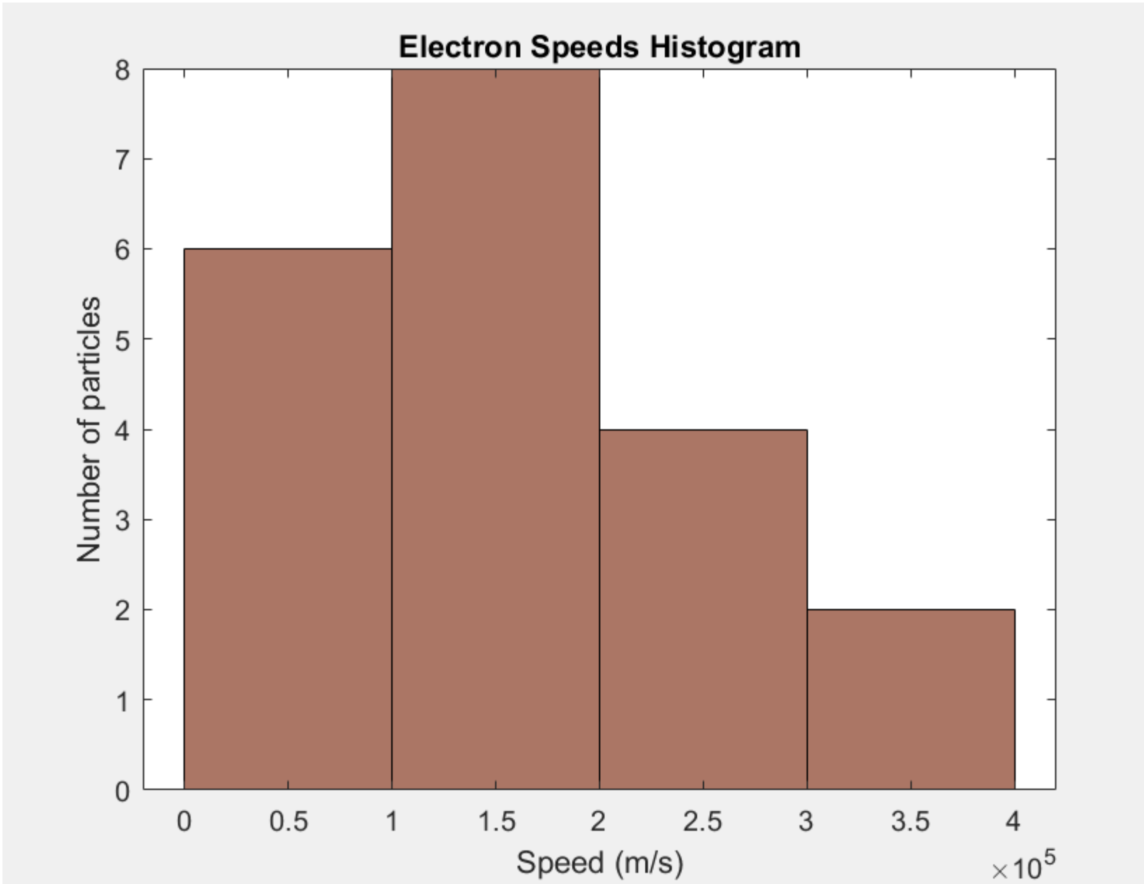

Figure 5: Histogram Plot of Particle Speeds

Question 3: Enhancements

This portion of the assignment adds in the inner rectangle "bottle neck" boundaries to model a similar particle behaviour as the previous part of the assignment. Making all boundaries capable to be either specular or diffusive.

Program Code:

```
%% Part 3: Enhancements

sqr = 1e-9.*[80 120 0 40; 80 120 60 100];
boxes = [0 1];

for i = 1:iter
    nextMat = time .*matrix1(:,3:4);
    matrix1(:,1:2) = matrix1(:,1:2) + nextMat;

    h = matrix1(:,1) > L;
    matrix1(h,1) = matrix1(h,1) - L;

    h = matrix1(:,1) < 0;
    matrix1(h,1) = matrix1(h,1) + L;
```

```
    h = matrix1(:,2) > W;
    topVal =0;
    bottomVal =0;


    if(topVal)
        matrix1(h,2) = 2*W - matrix1(h,2);
        matrix1(h,4) = -1*matrix1(h,4);
    else
    % The electron recoils off at random angle
    matrix1(h,2) = W;
    b = sqrt(matrix1(h,3).^2 + matrix1(h,4).^2);
    theta = rand([sum(h),1])*2*pi;
    matrix1(h,3) = b.*cos(theta);
    matrix1(h,4) = -1*abs(b.*sin(theta));
    end

    h = matrix1(:,2) < 0;

    if(bottomVal)
        matrix1(h,2) = -1*matrix1(h,2);
        matrix1(h,4) = -1*matrix1(h,4);
    else
    % The electron recoils off at a random angle
    matrix1(h,2) = 0;
    b = sqrt(matrix1(h,3).^2 + matrix1(h,4).^2);
    theta = rand([sum(h),1])*2*pi;
    matrix1(h,3) = b.*cos(theta);
    matrix1(h,4) = abs(b.*sin(theta));
    end

    % Scatter particles
    h = rand(popSize, 1) < pScat;
    matrix1(h,3:4) = random(pdfVol, [sum(h),2]);

    semiCon = Mn/K/2/popSize; ab = sum(matrix1(:,3).^2); cd =
sum(matrix1(:,4).^2);
    ZE1(i) = (ab + cd) * semiCon;

    % Record positions of subset particles that will be graphed
    for h=1:popPlot
        ZE(i, (2*h):(2*h+1)) = matrix1(h, 1:2);
    end

    % Update the simulation every 5 iterations
    if sim_OnOff && mod(i,5) == 0

    figure(6);
    hold on;
    plot(matrix1(1:popPlot,1)./1e-9, matrix1(1:popPlot,2)./1e-9, 'X');
    %hold on;

    % Plotting the square
    for h=1:size(sqr,1)
        plot([sqr(h, 1) sqr(h, 1) sqr(h, 2) sqr(h, 2) sqr(h, 1)]./1e-9,...
        [sqr(h, 3) sqr(h, 4) sqr(h, 4) sqr(h, 3) sqr(h, 3)]./1e-9, 'k-');
```

```
    end

    axis([0 L/1e-9 0 W/1e-9]);
    title(sprintf('Zeros for %d of %d Electrons (Part 3)',popPlot, popSize));
    xlabel('x - (nm)');
    ylabel('y - (nm)');

    if i > 1
        figure (7);
        hold on;
        plot(time *(0:i-1), ZE1(1:i));
        axis([0 time *iter min(ZE1(1:i))*0.98 max(ZE1)*1.02]);
        title('Semiconductor ZE1');
        xlabel('Time (s)');
        ylabel('ZE1 (K)');
    end

    figure (8);
    hold on;
    title('Electron Speeds Histogram');
    b = sqrt(matrix1(:,3).^2 + matrix1(:,4).^2);
    histogram(b);
    xlabel('Speed (m/s)');
    ylabel('Number of particles');
    pause(0.03);
    end
end

figure(6);
plot(matrix1(1:popPlot,1)./1e-9, matrix1(1:popPlot,2)./1e-9, 'X');
axis([0 L/1e-9 0 W/1e-9]);
title(sprintf('Zeros for %d of %d Electrons (Part 3)',popPlot, popSize));
xlabel('x - (nm)');
ylabel('y - (nm)');
hold on;

figure (7);
plot(time *(0:i-1), ZE1(1:i));
axis([0 time *iter min(ZE1(1:i))*0.98 max(ZE1)*1.02]);
title('Semiconductor ZE1');
xlabel('Time (s)');
ylabel('ZE1 (K)');
hold on;

figure (8);
hold on;
title('Electron Speeds Histogram');
b = sqrt(matrix1(:,3).^2 + matrix1(:,4).^2);
histogram(b);
xlabel('Speed (m/s)');
ylabel('Number of particles');
```

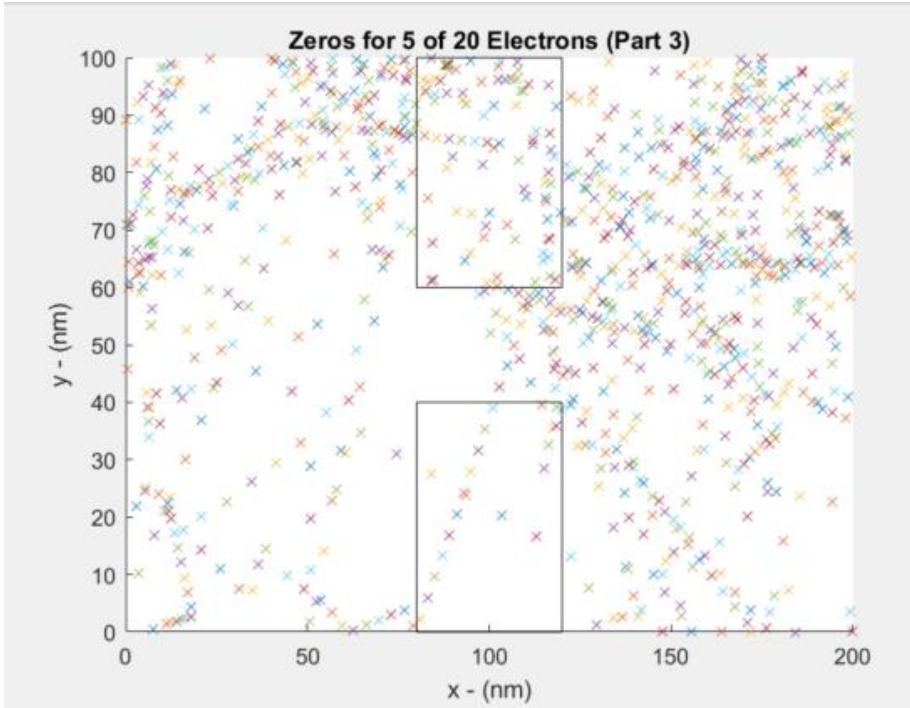

Figure 6: Particle Trajectory With "bottle-neck"

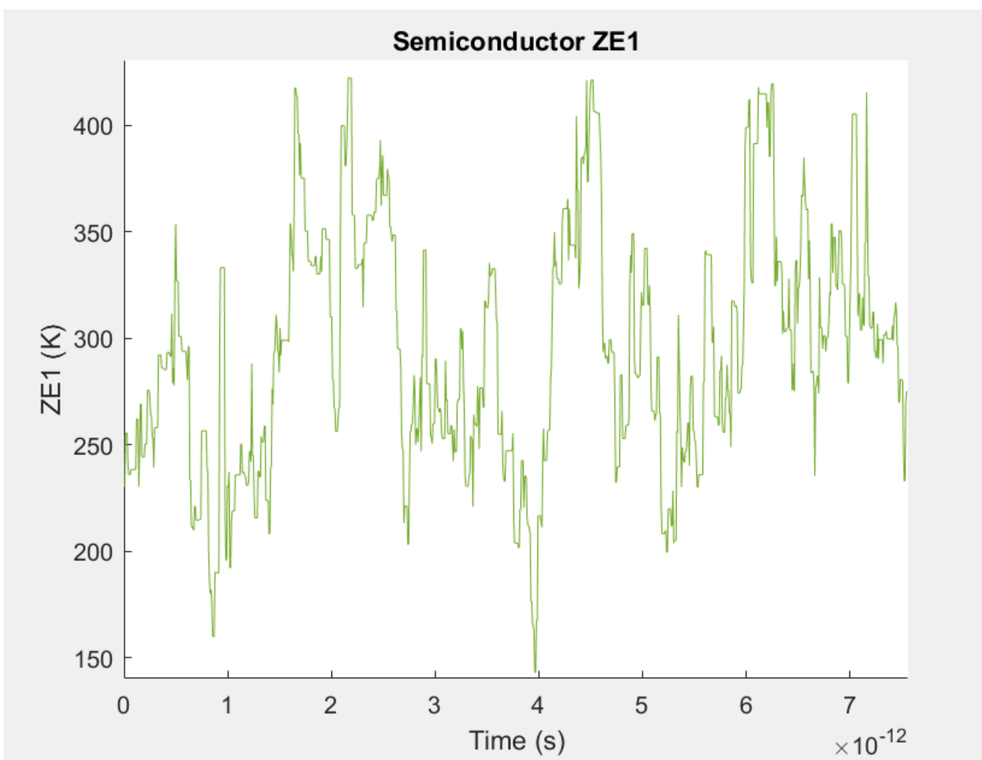

Figure 7: Temperature Plot

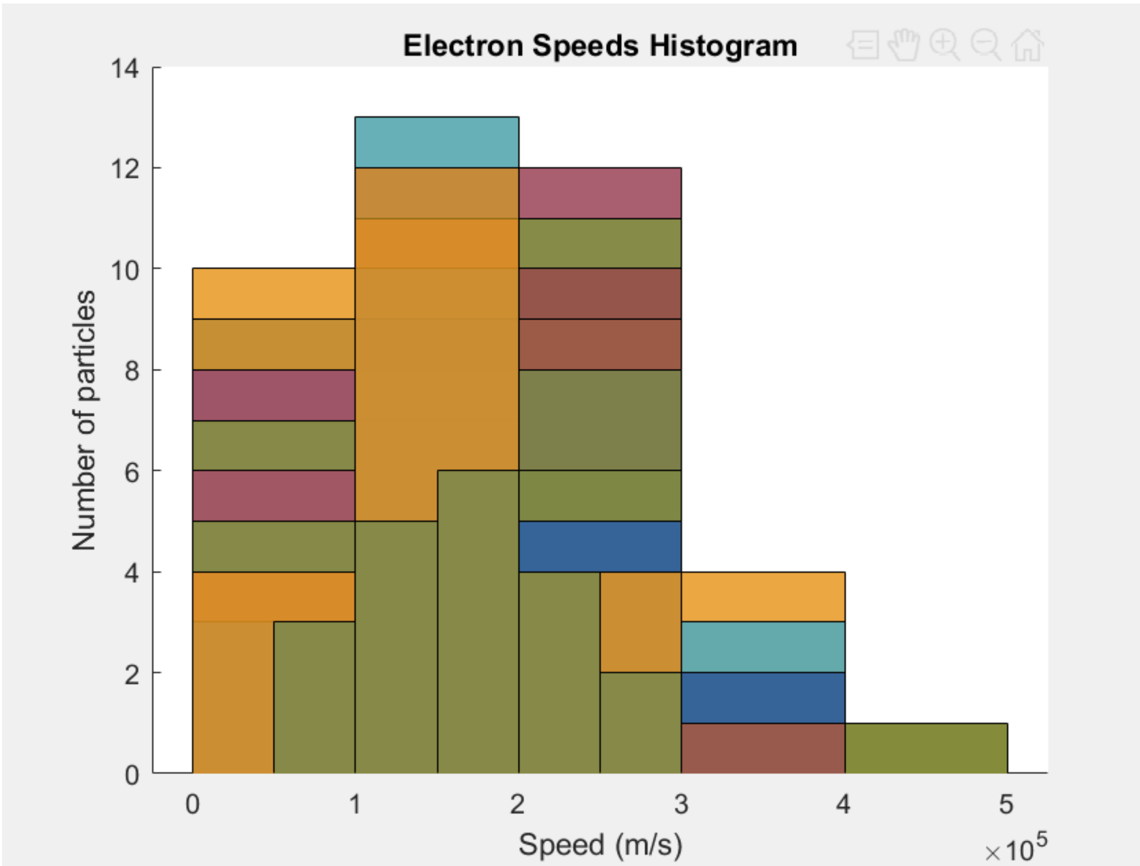

Figure 8: Histogram Plot of Particle Speeds