**ELEC 4700**

**Assignment - 1**

**Monte-Carlo Modeling of Electron Transport**

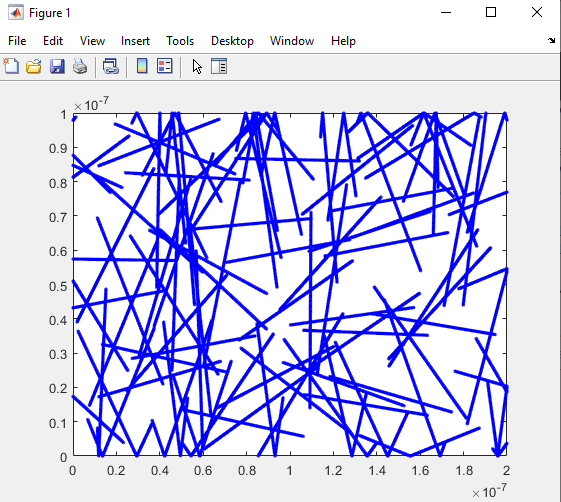
**Written by:**

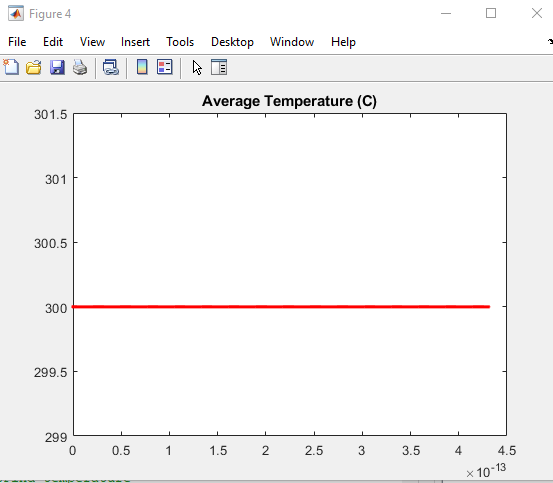
**Matthew Janok**

**101036060**

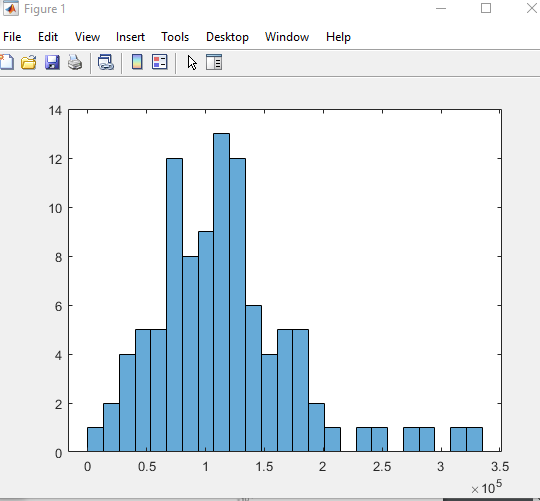
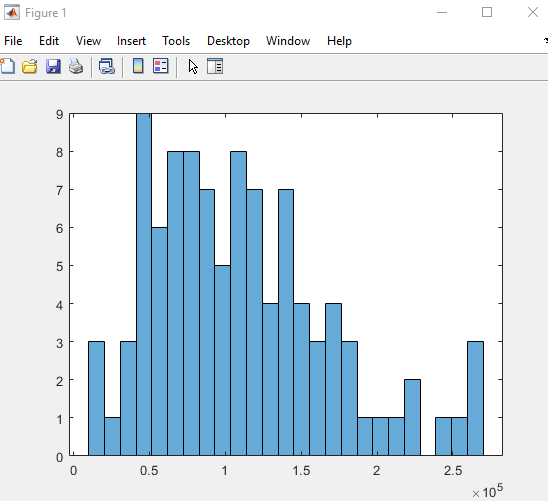
1. **Electron Modelling (40)**
2. vth = sqrt((kb\*T)/mn)

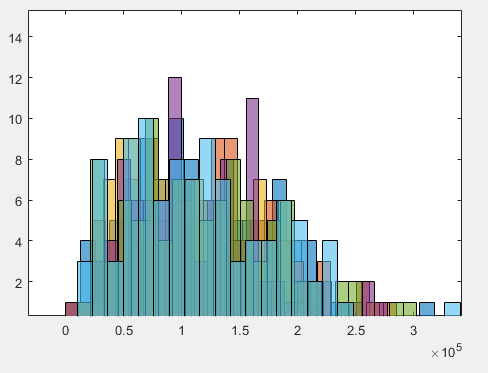
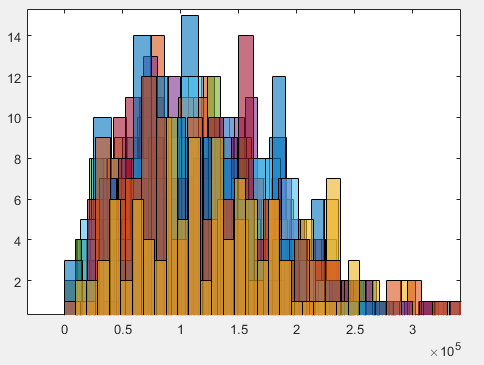
solving this equation in Matlab where T = 300 and kb = 1.381e-23 we get:

1. If the mean time between collisions is τmn = 0.2ps, then the mean free path can be found by multiplying the time by average velocity to get the distance of the mean free path. Using the average velocity as vth = 1.3225e+05 m/s, we find:
2. For code see appendix A

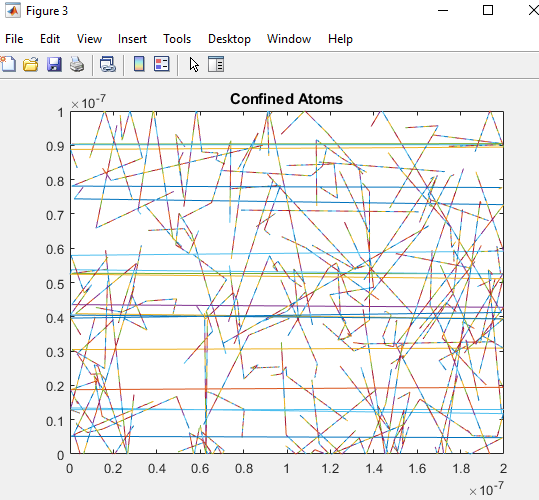
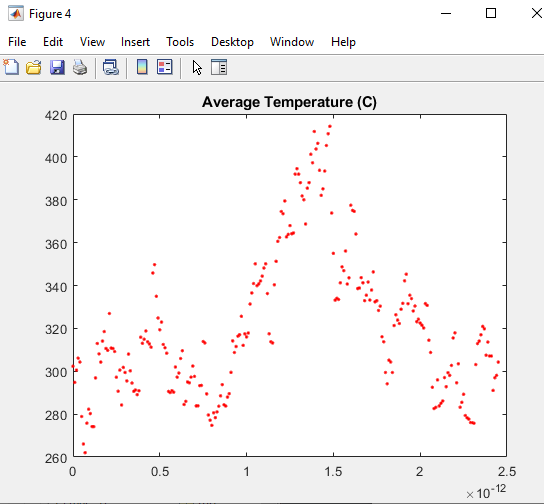
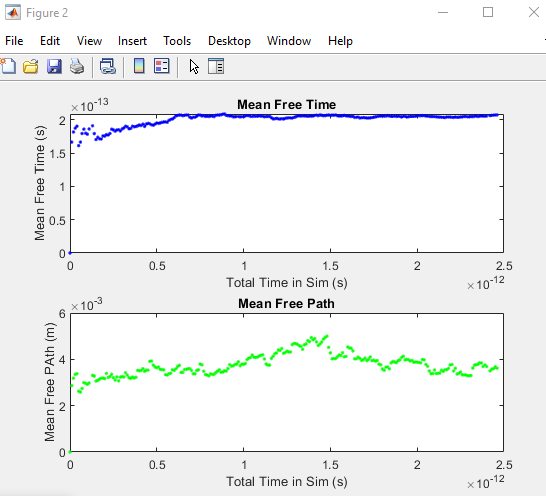


1. **Collisions with Mean Free Path (MFP) (25)**





The first two plots show the types of random velocities we see using the Maxwell-Boltzmann distribution. The two figures on the bottom show the random velocities as they iterate for new calculations. We can see from many iterations that this is indeed a Maxwell-Boltzmann distribution.

1. 
2. 
3. 
4. **Collisions with Mean Free Path (MFP) (25)**
5. Figure 3 shows the scattering atoms

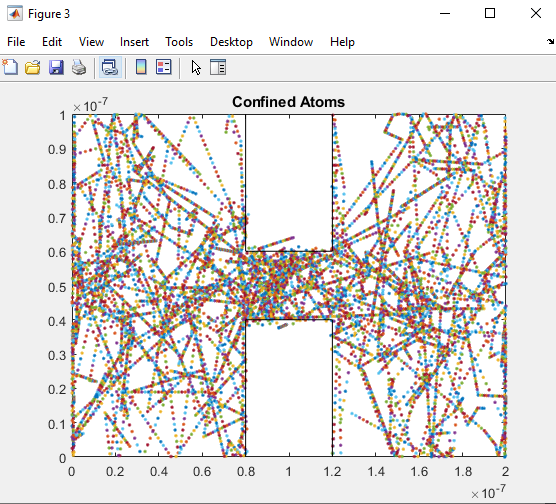
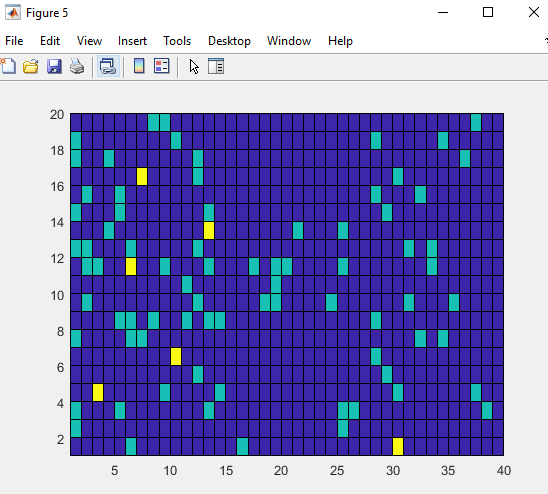
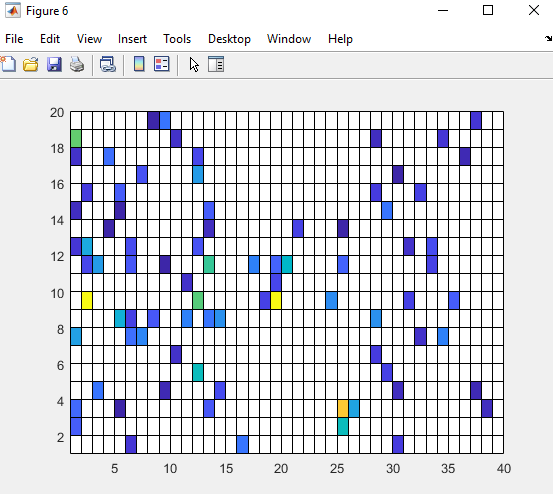


Figure 5 shows the Electron Density Map



1. Figure 6 shows the Temperature Map



APPENDIX A

clc

clear

mo = 9.11e-31;

mn = 0.26\*mo;

kb = 1.381e-23;

T = 300;

%Initialise the particles

initialX = 200e-9\*rand(100,1);

initialY = 100e-9\*rand(100,1);

axis ([0 200e-9 0 100e-9])

%Initialise angles

angleRad = 2\*pi\*rand(100,1);

%Set velocity

vth = sqrt((kb\*T)/mn);

velocityX = vth.\*cos(angleRad);

velocityY = vth.\*sin(angleRad);

for time = 0:1e-15:0.01

%Find new positions

newX = initialX + velocityX\*1e-15;

newY = initialY + velocityY\*1e-15;

%Find temperature

Vavg = mean((velocityX.^2) + (velocityY.^2));

T = (mn\*Vavg)/(kb);

%Check X boundary conditions

[NH,IH] = max(newX);

[NL,IL] = min(newX);

upperX = newX > 200e-9;

newX(upperX)= newX(upperX)-200;

lowX = newX < 0;

newX(lowX) = newX(lowX)+200;

%Check Y boundary conditions

[NumH,IndexH] = max(newY);

[NumL,IndexL] = min(newY);

upperY = newY > 100e-9;

velocityY(upperY)= -velocityY(upperY);

lowY = newY < 0;

velocityY(lowY) = -velocityY(lowY);

initialX = newX;

initialY = newY;

figure(1)

plot(newX,newY, 'b.')

hold on

axis ([0 200e-9 0 100e-9])

figure(4)

title('Average Temperature (C)');

plot(time, T, 'r.')

hold on

pause(0.01)

time

end