Jacob Podhraski ELEC 4700 2018-02-04
100970231 Assignment 1

1.0 Electron Modelling

The thermal velocity is easily calculated with the equation;

$$V_{th} = \sqrt{\frac{k_b T}{m}}$$

With m, k_b, and T all constant the thermal velocity is;

$$V_{th} = 1.8697e{+}05 \text{ m/s}$$

From this all electrons were assigned a velocity of V_{th} and were all constant. After the simulation was run this was an example of an output shown in figure 1 below.

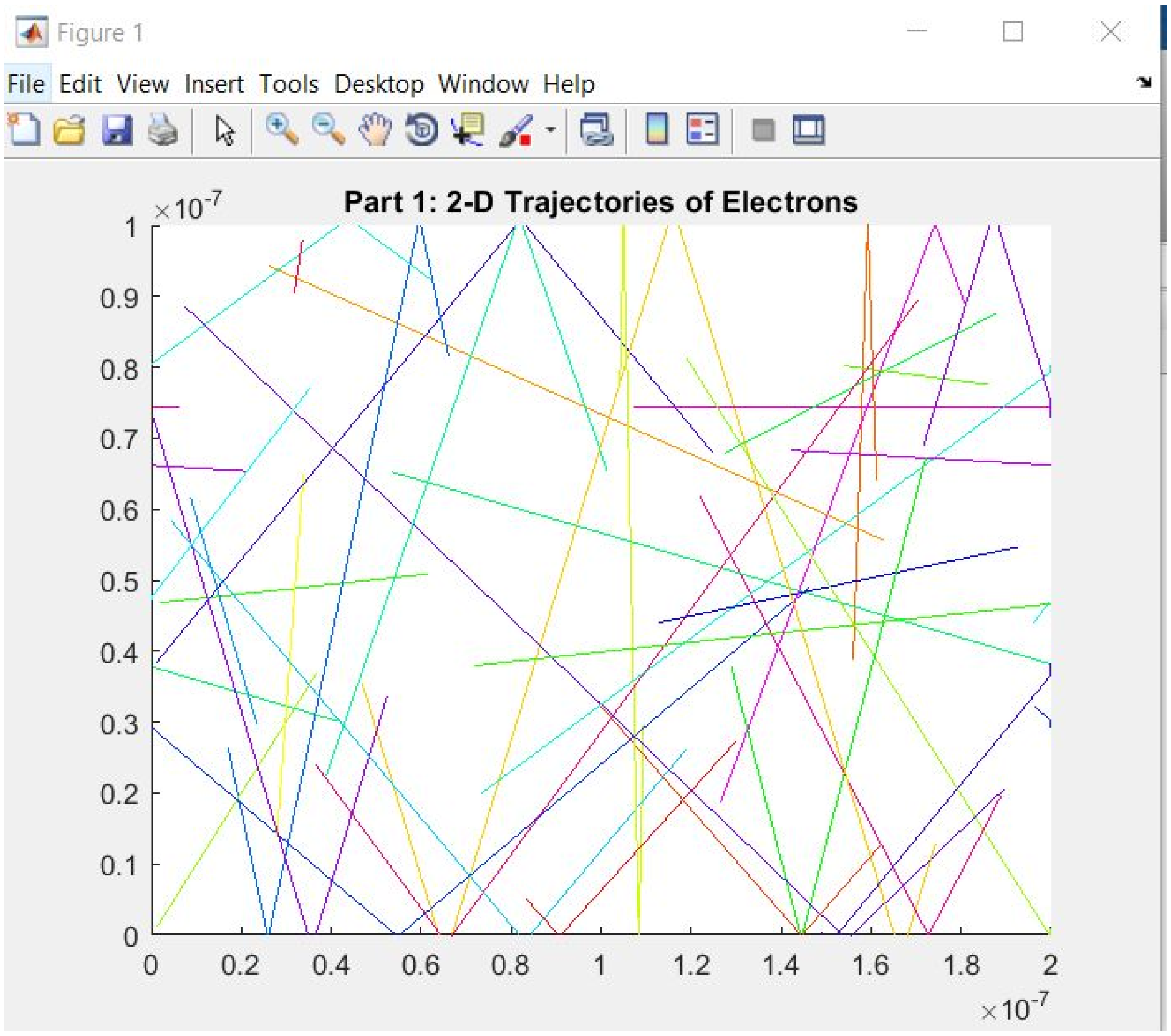

Figure 1: Simulation of Part 1

To run this simulation each particle was assigned an initial x and y position, a direction, and a velocity. An example of this code is shown below.

```
while i<=30
```

```
  j = 1;
  while j<=4


      if(j==1)
          randx = rand(1,1);
          init(i,j)= randx * (200*10^-9); %assign random x position
      elseif(j==2)
          randy = rand(1,1);
          init(i,j) = randy * (100*10^-9); %assign random y position
      elseif(j==3)
          randd= rand(1,1);
          init(i,j) = randd*2*pi;
      else
          randvx = randn(1,1);
          vx = (v/sqrt(2))*randvx;
          randvy = randn(1,1);
          vy = (v/sqrt(2))*randvy;
          vth = sqrt(vx^2+vy^2);
          init(i,j) = vth;
      end
    j = j + 1;
     end
  i = i +1;
end
```

Then the code was stepped through a for loop to run the total simulation, which plotting the current position and also the previous position to create a trace. Furthermore, a plot of the average temperature over the duration of the simulation was generated to see if the temperature remained constant throughout the duration of the simulation. This is shown below in figure 2.

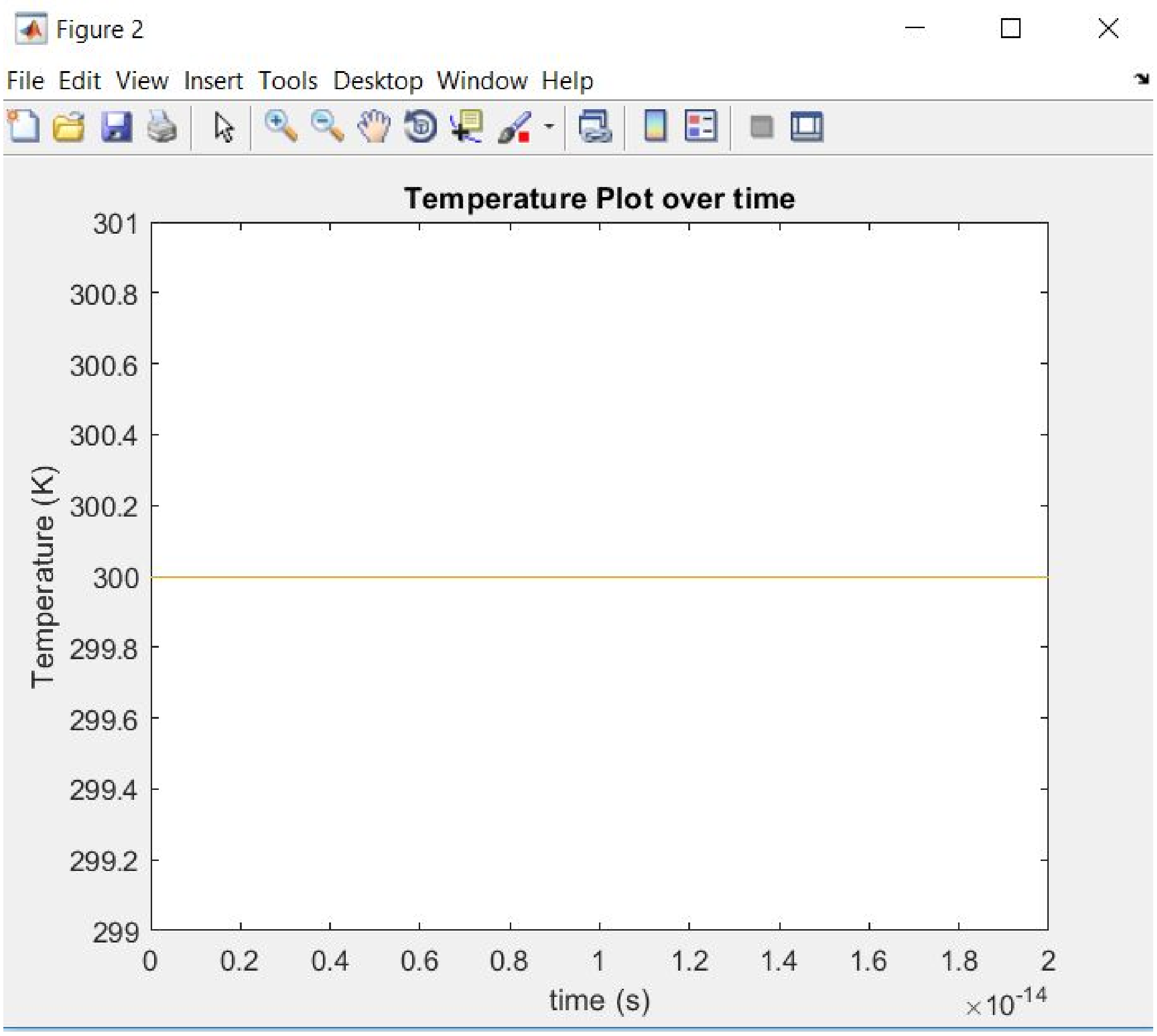

Figure 2: Part 1 Temperature plot

As expected, the temperature remained constant over the duration of the simulation with a constant velocity applied to each particle.

2.0 Collisions with Mean Free Path (MFP)

Part 2 required another simulation similar to part 1 except with a probability of scattering the electrons. This was done by using an exponential scattering probability formula;

$$P_{scat} = 1 - e^{\frac{-dt}{tmn}}$$

Each iteration of the loop would check if the each particle would scatter. This was implemented in an if statement as shown in the following code.

```
if pscat > rand()
        randnew = rand(1,1);
        init(j,3) = randnew * 2 * pi;
        randvx = randn(1,1);
         vx = (v/sqrt(2))*randvx;
         randvy = randn(1,1);
         vy = (v/sqrt(2))*randvy;
         vth = sqrt(vx^2+vy^2);
         init(j,4) = vth;
    end
```

After running the simulation, the particles behaved as expected. The output of the simulation is shown below in figure 3.

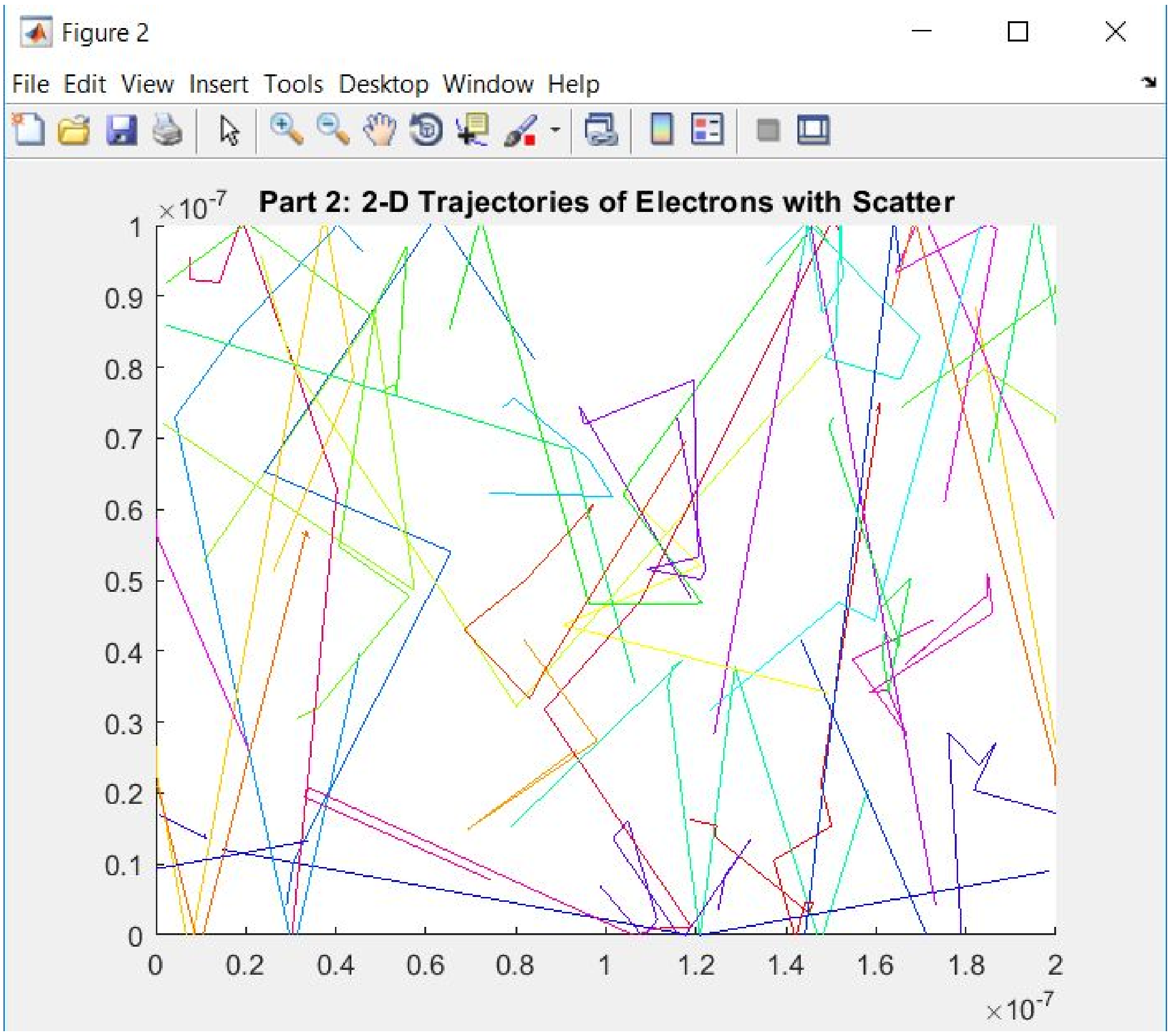

Figure 3: Part 2 simulation

As shown in figure 3, we can see the scattering in the particles. Also, if the particle scatters, the particle is assigned a new random direction and a random velocity from the Maxwell-Boltzmann distribution. The output of the distribution is shown below in a histogram in figure 4.

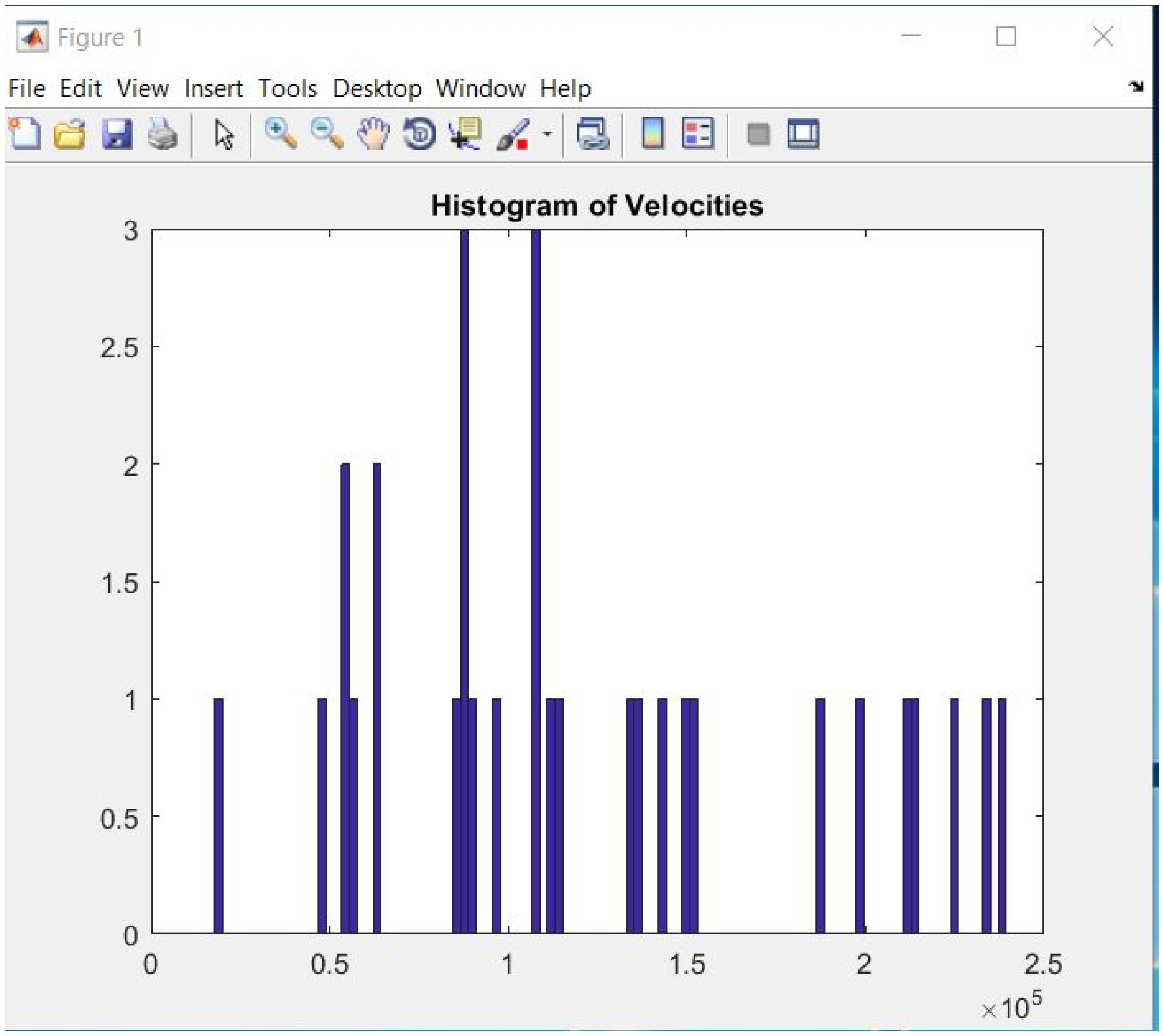

Figure 4: Histogram of velocities

This histogram represents the probability of the particles velocities in the simulation. We can see it has the general shape of a Maxwell-Boltzmann distribution.

3.0 Enhancements

The final part of the assignment is to place boundaries in the simulation to see how the particles behave when they encounter a boundary. The result of the simulation is shown below in figure 5.

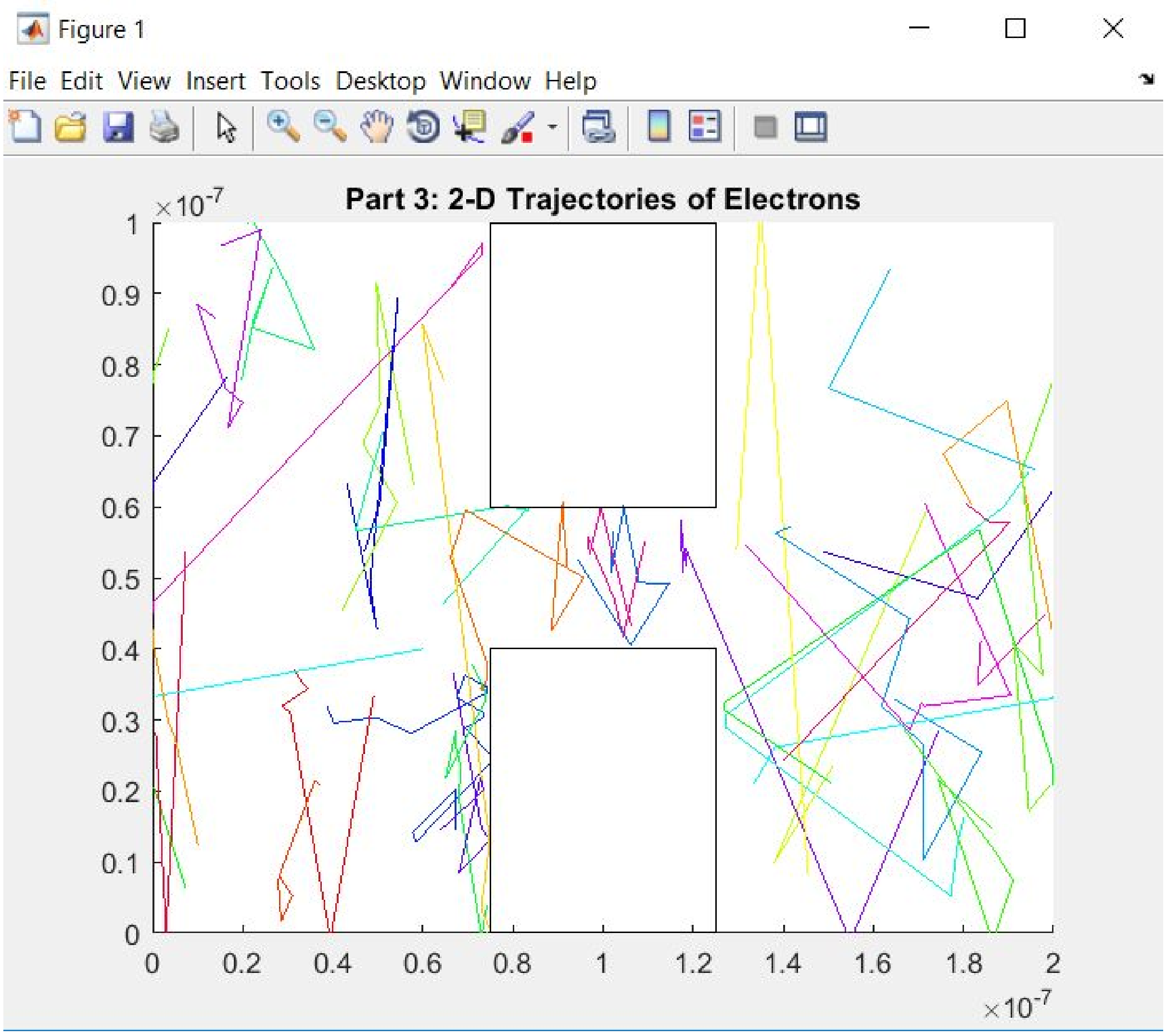

Figure 5: Part 3 simulation