

ELEC 4700
Assignment 1
Monte-Carlo Modeling
of Electron Transport

Student Name: Jeremy Schelhaas

Student Number: 100976020

Date: Feb 3, 2019

Table of Contents

Part 1: Electron Modeling.....	3
Part 2: Collisions with Mean Free Path.....	5
Part 3: Enhancements.....	7

Figures

Figure 1: Electron Modelling and Temperature.....	5
Figure 2: Maxwell-Boltzmann Distribution of Thermal Velocities.....	6
Figure 3: Electron Modelling with Scattering and Temperature.....	7
Figure 4: Electron Modelling and Temperature after Enhancements.....	8
Figure 5: Electron Density.....	9
Figure 6: Temperature Distribution.....	10
Figure 7: Simulation Figures.....	11

Part 1: Electron Modelling

There are many different ways to model the behavior of electrons in a material. One of the most common ways is to do this through a Monte-Carlo Simulation. This is done by determining the behavior of a single electron anywhere in the surface, plotting the required behavior, and then inserting another hundred or thousand electrons and observing the behavior of all of the different electrons simultaneously. For the purposes of keeping the explanations as simple as possible, a single electron's behavior was determined, and then 9,999 others were extrapolated from the surface conditions.

Each electron has a given energy due to having some given temperature, and due to the energy, it has a velocity with which it travels through a medium. This comes from the fact that energy is a force, and since electrons have mass, the force being applied to a mass gives it acceleration. If the effective mass of the electron is 0.26 times its rest mass, and the medium that the electron is in has a temperature of around room temperature, or 300K, the thermal velocity of that electron can be determined through the following equation:

$$V_{th} = \sqrt{\frac{kT}{m_n}}$$

where V_{th} is the thermal velocity, k is Boltzmann's constant, T is the temperature, and m_n is the effective mass of the electron. After substituting values into the variables, the thermal velocity of the modeled electron is 1.323×10^5 m/s. If we assume that electrons can bounce off different objects or surfaces, a mean time between collisions, or τ , can be determined, and thus a mean free path, which is the mean distance an electron can travel before it collides with something. If we assume the mean time between collisions of $\tau = 0.2ps$, the mean free path can be calculated by multiplying the time between collisions and the thermal velocity of the electrons. By using the values above, mean free path is 26.45nm.

Now a Monte-Carlo simulation can be modelled. First, the behavior of a single electron in a surface is modelled. This electron has a random x and y coordinate within the surface region, as well as a given thermal velocity and angle to travel in. At a given time interval, the position of the electron, both the x and y coordinates, get updated based on the velocity of the electron using the following equation:

$$x_f = x_i + t * v_x$$

where x_f is the final position of the electron in the x direction, x_i is the initial position of the electron in the y direction, t is the time duration of the time step, and v_x is the velocity of the electron in the x direction. This calculation is repeated for the y coordinate to obtain the full new coordinate of the electron.

Next, some boundary conditions are set up for the region. The modelled surface is 200nm wide and 100nm high. If the electron travels past the top or bottom side of the region, where the

x coordinate does not matter, but the y coordinate is 100nm or 0nm respectively, the angle that the electron is travelling at will be inverted along the x axis, meaning the velocity in the y direction is inverted, but stays at the same value. If the electron travels past the left or right side of the region, where the y coordinate does not matter, but the x coordinate is 0nm or 200nm respectively, the electron will continue to travel, but will be “transported” to the other side of the region.

Once this modelled behavior is correctly implemented for the single electron, the rest of the 10,000 electrons can be modelled. If implemented correctly, all 10,000 electrons should behave similarly to the single electron around all boundary conditions.

Drift velocity is the instantaneous average velocity of all electrons in a region. Since each electron has an x velocity and a y velocity, the two velocity components can be combined to produce the overall velocity of the electron. This is done through the following equation:

$$V = \sqrt{V_x^2 + V_y^2}$$

where V is the velocity of the electron, V_x is the x velocity of the electron, and V_y is the y velocity of the electron. The temperature of the electrons in the area can be calculated by taking the mean value of the velocity squared, multiplied by the effective mass of the electrons and divided by Boltzmann’s constant, as shown below:

$$temperature = \frac{mean(V^2) * m_n}{k}$$

Once the temperature of the region has been calculated, both the electrons moving around in the region as well as the temperature can be plotted for each time step to see how the electrons move and the temperature changes. These plots are shown below, where the first six electrons that were generated are being plotted, and the temperature of all 10,000 electrons is plotted below:

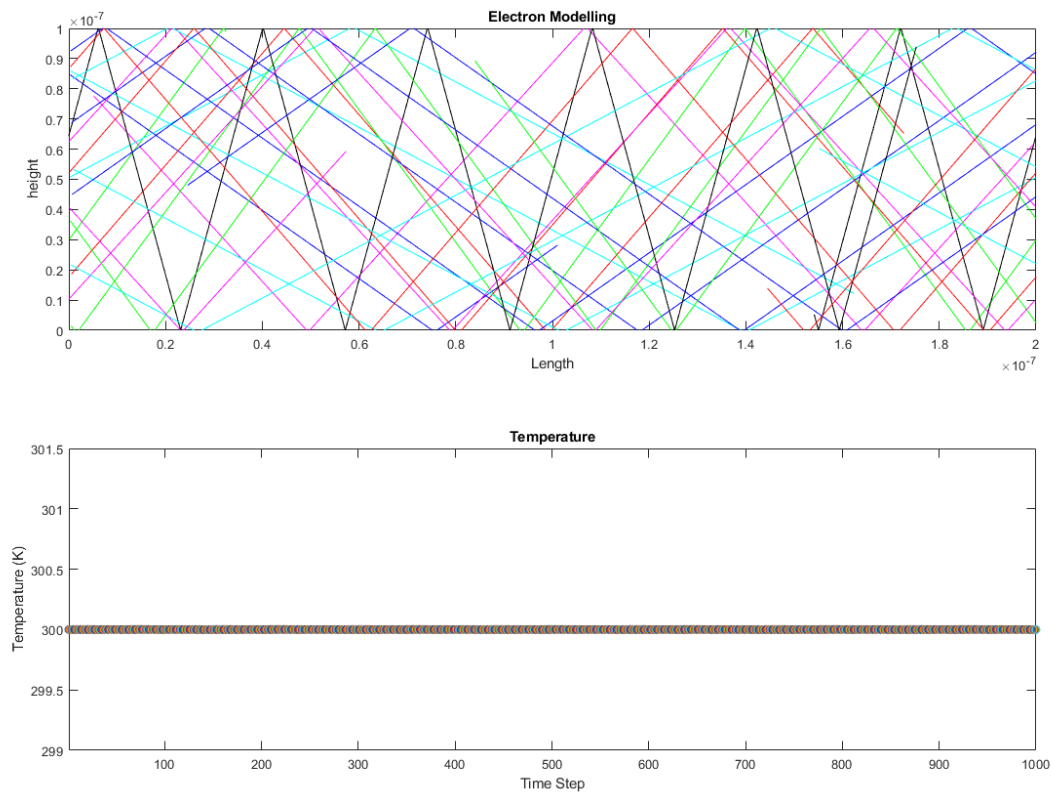


Figure 1: Electron Modelling and Temperature

Part 2: Collisions with Mean Free Path (MFP)

Once the basic Monte-Carlo simulation has been implemented, some changes can be made. First, the thermal velocity can be changed from being directly related to the manipulation of the initial thermal velocity with a random angle to being picked from a Maxwell-Boltzmann distribution for both the x and y velocity of each electron. The Maxwell-Boltzmann distribution will assign random values to the x and y velocities centered around the thermal velocity. The magnitudes and direction of each velocity component will determine the overall velocity of the electron. The Maxwell-Boltzmann Distribution is plotted below with the thermal velocity with it:

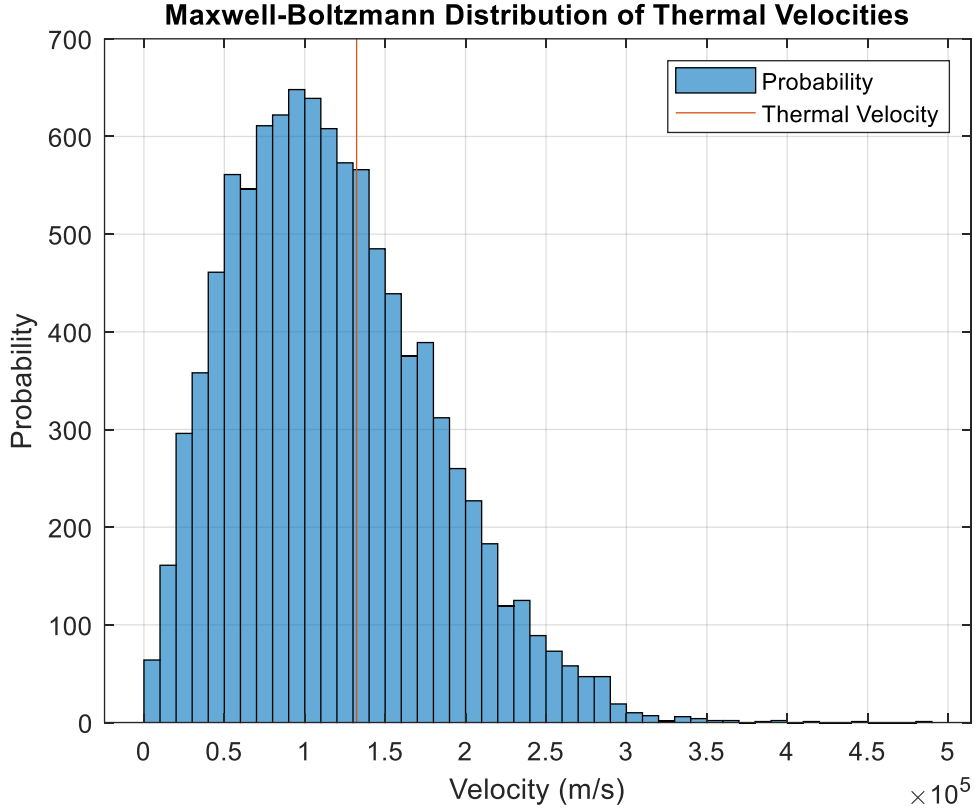


Figure 2: Maxwell-Boltzmann Distribution of Thermal Velocities

Next, scattering can be implemented. Scattering is random for all electrons, but the probability of scattering is based off of the following equation:

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

where P_{scat} is the probability to scatter, dt is the time since the last time step and τ_{mn} is the mean time between collisions. For every time step in the simulation, for every electron, this value is compared to a random number, and if the probability to scatter is greater than the random generated number, the velocity in both the x and y direction are reassigned based on the distribution above.

Due to the velocities of the electrons randomly changing because of scattering, the temperature of the region will fluctuate randomly. The calculation for the temperature of the region is the same as in Part 1, but now the velocities of a single are not exactly identical to every other electron in the simulation. Once the scattering and temperature calculations have been completed, the simulation can be run again, which is shown below, where the temperature of all electrons at every time step is represented as the red line, and the overall average temperature from the entire simulation is represented by the blue line:

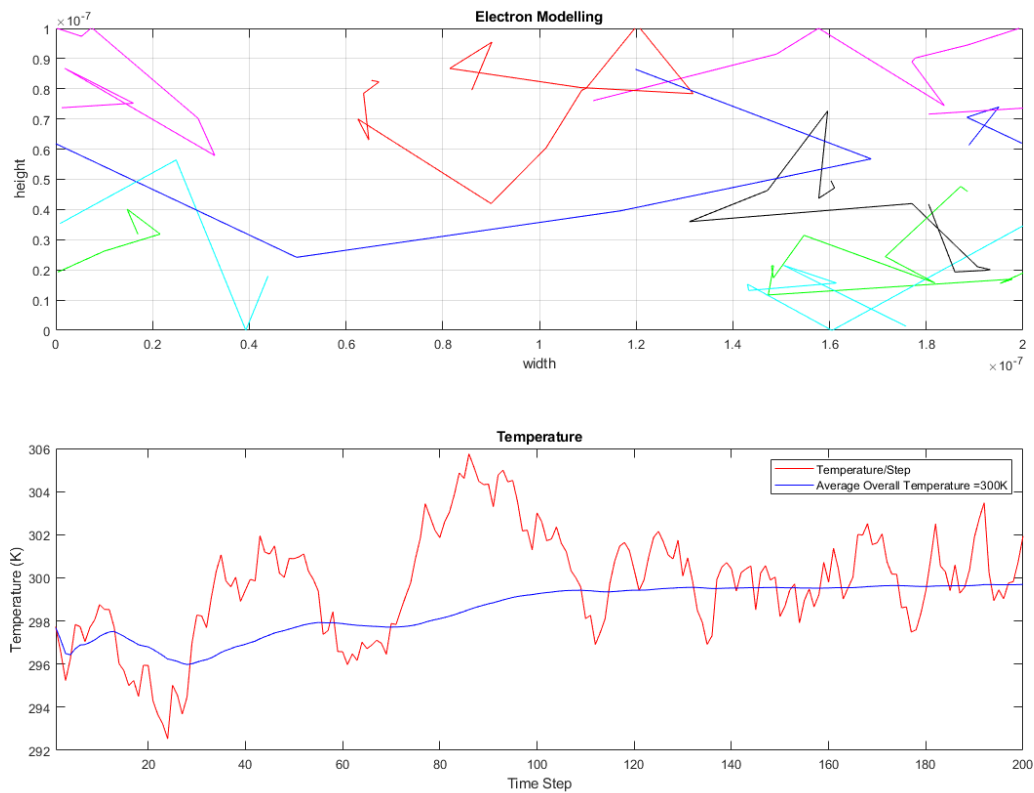


Figure 3: Electron Modelling with Scattering and Temperature

As shown above in the temperature graph, the temperature fluctuates a lot over time, but the average temperature is centered around 300K. The mean free path and mean time between collisions can also be calculated based on the scattering of electrons, and determined to be 27nm and 0.2ps respectively.

Part 3: Enhancements

Once scattering and temperature fluctuating has been implemented, some enhancements can be done to the region. First, two boxes can be added to the region where electrons cannot start in the boxes. This is implemented by logically indexing through the electron position vectors to find electrons that start in each box. For this simulation, all electrons that start in the bottom box will have their x positions moved 40nm to the left of the box and electrons that start in the top box will have their x positions moved 40nm to the right of the box. This ensures that before the simulation even starts, any electrons that randomly spawn in the boxes will be moved outside of the boxes.

Next, the boundary conditions corresponding to each side of the box were implemented. If electrons collide with any side on either box, they will be reflected off of the sides. With each time step, the new x and y coordinate for each electron was calculated. If the old coordinate is on one side of the box boundary and the new coordinate is on the other side, the velocities will be inverted. If the electron crosses the left or right side of either box, the x velocity of the given electron will be inverted. If the electron crosses the top of the bottom box or the bottom of the top box, the y velocity of the given electron will be inverted. These boundary conditions and the temperature of the region for every time step are shown below:

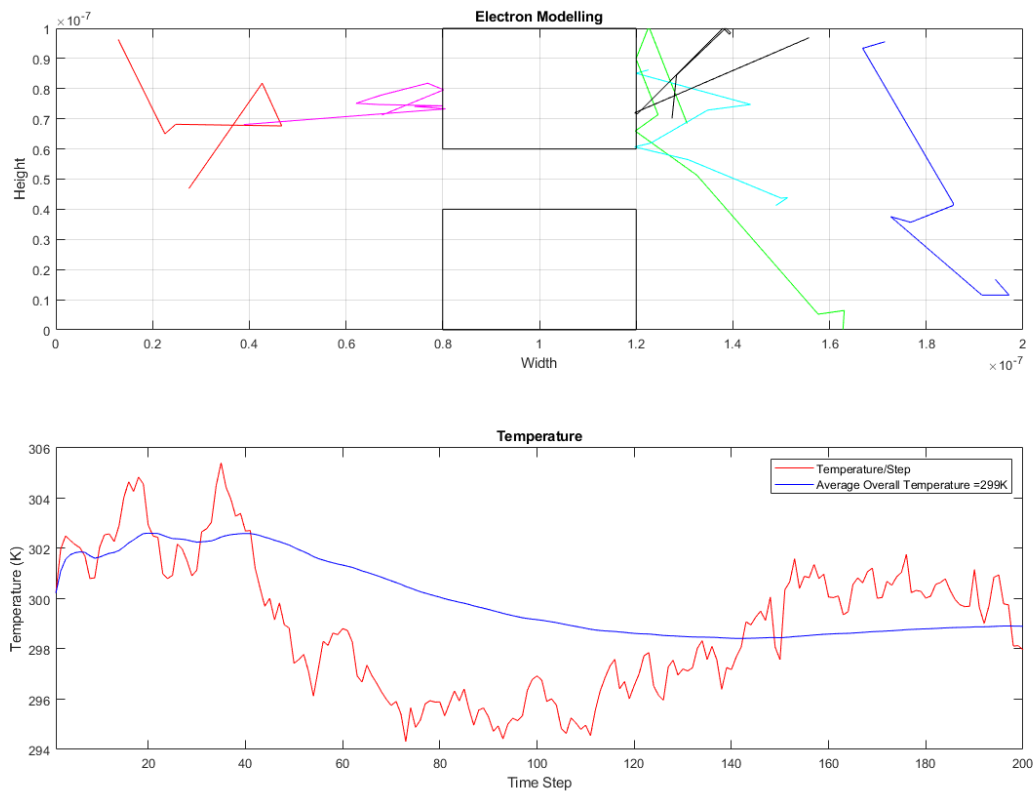


Figure 4: Electron Modelling and Temperature after Enhancements

As each time step is simulated, the region is separated into many smaller regions so that the density of electrons in each small region can be determined and plotted. This can show how all electrons travel throughout the large region. The following graph shows the electron density through the region after the simulation is finished:

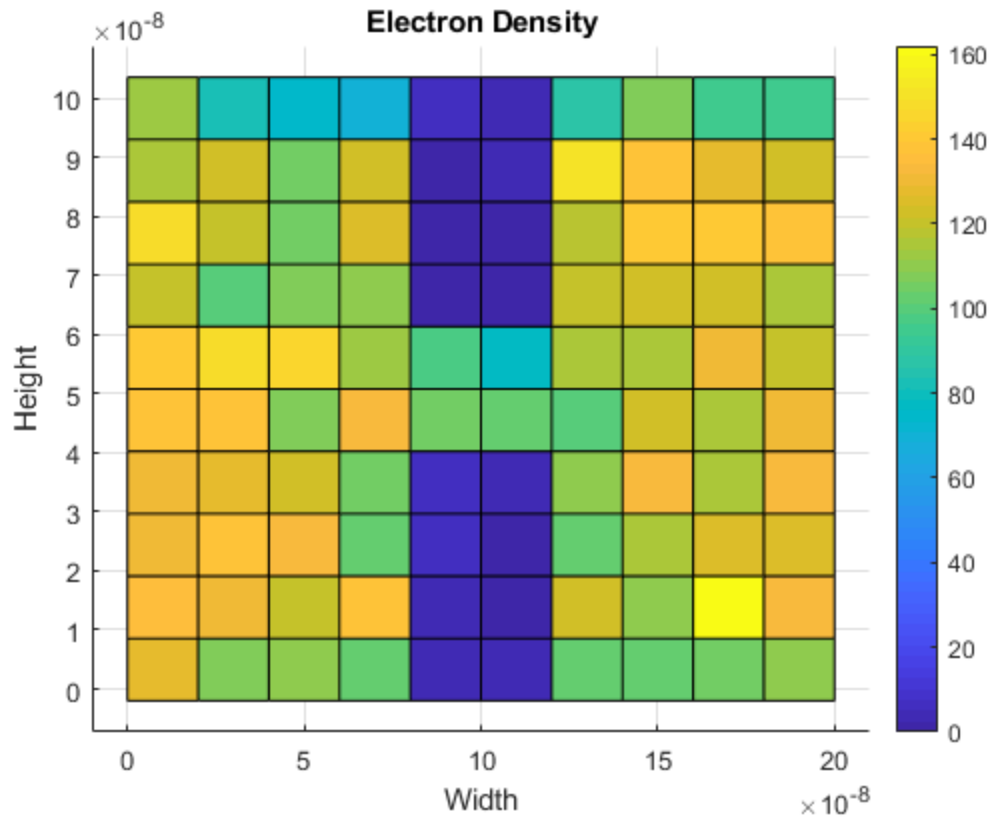


Figure 5: Electron Density

Once the density has been determined, the temperature in the region can be calculated in the same way. The large region is split up into many smaller regions and the velocities are used to calculate the temperature in each region. The temperature calculations are done in the same way as in Part 1 and 2. The temperature distribution after the simulation is shown below:

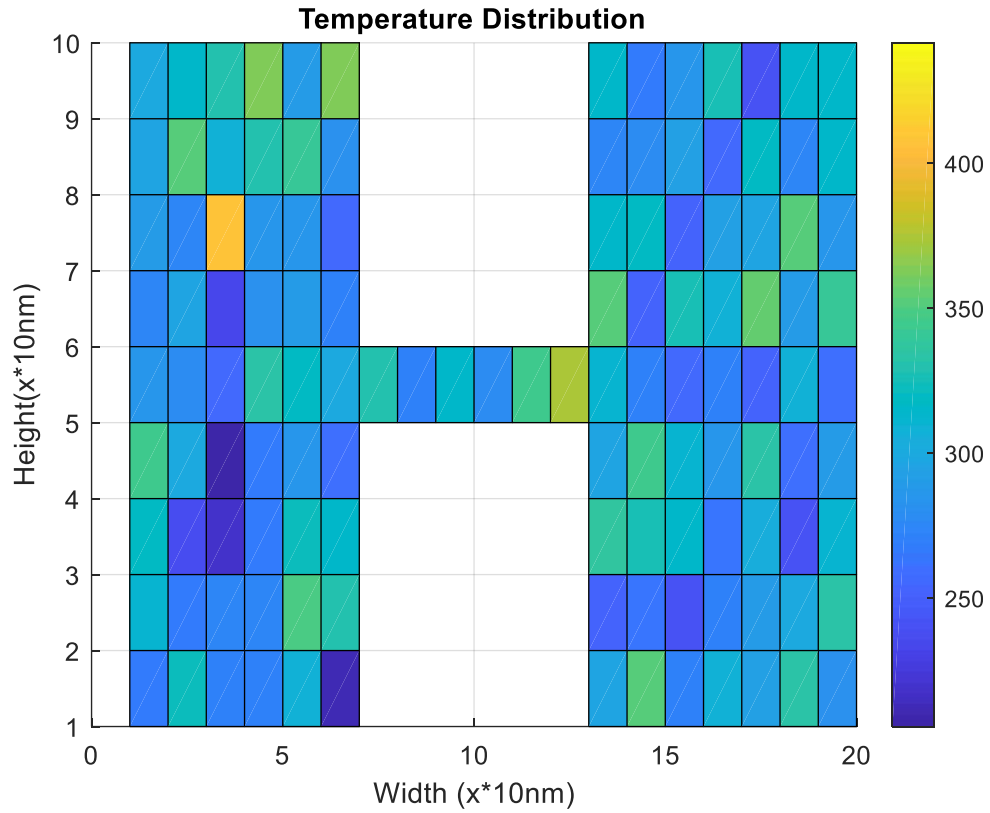


Figure 6: Temperature Distribution

Once all of the different graphs have been produced, they can be combined and shown in a single image so all available information can be displayed at once:

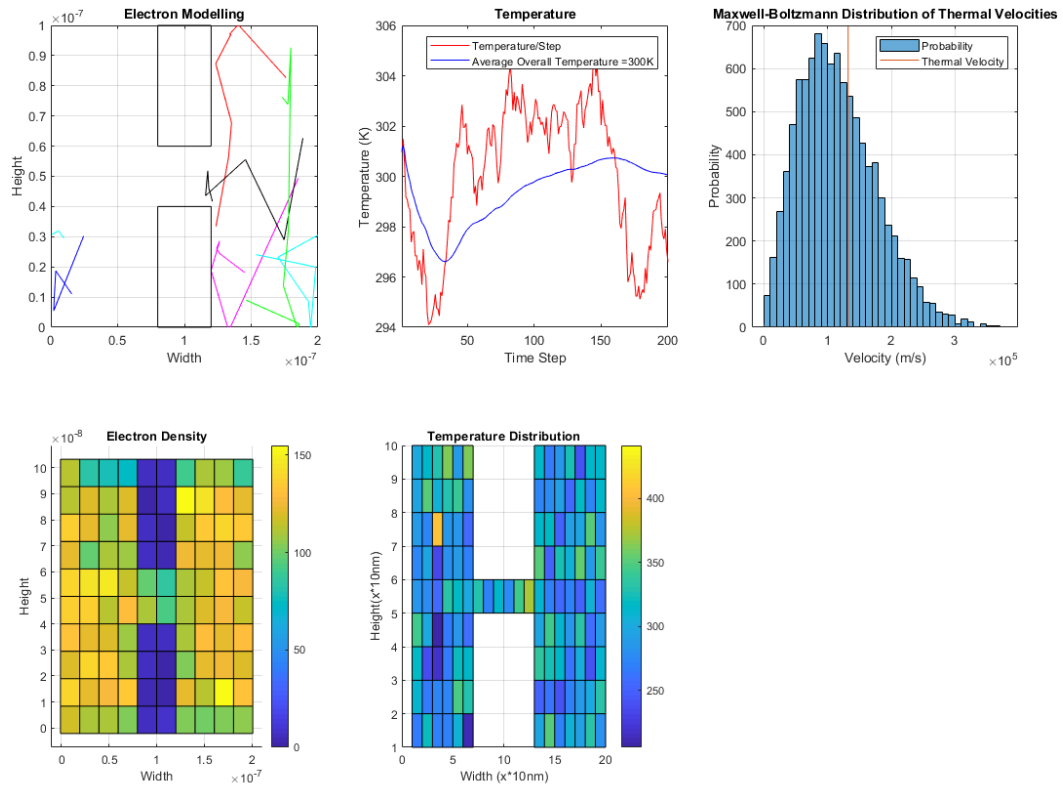


Figure 7: Simulation Figures