ELEC 4700

Assignment 1

Monte Carlo Modeling of Electron Transport

Brandon Schelhaas
101036851

February 7, 2021

Prof. Tom Smy

Part 1: Electron Modeling

Modeling electrons may be accomplished with several different methods, but of such methods, two popular ways are to model singular electrons as singular electrons, or model singular electrons that represent many electrons. In this assignment, the behaviour of 5000 electrons were modeled by specifically observing 10 electrons – both of which parameters are user-definable. A Monte Carlo model was defined by initializing each electron with a random position in 2D space. From here, the behaviour of these electrons was observed, given a thermal velocity as a force on the electrons.

Given a non-zero temperature, each electron has some energy due to the temperature, which causes the electrons to have a velocity – denoted as the thermal velocity. To add to the Monte Carlo model, each electron was given a random direction, but of which, the magnitude of the components was still equal to the thermal velocity. This behaviour causes each electron to have the same speed, but travel in different directions. The thermal velocity requires some theory of partitioning energy to calculate. Specifically, the relationship of $\frac{1}{2}mv^2 = \frac{1}{2}kT$ only holds true if we partition $\frac{1}{2}kT$ to each degree of freedom. Since the simulation is in 2D (x and y dimensions), the amount of energy is equal to $2(\frac{1}{2}kT)$. Therefore, the thermal velocity is extracted from the following relationship:

$$\frac{1}{2}mv_{th}^2 = 2\left(\frac{1}{2}kT\right)$$

$$v_{th} = \sqrt{\frac{2kT}{m}}$$

where k is Boltzmann's constant, T is the temperature (in Kelvin), and m is the effective mass of the electron (equal to 0.26*m_e). With this equation, the thermal velocity, at 300K, was calculated to be:

$$v_{th} = \sqrt{\frac{2kT}{m}} = \sqrt{\frac{2(1.38x10^{-23})(300)}{0.26(9.11x10^{-31})}} = \mathbf{1.8702x10^5\ m/s}$$

The next important system characteristic to model is the mean free path. This concept comes from the fact that electrons can collide with objects, such as other electrons, or boundaries. Therefore, the mean free path indicates the mean distance an electron may travel before a collision. Given a mean time between collisions of $\tau_{mn} = 0.2ps$, the mean free path, λ, is calculated as follows:

$$\lambda = \tau_{mn}v_{th} = (0.2ps)(1.8702x10^5\ m/s) = \mathbf{37.404\ nm}$$

A simulation was then setup to showcase the electrons' behaviours. While all settings are user-definable, the simulation was setup to simulate 200 time steps, where each time step was set to ensure the electron only moved 1/100 of the 2D space's length of 200nm. Specifically, this was accomplished with the following calculation, which forces the specific change of position and is all dependent on the region's length and the thermal velocity.

$$dt = \frac{0.01 * regionLength}{v_{th}}$$

Once the simulation loop starts, a new position is calculated with a kinematics equation to calculate a new position based on the thermal velocity. This equation is given below:

$$(x, y) = (x_0 + v_x dt, y_0 + v_y dt)$$

where x_0, y_0 are the previous positions, v_x, v_y are the component velocities, and dt is the time step. Once the new positions are calculated, the boundary conditions are enforced. The top and bottom boundaries were meant to reflect the electron, such that the angle in equals the angle out. To accomplish this, the y-component of the velocity must simply switch signs. Next, the left and right boundaries were meant to cause the electron to come out the other side. This behaviour models a long path with a small distance. To accomplish this, both the initial and final positions were changed, based on the region's length, as to ensure the new trajectory did not cause a horizontal line through the model when graphed. Then, once all boundary conditions were enforced, visualizing the behaviour is as simple as plotting the initial and final positions.

In addition to the movement, the lattice temperature was also calculated. This was accomplished with the same formula used to find the thermal velocity, but reworked for the temperature. Specifically, the equation is as follows:

$$T = \frac{m_n \overline{v_{th}^2}}{2k}$$

where T is the temperature, m_n is the effective mass, $\overline{v_{th}^2}$ is the average squared velocity, and k is Boltzmann's constant. However, since the simulation uses component velocities, the average squared velocity is calculated through:

$$\overline{v_{th}^2} = mean(v_x^2 + v_y^2)$$

The final simulation and temperature of the basic electron modelling is shown in Figure 1.

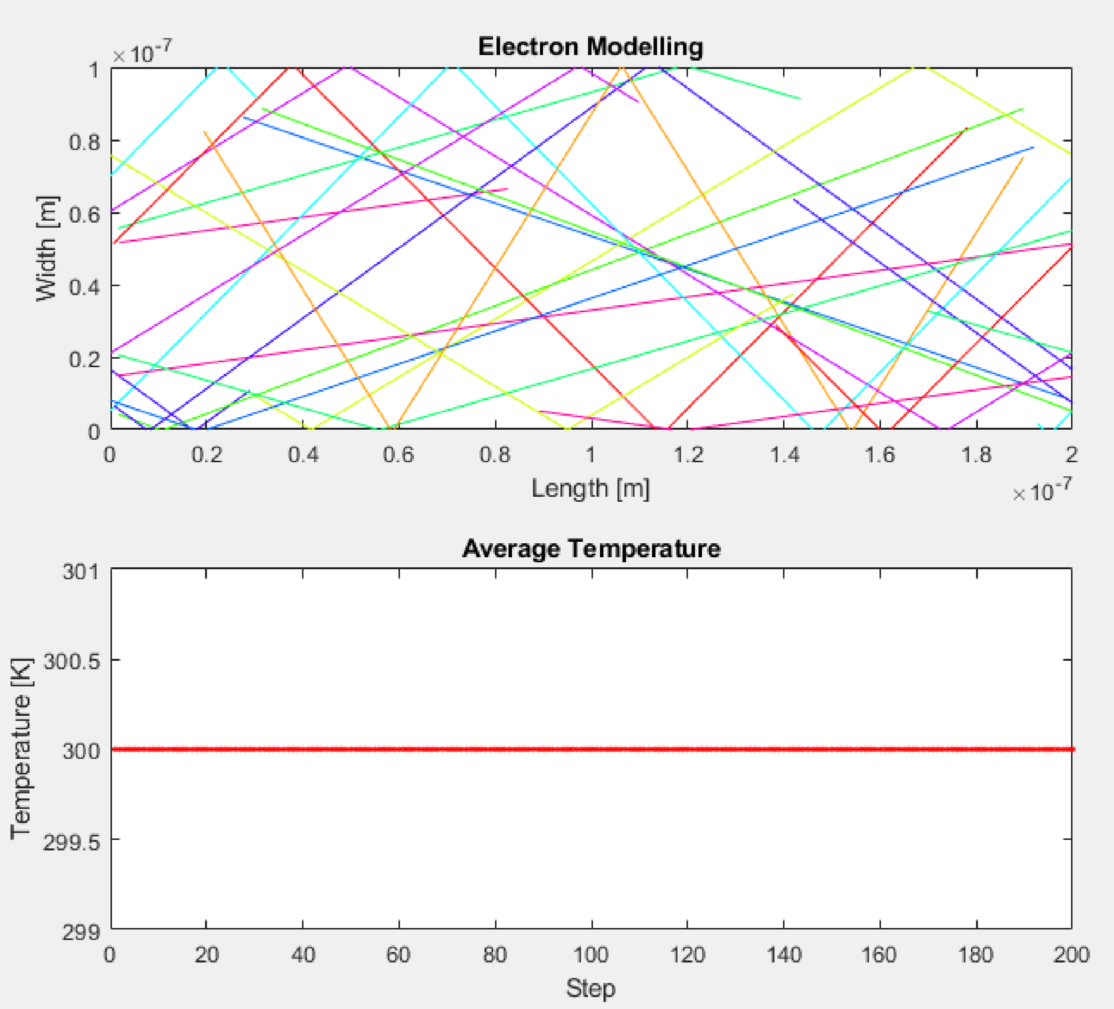

Figure 1: Electron Modelling Results

As seen in the results, the boundary conditions hold. The top and bottom reflect the electron as they are supposed to, and the left and right sides transition the electron to the other side of the region. In addition, the temperature remains at 300K. The temperature result is expected since the velocity of the electrons does not change, and therefore, the temperature does not change.

Part 2: Collisions with Mean Free Path (MFP)

The purpose of this part of the assignment was to assign the electrons a new velocity (re-thermalize) such that they scatter, and all based on a probability of scattering. The velocities were assigned using the Maxwell-Boltzmann distribution for each component. To accomplish this, the randn() function in MATLAB was used, along with an accompanying factor to scale to the thermal velocity.

$$random\ number = \frac{v_{th}}{\sqrt{2}} randn()$$

The factor of $\frac{v_{th}}{\sqrt{2}}$ is σ, proven as follows:

$$\frac{v_{th}}{\sqrt{2}} = \frac{\sqrt{\frac{2kT}{m_n}}}{\sqrt{2}} = \sqrt{\frac{kT}{m_n}} = \sigma$$

for the Maxwell Boltzmann Distribution. The distribution of velocities is shown in Figure 2.

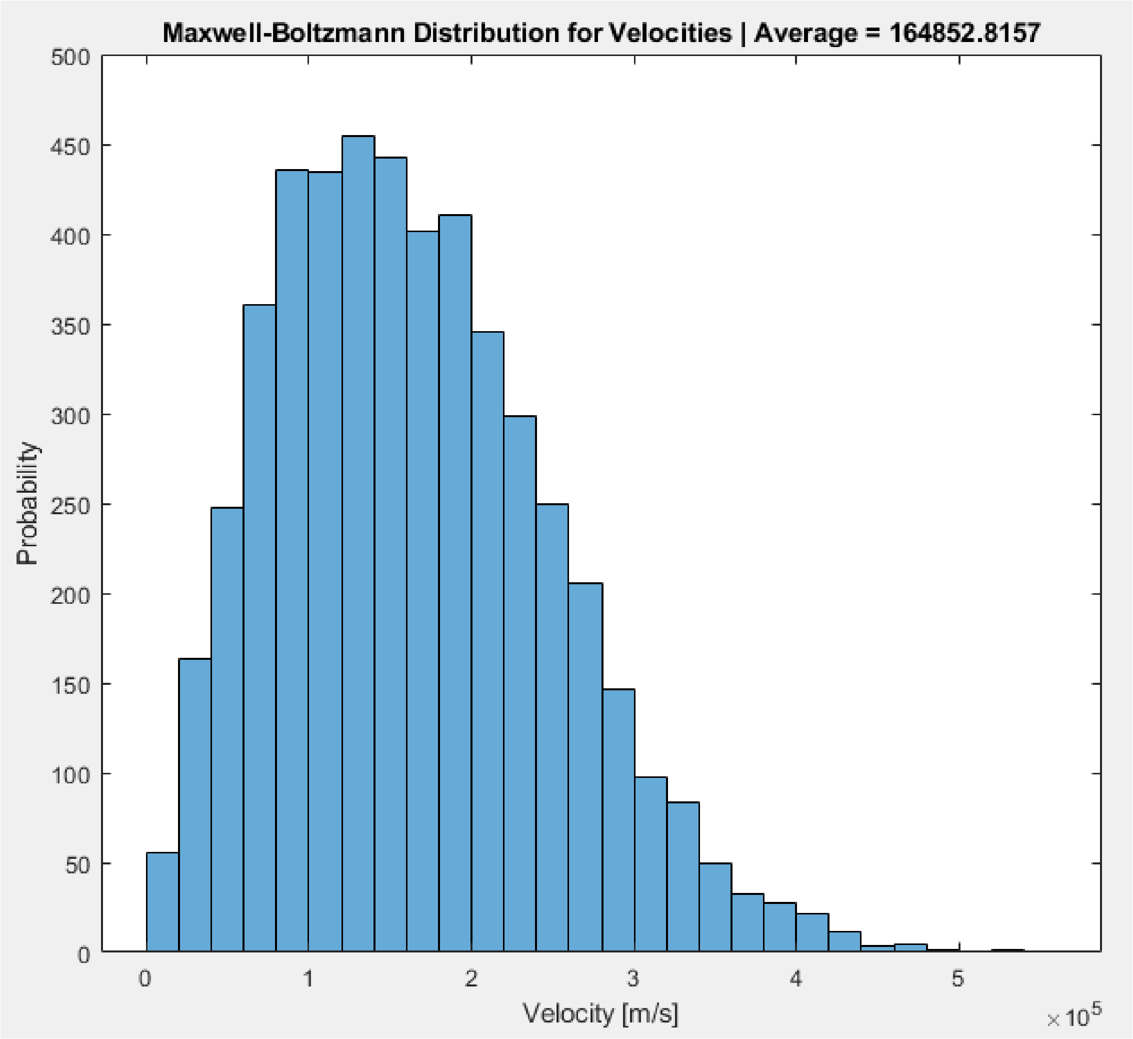

Figure 2: Maxwell Boltzmann Distribution of Velocities

Note that this distribution also has an average close to that of the original thermal velocity of $1.8702x10^5\ m/s$.

The probability of scattering was also defined as an exponential function:

$$P_{scat} = 1 - e^{-\frac{dt}{\tau_{mn}}} = 0.0521 = 5.21\%$$

for this specific time step. Recall that the time step was calculated for an electron to move 1/100 of the region’s length at a lattice temperature of 300 K.

The simulation was then ready to begin, after calculating the distribution and probability of scattering. Each step starts by generating a random number for each electron to determine whether or not the particle scatters, based on the condition that the particle scatters if the random number is less than the probability of scattering. Overall, this behaviour models and implies that each particle has a 5.21% chance of scattering each time step, but this happens randomly. Then, if the particle does scatter, a new velocity is taken from the Maxwell-Boltzmann distribution. Regardless of whether the particle scatters, the position is updated, and the boundary conditions are enforced, as in Part 1. Finally, the temperature is re-calculated as before, but due to the changing velocity from the distribution, the temperature of the lattice should change every time step. Therefore, the overall behaviour further helps to model the random movement of electrons in a medium. The final simulation results are shown in Figure 3.

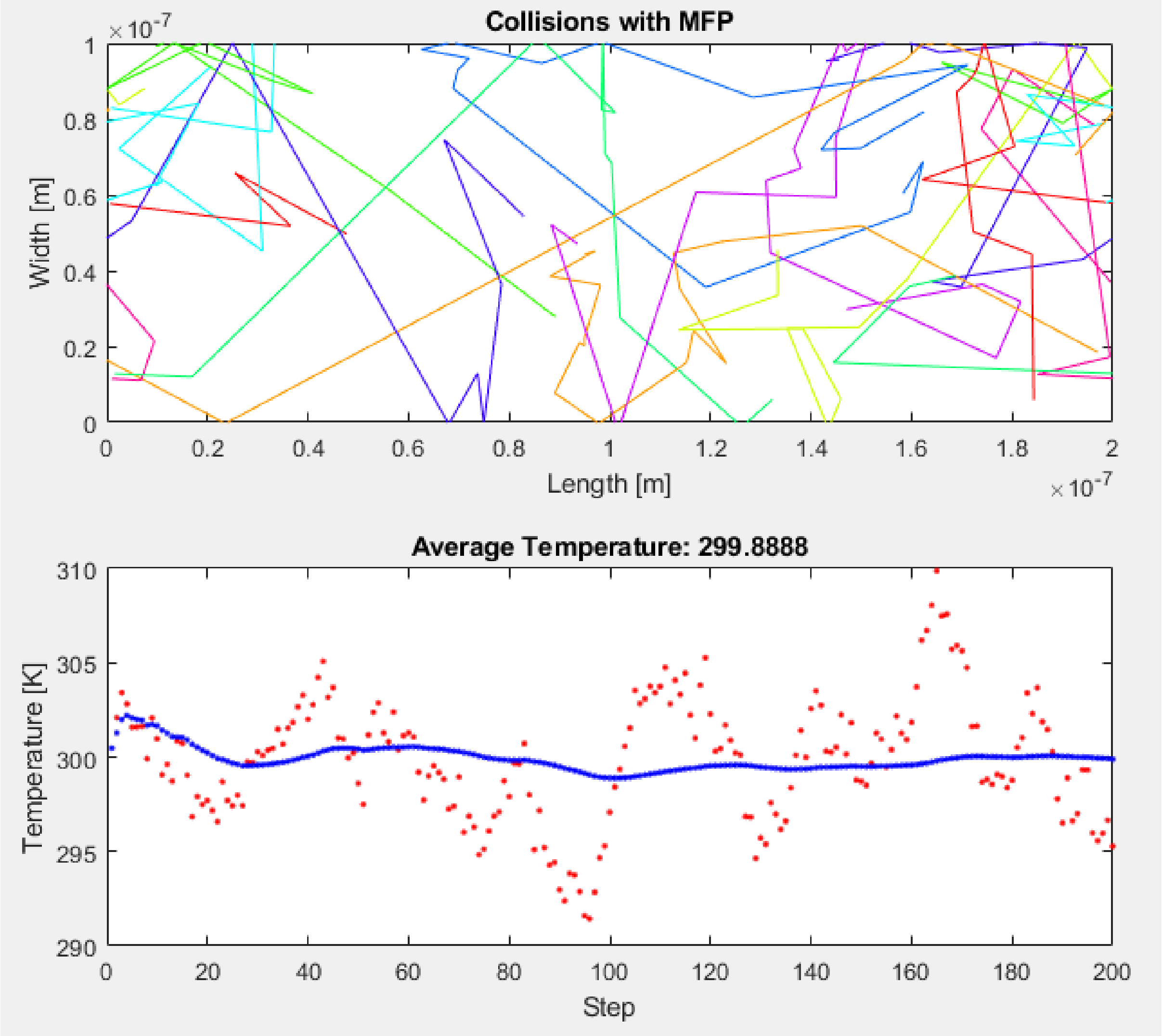

Figure 3: Collisions with MFP

As seen in the results, the overall behaviour is as expected. Electrons seem to scatter, but the boundary conditions hold. In addition, it is noted that the temperature does change every time step (shown in red), but the average (shown in blue) is 300K the more steps there are. Therefore, the average of the distribution remains to be the original thermal velocity, and because of this, the measured mean free path and time between collisions aligns with the previous results of $\lambda = 37.404\ nm$ and $\tau_{mn} = 0.2ps$.

Part 3: Enhancements

The purpose of this part was to utilize the logic in Part 2, but add in further constraints by using boxes to create a bottle neck. These boxes were placed in the middle of the top and bottom areas of the region, each 40nm tall, and thereby providing a 20nm wide space for electrons to travel in. However, these boxes also present other boundary conditions that must be met, and are dependent on whether the box itself is defined as a specular box (Part 1's BC's) or a diffusive box (re-thermalize, as in Part 2).

The boxes were created by using a structure to hold structures of each box. Each structure inside the boxes structure was its own box – this format allows additional boxes to be defined in the future. Then, each box structure consists of coordinates to be used with the rectangle() function in MATLAB, and a string variable to determine its type: specular or diffusive.

As previously mentioned, the addition of these boxes introduces a new boundary in the original problem. To accommodate this, the electrons not only have to be placed outside of the boxes, but the BC's also have to be checked every time step. The positions were calculated in a very forced way. Specifically, a random position inside of the overall region was generated, and if the position was in a box, a new position was generated until the position was outside of the box. After this condition was true, the position was added to the position array, and this process was completed for each electron. This solution may be improved, but since it works, it was used. Once all the positions were chosen, a velocity was generated based on the Maxwell-Boltzmann distribution. Finally, the simulation was ready to begin.

The simulation itself was copied from Part 2, and modified to include the BC's for the boxes. The code to accommodate the BC's was placed at the end of the logic, but before plotting, so that new positions and velocities would already be calculated and would be available to check the BC's of the boxes. Any electrons that were inside either of the boxes were found, and their indices in the position array were placed in a checking variable. Then, the y-coordinate of the old positions for each electron can be used to determine which box the electron is interacting with. This is helpful in the case that boxes may be different types. Finally, since the walls of the boxes have different implications, the initial x-coordinate was used to determine if the electron was in the left side of the box (and therefore, has positive x-velocity), the right side of the box (and therefore has negative x-velocity), or in the bottle neck (and therefore has positive or negative y-velocity, depending on which box it is interacting with). Then, the following conditions were set with the following logic, based on the box type:

- Specular type – must reflect with same velocity
    - Left/Right Wall: negate the x-velocity
    - Bottom/Top Wall: negate the y-velocity
- Diffusive type – must generate new velocity from Maxwell-Boltzmann distribution
    - Left Wall: generate new velocity, but x-component must be negative to avoid going inside the box
    - Right Wall: generate new velocity, but x-component must be positive to avoid going inside the box
    - Top Wall: generate new velocity, but y-component must be positive to avoid going inside the box

- ○ Bottom Wall: generate new velocity, but the y-component must be negative to avoid going inside the box

After the new velocities were calculated, given the conditions above, new positions were immediately calculated such that the electrons never went inside the boxes. Then, the initial and final positions were plotted. The final simulation results are shown in Figure 4.

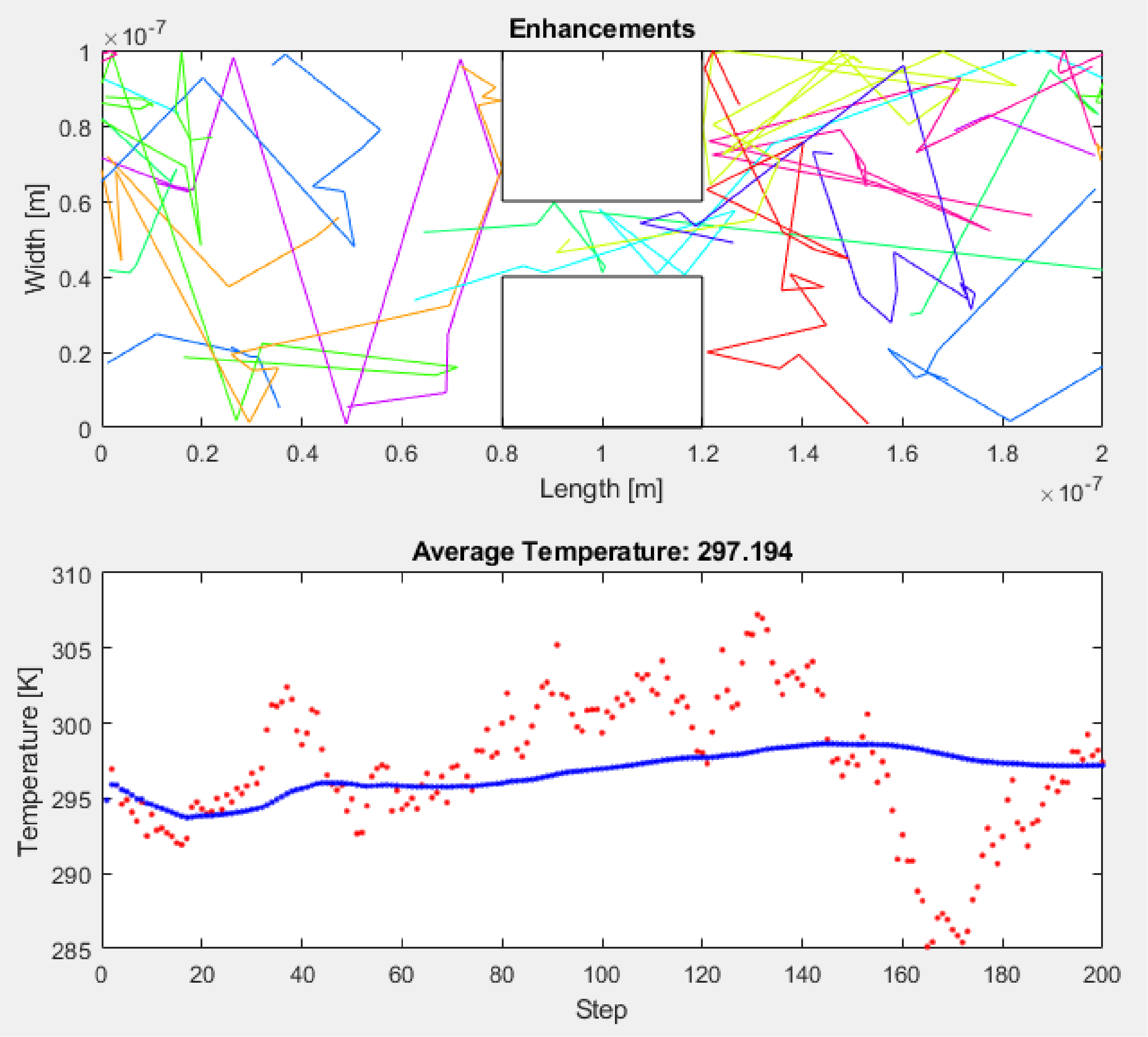

Figure 4: Enhancements Simulation Results

As seen in the results, the model accurately reflects the bottle neck behaviour, as proven by the fact that the electrons never penetrate the boxes. In addition, the top box was defined to be of specular type, whereas the bottom box was defined to be of diffusive type. The results also show how the conditions on the type are implemented correctly. Specifically, every electron that hits the top box maintains the same angle going into and out of the boundary. Every electron that hits the bottom box does have a new velocity. Both of these characteristics are clearly seen in the green/teal trace between the bottle neck. Finally, the temperature is expected to vary around 300K, similar to Part 2, and in this case, averages below 300K. However, because of the random nature, the temperature may also average higher than 300K as well.

After the simulation was completed, a density map of the electrons was created to ensure no electrons were inside the box. This is important as the simulation only displays 10/5000 electrons and a density map can easily show the locations of all electrons. The density map of the electrons is shown in Figure 5.

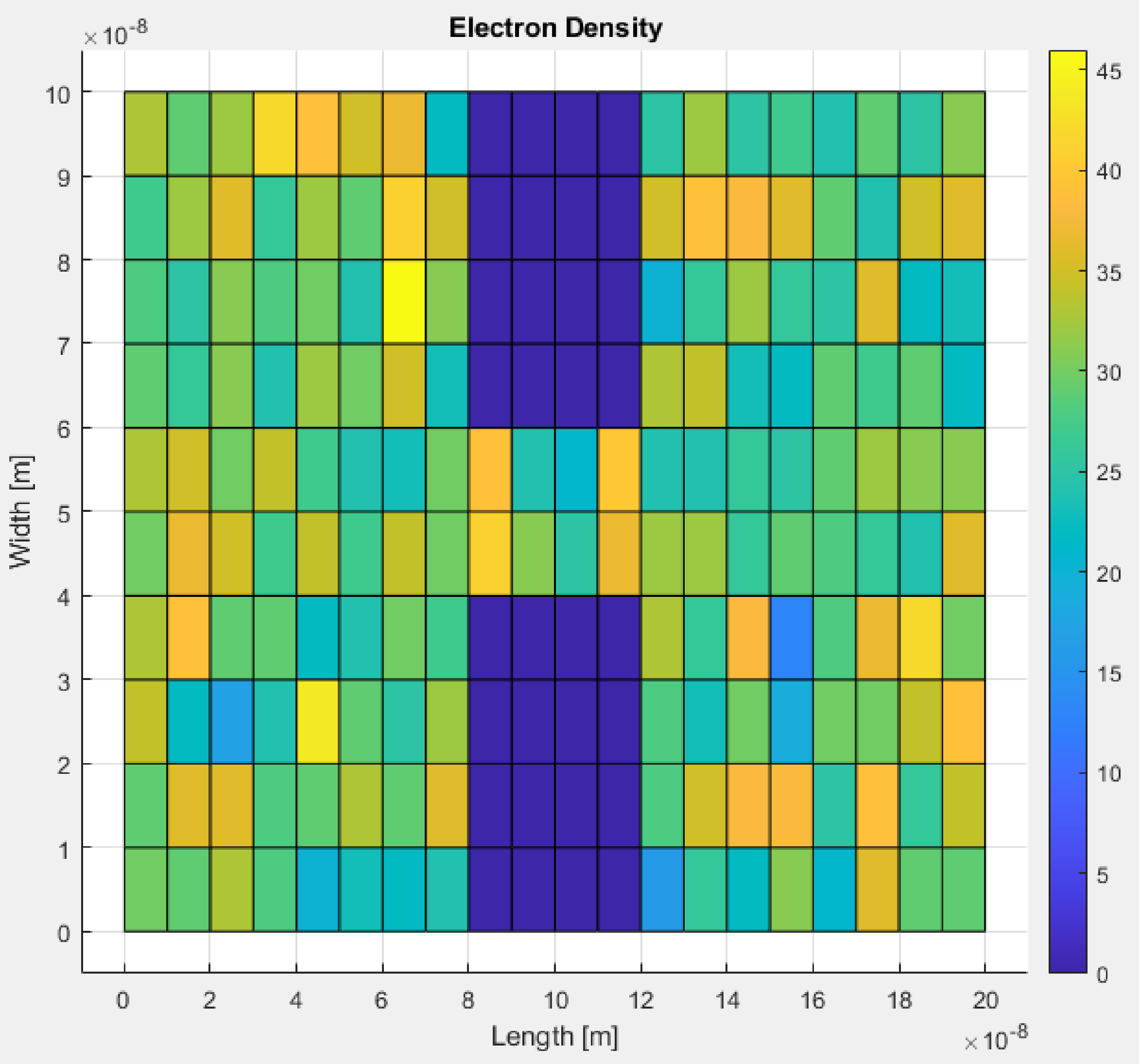

Figure 5: Electron Density Map

As seen in the density map, none of the electrons are in the boxes, thereby confirming the model and enhanced behaviour.

Finally, a temperature map was also generated as to observe the areas that have faster electrons, and therefore, more heat. The temperature map is shown in Figure 6.

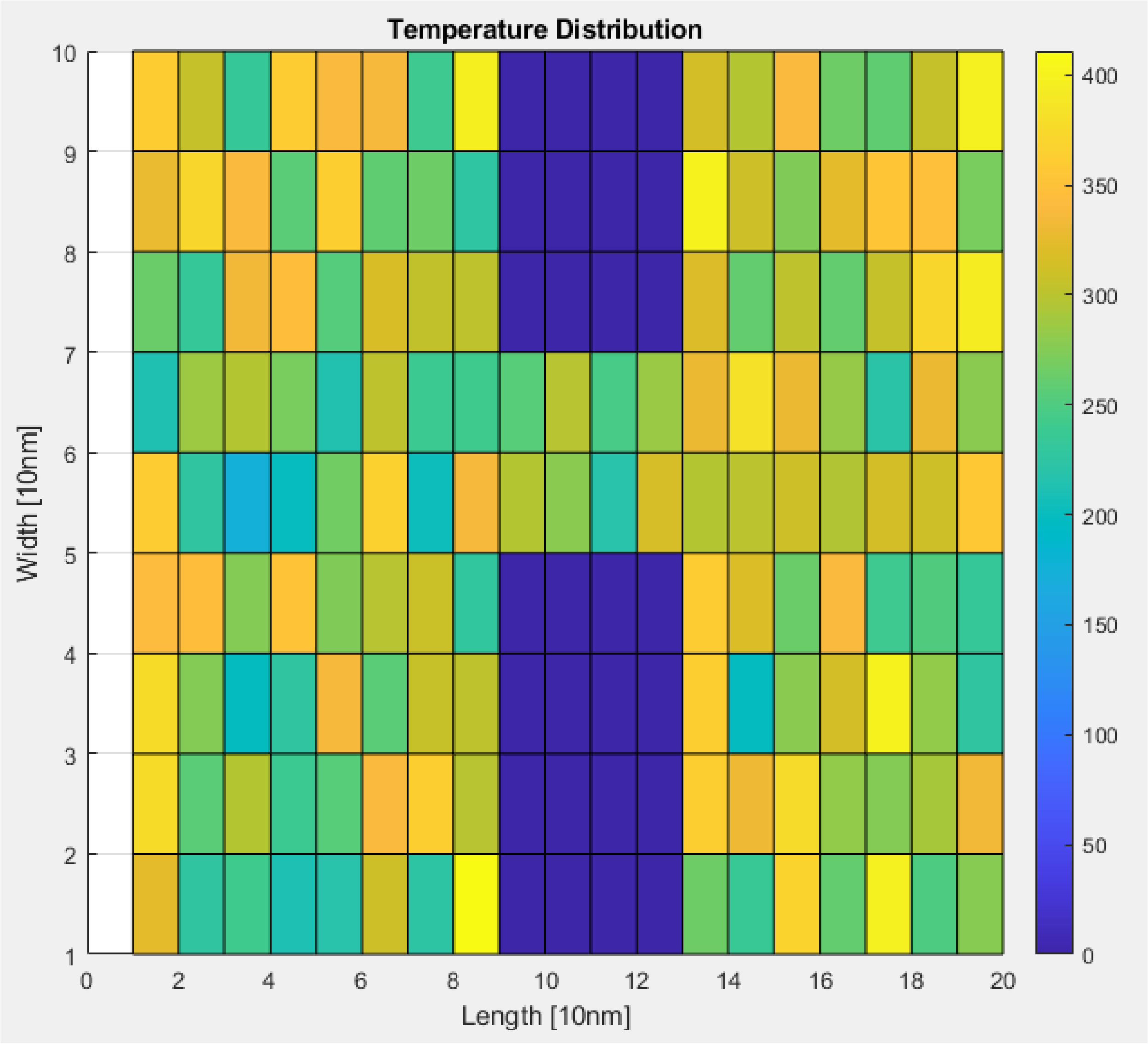

Figure 6: Temperature Distribution Map

As seen in the map, the bins are not set similar to the electron density map (due to using different methods), but regardless, the results appear to be as expected. That is, since there are no electrons inside the boxes, there is no thermal velocity, and thus, the temperature is 0 K. In addition, the temperature is significantly lower in areas that have lower electron density (such as the bottom left area), and higher in areas have a larger electron density (such as the top-right area of the left side).