ELC 4700 Assignment 1:

Monte-Carlo Modeling of Electron Transport

Wesley Stakes 100971493

Sunday February 3, 2019

Introduction

This assignment will involve using the Monte-Carlo method to model electron movement and scattering in a closed system. This assignment is broken apart into three sections, Electron Modelling, Collisions with Mean Free Path, and Enhancements. Each of these sections is written as its own Matlab script, with majority of the code being copied over to the next section.

Electron Modelling

The first part of the assignment was to create a program that would model the path of the electrons in a closed system. To do this, the electrons were assigned a random location on the x-y plane with a random angle. This random angle was used along with the calculated thermal velocity of an electron to get a random electron path. This simulation could be run for however many electrons necessary, but in this case it was run for 10000 electrons. The temperature was also plotted, and was steady at 300K. This is what is expected as we set the initial temperature of the system, and no scattering occurs. In this section the mean free path or MFP and the thermal velocity were calculated. The MFP was found to be 2.644852×10^{-8}, and the thermal velocity was found to be 1.322426×10^{5}. The final plots can be found in Figure 1 below.

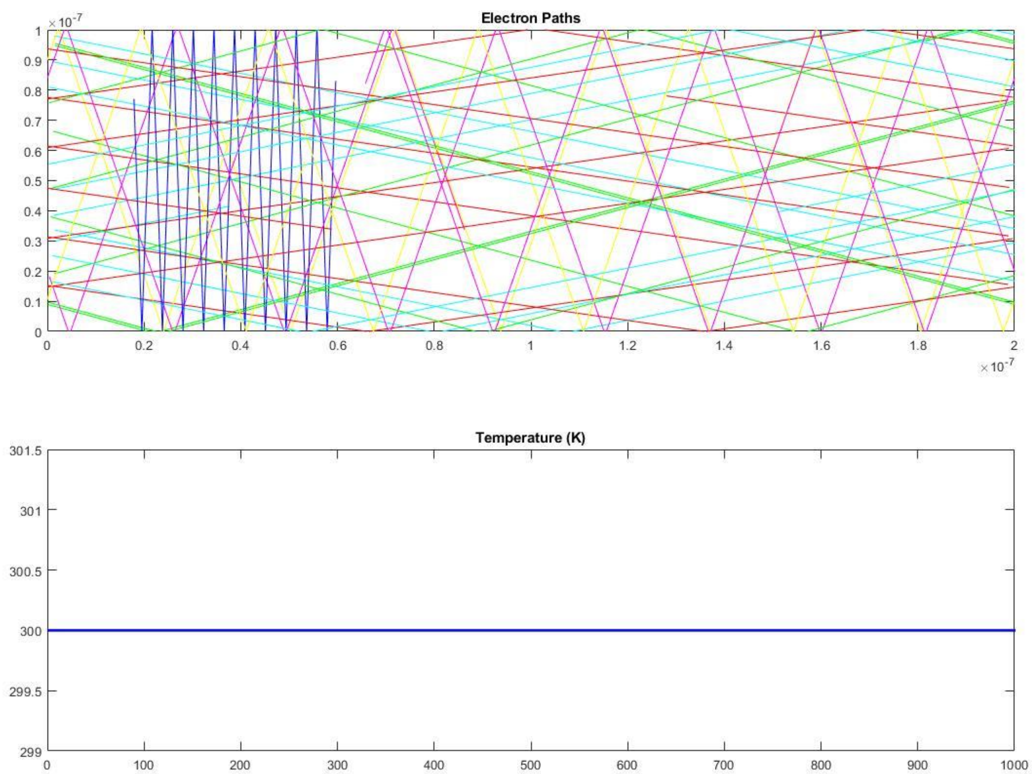


Figure 1: Simulation Results from Part 1

This concluded the coding for the first section, and the code can be found in the appendix, or in the Matlab script file labelled a4100Assignment1_part1.m. The electron reflection and movement was created using matrix indexing. By finding the specific electrons that are going cross the barriers, the velocity or position is changed to suit the specific boundary. This will become important in the enhancements section later.

Collisions with Mean Free Path

This section took what was created in the first section and added to it. The main difference from the code in the first section is that the initial velocities were now set as a random from a distribution. This distribution can be found in Figure 2 below. This section also introduced the probability of an electron scattering, or in other words the direction and velocity of the electron change to another random direction and velocity. This was done by implementing a variable containing the probability of scattering, then creating a vector the same size as the number of electrons and populating it with a random value between 0 and 1. Then each location was checked to see if it was less than the probability, and if so the velocities were regenerated randomly to create scattering. Again the mean free path and thermal velocity were calculated. The mean free path was calculated to be 3.740385^{-8} and the thermal velocity to be 1.870193^{5}. The results of the simulation can be found in Figure 2 below.

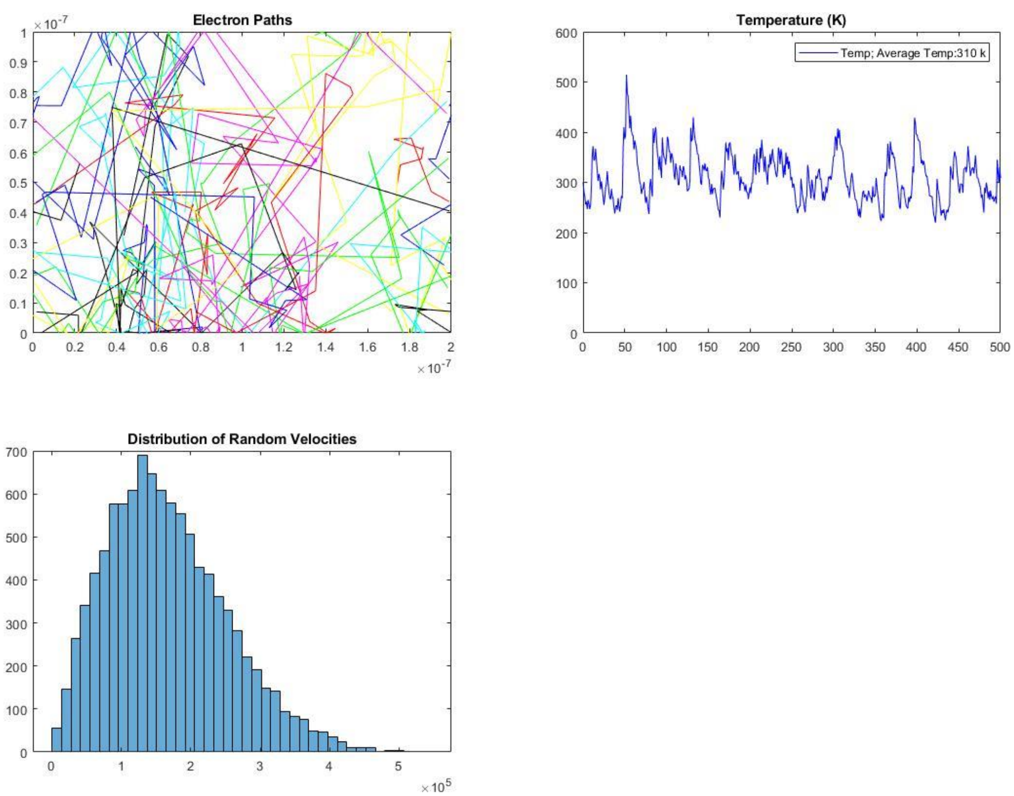
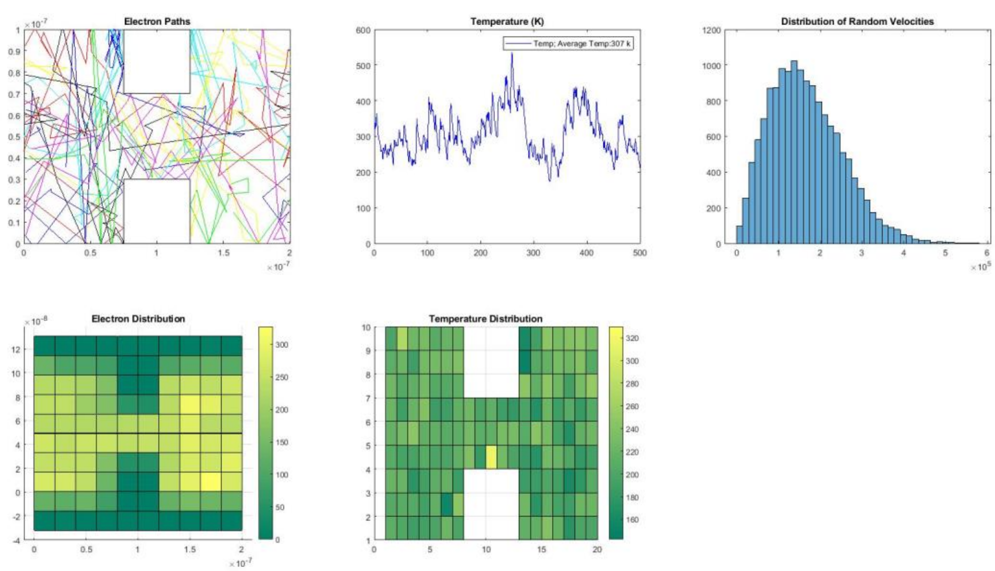


Figure 2: Simulation Results from Part 2

Here it can been seen that the effect of scattering causes a fluctuation in temperature in the system. Since temperature is a function of particle velocity, when the particle scatters to a distributed velocity, it can raise or lower the temperature depending on what velocity is chosen. A slower velocity will lower the temperature, and a higher velocity will raise the temperature. This can be seen on the temperature plot. The areas where the temperature increases is a result of scattering to higher speeds.

Enhancements

Once the script for the second part was complete, it was time to increase the number of particles, and create a new geometry for the electrons to move around. The first thing added to the script was two boxes to act as a blockade for the electrons. These were implemented using the rectangle command in matlab. The boundary conditions were set using the array indexing as before with the velocity inverting in the perpendicular axis to the face of the object. This means that if an electron hit a side of the box, the velocity in the x direction is inverted. Also added was an electron density map. This was created using the hist3 function and plotted the number of electrons in each “box” of the geometry. The hist3 plot was then changed to be a 2D plot with color indicating the number of electrons. Finally, a temperature density map was generated. This was done be indexing the specific electrons in a given box and then calculating the average velocity. Once this was done, the temperature was calculated and plotted using surf. The resulting plots can be found in Figure 3.



The code for this section can be found in a4700Assignment1_part3.