

ELEC 4700 Assignment 1: Monte-Carlo Modeling of Electron Transport

Alison Fong

101048052

Introduction

This assignment was done to learn about the Monte-Carlo method of modelling electron trajectories through semi-conductors. The goal of the first part was to make a simplistic Monte-Carlo model of electrons in silicon. The second part adds the probability of each electron scattering within the silicon to the model using a Maxwell-Boltzmann distribution. The third part adds two boxes to the model to mimic a bottleneck effect. All parts were modeled using MATLAB and given value for effective mass ($m_n = 0.26m_0$) and region size (200nm x 100nm).

Part 1: Electron Modeling

The goal of this part was to create a simple Monte-Carlo model. The thermal velocity was calculated using the effective mass, the assumed $T = 300K$, and the constant k_b in the following equation:

$$V_{th} = \sqrt{\frac{k_b T}{m_n}} = 1.3224 \times 10^5$$

With this value and the assumed value $\tau_{mn} = 0.2ps$, the mean free path can be calculated as:

$$MFP = V_{th} \tau_{mn} = 2.6448 \times 10^{-8}$$

The code was made up of four position vectors (x & y, old & new) and two velocity vectors. Using a for loop to repeat multiple steps, the code was written to plot two points on the graph at a time per particle. The for loop also contains the boundary conditions and what type they were, either wrap-around or reflective. This was done using logical indexing. The attached code Part1_try7.m also contains an attempt to plot the temperature plot with no success before the due date.

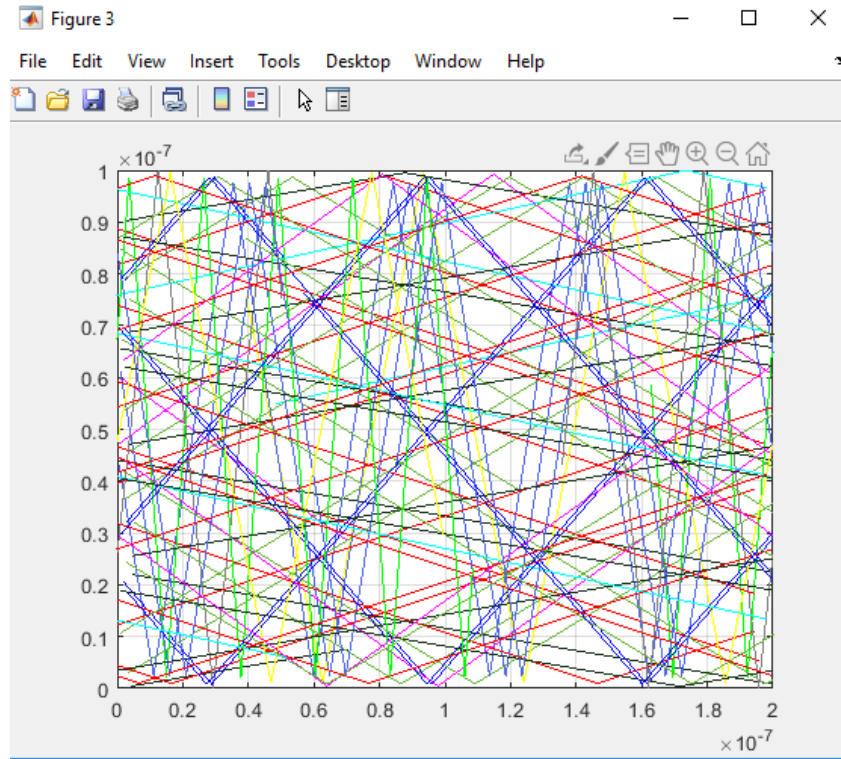


Figure 1: 2-D plot of particle trajectories

Part 2: Collisions with Mean Free Path

This part of the assignment was not finished due to lack of time management. Beyond this point is an assumption of what should have been done to create the wanted model.

Based on the code for Part 1, a probability factor for the particles would need to be added. Using the Maxwell-Boltzmann distribution, the velocity of each particle would be dictated by a probability parameter. For example, if the probability of scatter was more than 0.95 (arbitrary number), then the particle will scatter and assign it a new velocity and/or direction. Adding this parameter into the for loop will make the particles in the model look as though they've hit something and ricochet away.

The temperature plot would, ideally, have been plotted the same way as Part 1; taking the mean of the velocities at each timestep and plot it.

Part 3: Enhancements

This part of the assignment was also not finished due to lack of time management. Again, the following will be a summary of what was assumed to have been needed to be done.

This part adds two boxes parallel to each other in the middle of the model. To plot these boxes, the function `plot()` would be used multiple times to draw a line between two points, four times per box. Each box would correspond to three boundary conditions visible to the particles travelling in the silicon (assuming the boundary conditions for the sides the boxes touch the graph boundaries are redundant) with parameters to either be specular or diffusive. Also, conditions would have to be set for the starting position of each particle to prevent them from spawning inside the boundary boxes.

The temperature map is assumed to be the same plot as in Part 1 and 2.

Conclusion

The goal of the assignment was not met due to lack of time management. Summaries of assumptions were written in place of the actually created models. Proper time management will be used in future to ensure this doesn't happen again.