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

Due: Sunday, Feb. 6, 2022 23:59

Submission

The assignment submission should be two files, a .zip of the repo and code, and a .pdf of the report. BrightSpace will accept only two files on your submission so make sure they are these.

The code should be organized such that the master branch of the repo contains the submission version and the entry point script (the one that creates the report or runs all the other code) is titled as the repo. Any additional code should be stored in a folder titled code.

In order to create a repo for your assignment, you can follow these instructions:

1. On your local machine, make sure you do git init in your assignment folder.
2. In the same directory as your assignment code and PDF commit all changes you've made to your project with first git add −A and then git commit −m "MyMessage"

As mentioned before, the submission should have the format of:

1. The report
2. A zip folder name "code" containing the code used to create the report

Your code will be checked using the version of MATLAB as found on DOE computers. Please ensure that your code clearly indicates which question and which part it is answering.

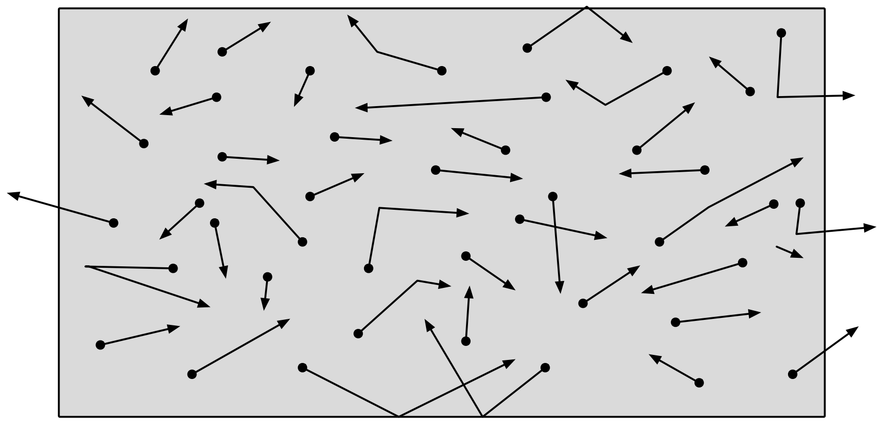


Figure 1: Semiconductor with a population of electrons moving with thermal velocities and scattering.

Modeling Drift Current In a Semiconductor

Consider the semiconductor region in Figure 1. You will be modeling the carriers as a population of electrons in an N-type Si semiconductor crystal.

Use the following values:

- Effective mass of electrons $m_n = 0.26m_0$ where m_0 is the rest mass.
- The nominal size of the region is $200\,\text{nm} \times 100\,\text{nm}$.

1 Electron Modeling (40)

Model the electrons in the silicon as particles with the effective mass above using a simplistic Monte-Carlo model.

1. What is the thermal velocity v_{th}? Assume $T = 300\,\text{K}$.
2. If the mean time between collisions is $\tau_{mn} = 0.2\,\text{ps}$ what is the mean free path?
3. Write a program that will model the random motion of the electrons. The program should do the following:
   - Assign each particle a random location in the $x-y$ plane within the region defined by the extent of the Silicon. For simplicity you may use a small number of particles (1000-10000 works well) but you can start much smaller initially if you like.
   - Assign each particle with the fixed velocity given by v_{th} but give each one a random direction. To do this pick a random direction ϕ and generate v_x and v_y. Later you will generate these from the Maxwell-Boltzmann distributions.
   - At a fixed time interval of $\triangle t$, update the particle location using Newton's laws of motion. You will need to pick a time step size that takes into account the velocity of your particles and the size of the region. Typically the spacial step should be smaller than 1/100 of the region size. Simulate for nominally 1000 timesteps. This should allow each particle to bounce around quite a bit inside the region.

- For a few of the particles trace out their trajectories using the *'plot'* command in Matlab. To plot the trajectories you should keep the previous x and y positions. *Hint: use breakpoints to step through the trajectory and see what is "going on"; use the 'pause' command in Matlab to have the plot update in a loop.*
- Show a 2-D plot of all (or a subset) of the particles that updates with each time step. *Hint: use the 'pause' command in Matlab to have the plot update in a loop.*
- For the y direction use a boundary condition where the particle reflects at the same angle (specular) and retains its velocity.
- For the x direction use a periodic boundary condition where the particle jumps to the opposite edge. i.e. if it reaches the right side it appears at the left with the same velocity.
- Calculate and display the semiconductor temperature on the plot at a fixed time interval and verify that it stays constant.
- Your program should plot trace trajectories producing something like Figure 2.
- Hint: if you use arrays of positions and velocities (P_x, P_y, V_x, V_y) in Matlab many of your calculations can be done using array math which will be much quicker!

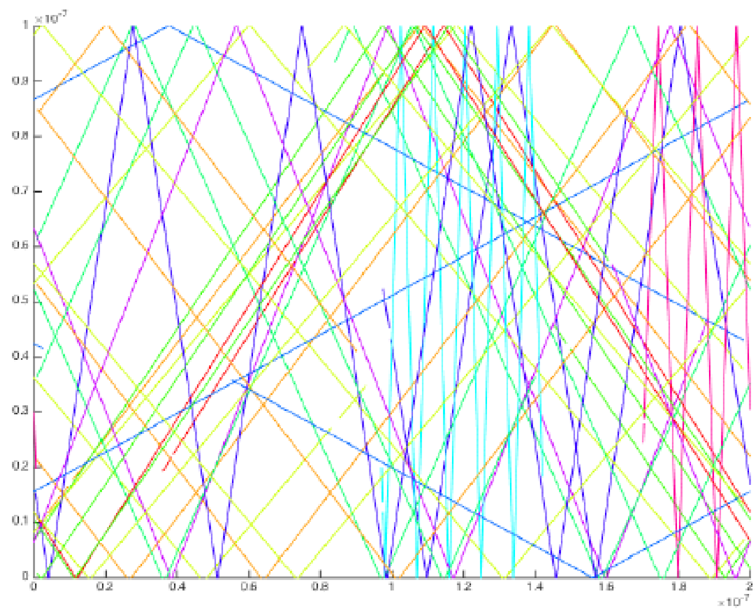


Figure 2: Sample particle trajectories.

Include in Report a) v_{th} b) MFP c) i) 2-D plot of particle trajectories ii) Temperature plot

2 Collisions with Mean Free Path (MFP) (25)

1. Assign a random velocity to each of the particles at the start. To do this you can use a Maxwell-Boltzmann distribution for each velocity component. Ensure that the

average of all the speeds will be v_{th}. Plot the distribution in a histogram (Matlab *'hist'* function).

2. Model the scattering of the electrons using an exponential scattering probability: $P_{scat} = 1 - e^{-\frac{dt}{\tau_{mn}}}$, where dt is the time since the last timestep (and P_{scat} calculation), and τ_{mn} is the mean time between collisions. Use the τ_{mn} given above. At every time step (for every electron) use something like this: if $P_{scat} > rand()$ then the particle scatters. When the electron scatters re-thermalize its velocities and assign new velocities V_x, V_y from Maxwell-Boltzmann distributions. Refer to Figure 3 for a sample plot.

3. What happens to the average temperature over time?

4. Measure the actual Mean Free Path and mean time between collisions to verify your model.

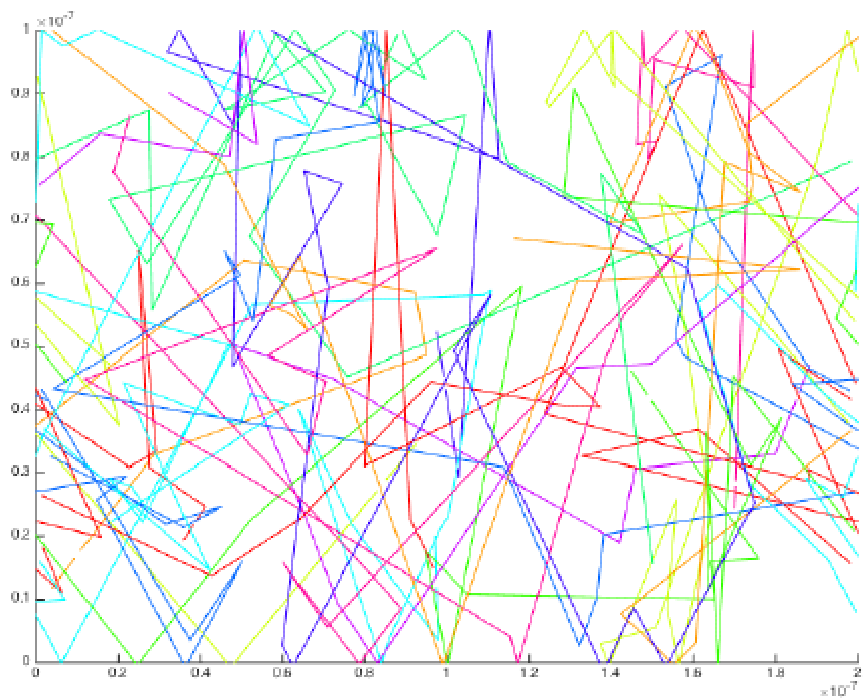


Figure 3: Sample trajectories with scattering.

Include in Report a) Histogram b) 2-D plot of particle trajectories c) Temperature plot d) MFP and τ_{mn}

3 Enhancements (35)

1. Basic Features and Analysis:

(a) Add in the inner rectangle "bottle neck" boundaries as in Figure 4. There are a number of ways to do this. I did it by defining "boxes" that reflect particles and then adding a number of "boxes" to the region. You will need to be careful not to put electrons in the "boxes" at the start. You should also check that no electrons are leaking through the "boxes".

(b) Make all boundaries capable to be either specular or diffusive (ie re-thermalized).

(c) Calculate an electron density map from the final electron positions.

(d) Calculate a temperature map and display with colors.

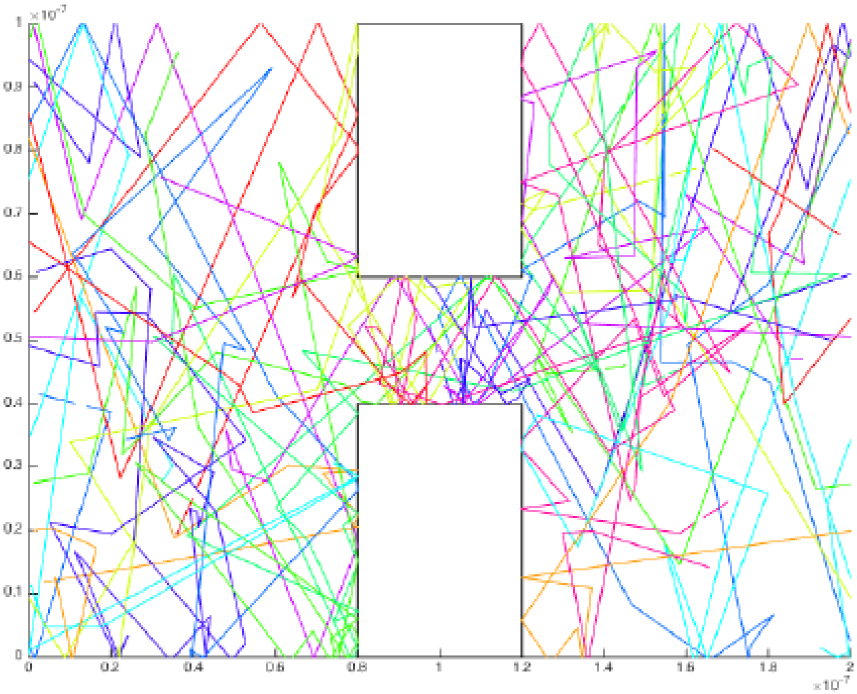


Figure 4: Sample trajectories with "bottle-neck".

Include in Report a) 2-D plot of particle trajectories c) Electron density map d) Temperature map

2. Complete one of the following two tasks. Bonus marks for doing both.

   (a) **Curved Surfaces:** Add curvilinear geometry into the simulation. An example would be a circle placed inside the region that reflects electrons. To do this you will need to (for each electron trajectory):

   - Determine if the electron has moved inside the circular inclusion
   - If it has calculate the intersection of the trajectory with the circle.
   - Then calculate the angle of incidence ϕ with respect to the normal of the surface.

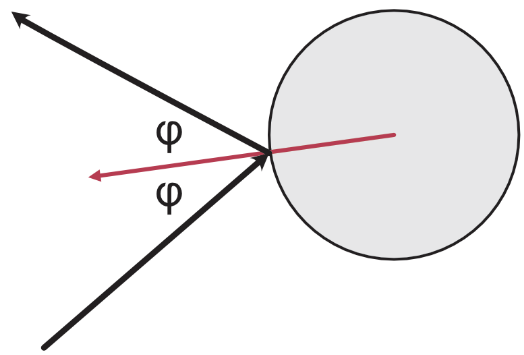


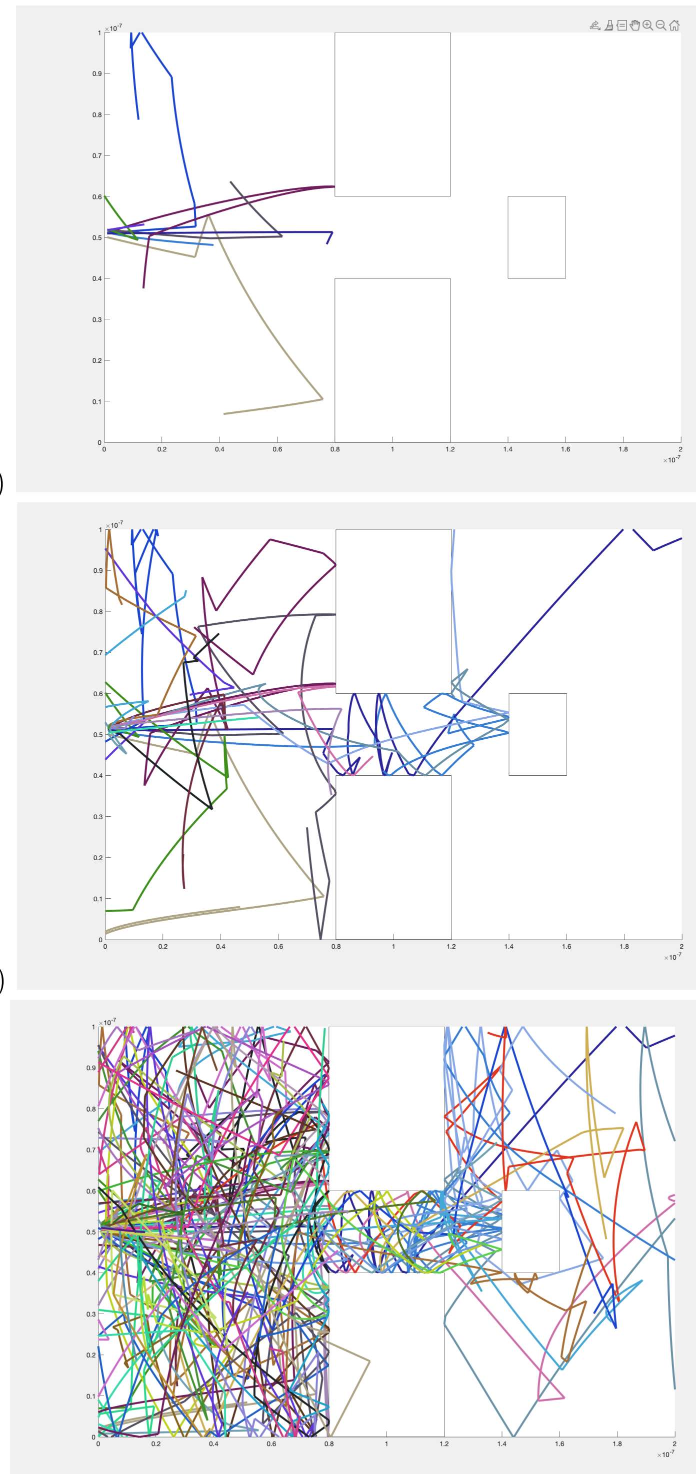
Figure 5: Electron reflection off a circle

- Finally, calculate the reflection direction using ϕ, assuming specular reflection (angle in equals angle out) and use this to determine a new position and velocity.

If you are ambitious you could try other curvilinear geometries such as ellipses and parametric curves.

(b) **Injection:** Model electron injection into a region:

i. Do not have any electrons present at the beginning.
ii. Introduce them during the simulation from the left side with a positive v_x – derived from a thermalized velocity within a small central region.
iii. Turn off the periodic BC conditions in x.
iv. Investigate turning on and off scattering and adding boxes.
v. Play with the initial value of v_x and v_y.



a)

b)

c)

Figure 6: Electron injection from the right side at three different times.