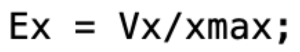
ELEC 4700 Assignment - 3
Monte-Carlo/Finite Difference Method

Submitted by: Chloe Ranahan (101120978)

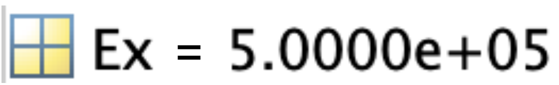
1. Monte-Carlo Simulator (A1) without the Bottle-Neck

a) Assuming an applied voltage of 0.1 V created a constant electric field across the x-direction of the device, the following equation could be used to determine the electric field:



```
Ex = Vx/xmax;
```

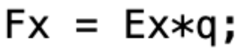
The electric field was calculated to be:



```
Ex = 5.0000e+05
```

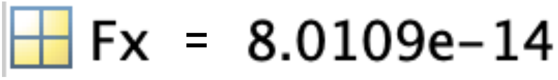
V/m

b) The force on each electron by the electric field was calculated by the following equation:



```
Fx = Ex*q;
```

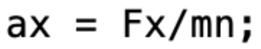
The force was calculated to be:



```
Fx = 8.0109e-14
```

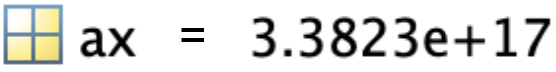
N

c) The acceleration of the electrons was calculated by the following equation:



```
ax = Fx/mn;
```

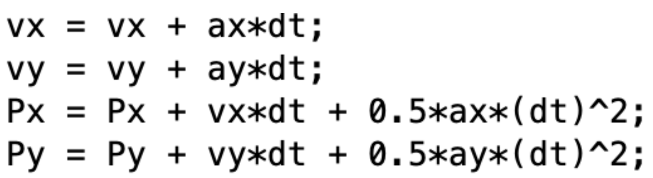
The acceleration was calculated to be:



```
ax = 3.3823e+17
```

m/s^2

The following equations were in the main for loop of the code and were used to update the electrons positions & velocities based on their current positions, velocities and acceleration.



```
vx = vx + ax*dt;
vy = vy + ay*dt;
Px = Px + vx*dt + 0.5*ax*(dt)^2;
Py = Py + vy*dt + 0.5*ay*(dt)^2;
```

In assignment 1, ax (acceleration in the x direction) and ay (acceleration in the y direction) were set to 0 as there were no external forces acting on the particles. For this assignment, ay was left as 0; however, ax was calculated using the applied electric force (Fx) and the electrons rest mass (mn).

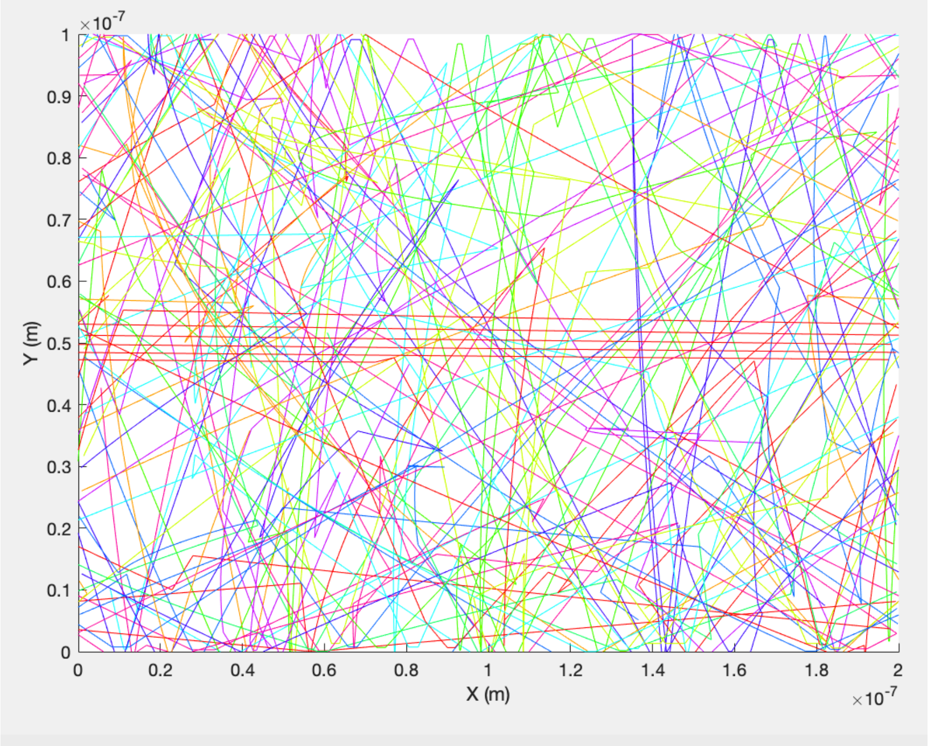
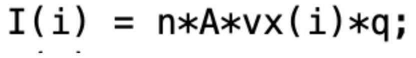


Figure 1: A sample 2-D plot of the particle trajectories with scattering and without the bottleneck. A voltage of 0.1 V was applied across the x-direction of the semiconductor.

The figure above shows a sample trajectory of the electrons over 1000 time steps. Even with just the small applied voltage of 0.1 V, a slight curving of the electrons in the positive x-direction (right) is observable. For parts 2 and 3, the curving is more noticeable due to a higher applied voltage across the device.

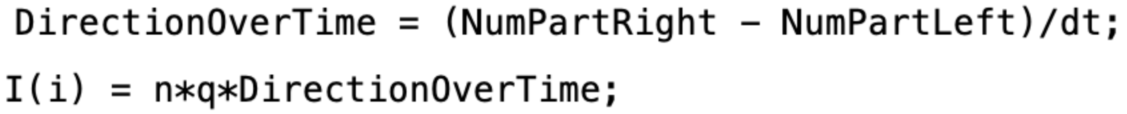
d) Originally, the drift current was calculated using the following formula:



```
I(i) = n*A*vx(i)*q;
```

Where n is the electron concentration, A is the area, vx is the velocity in the x direction, and q is the electron charge.

However, this gave a very noisy drift current over time, making the overall trend difficult to discern. Therefore, the equations below was used to find the overall average current by taking the difference of the electrons moving to the right of the area and the electrons moving to the left of the area, over time:



```
DirectionOverTime = (NumPartRight - NumPartLeft)/dt;
I(i) = n*q*DirectionOverTime;
```

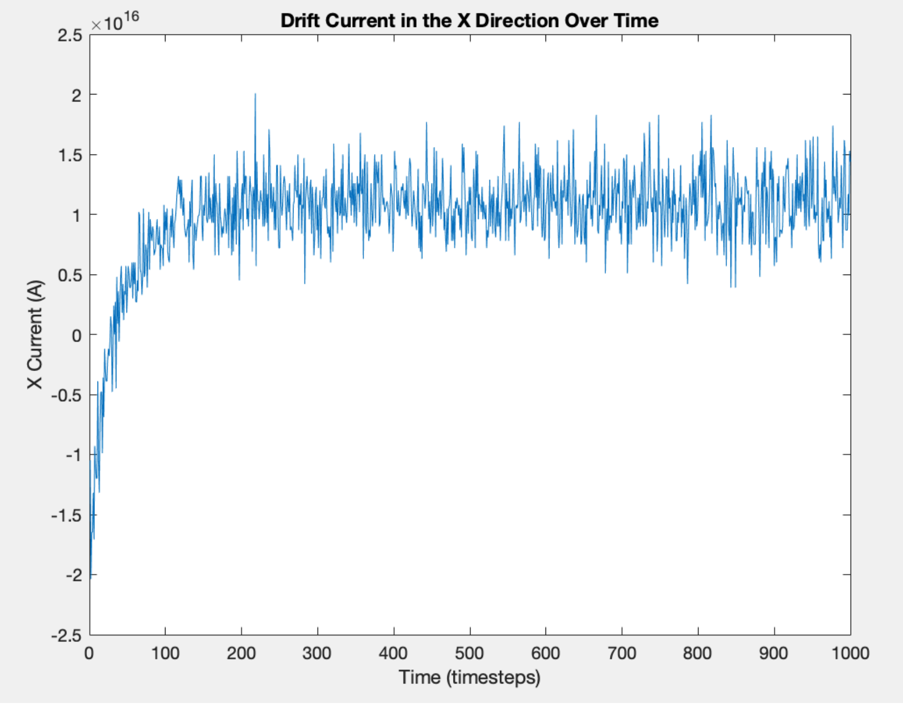


Figure 2: The drift current of the electrons in the x-direction over 1000 time steps.

The trend in the drift current started out as inconsistent and near-zero, likely due to the random initial positions and velocities assigned. Over time, the drift current in the x-direction increased as the electrons were accelerated in the x-direction by the electric field. After roughly 200 time steps, the current levelled out and reached a steady state drift current (~1.2 x 10^{16} A) due to the random scattering preventing the electrons from continuously accelerating.

e)

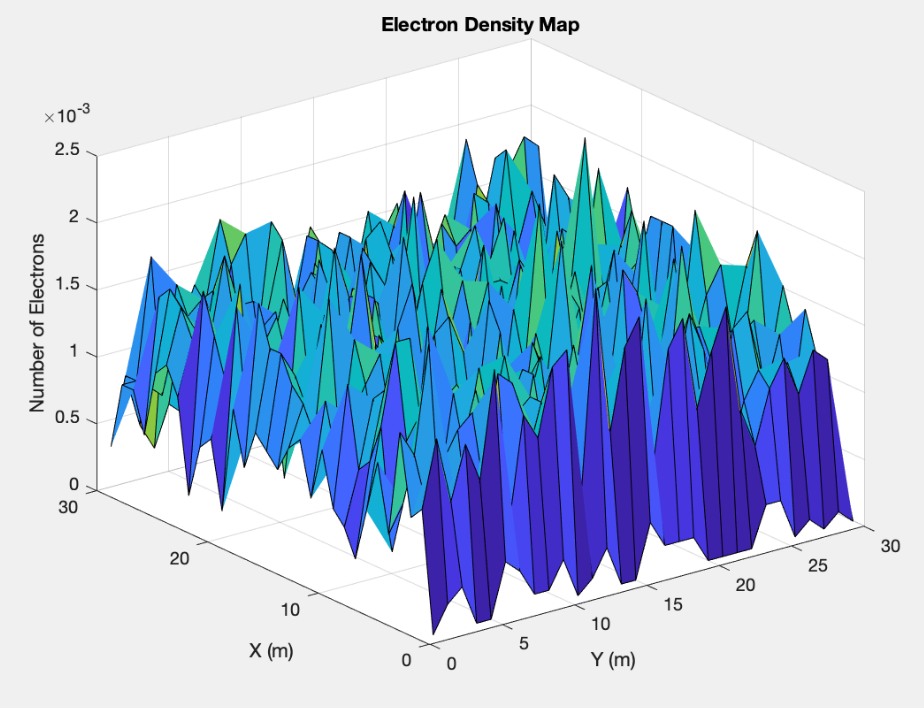


Figure 3: The electron density map after 1000 time steps, with no bottle-neck.

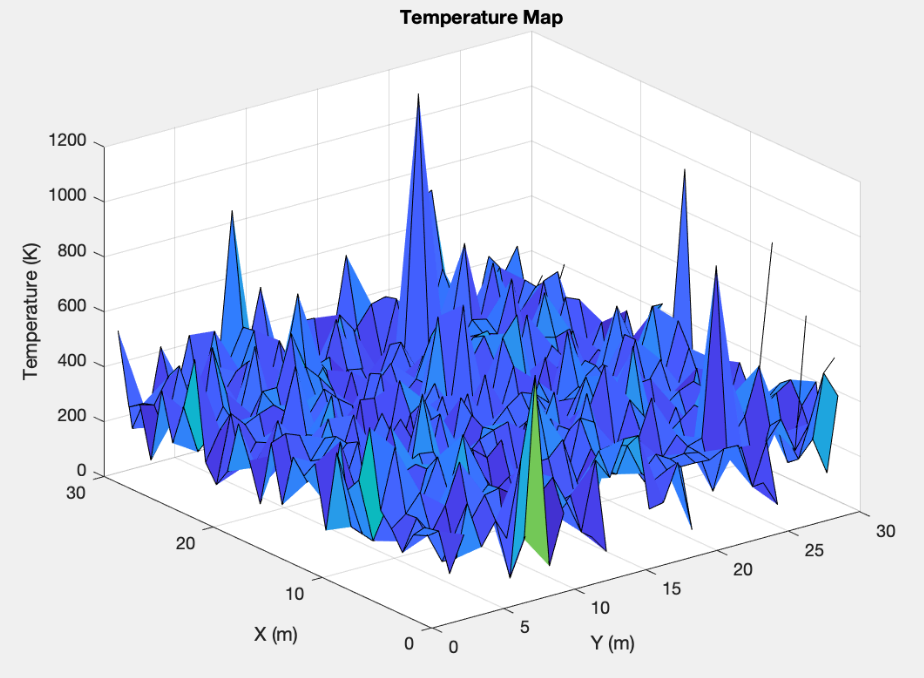


Figure 4: The temperature map after 1000 time steps, with no bottle-neck.

2. Finite Difference Method (A2) used to Calculate the Electric Field for the Monte-Carlo Bottle-Neck Simulation

a) The code from the Finite Difference Method assignment was taken and used to generate the surface potential across the device when a bottle-neck was present (see Figure 5).

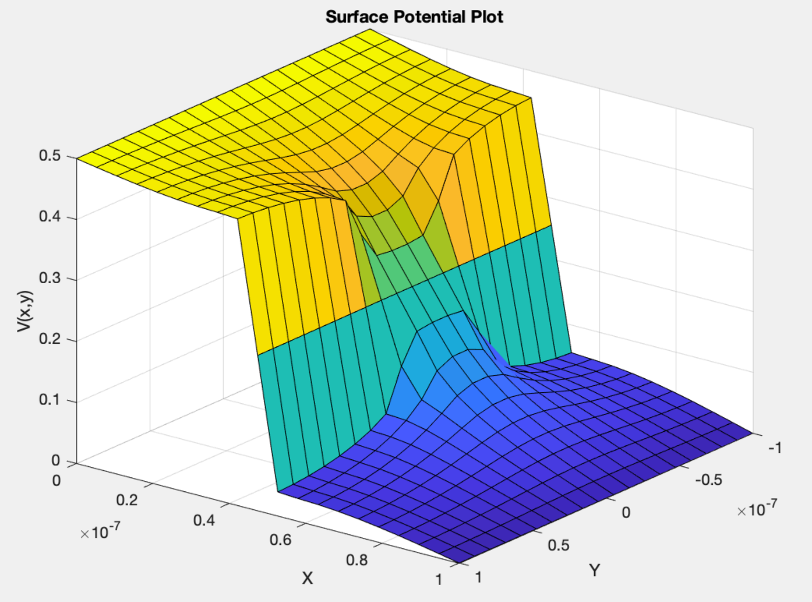


Figure 5: The surface potential plot of the bottle-neck device.

b) The gradient of this potential plot was used to generate the electric field in the x and y directions at each of the particles positions over time. This was done by first creating a mesh grid over x and y, then using the function 'interp2' in the for loop in order to interpolate the x and y direction electric fields at each of the particles' positions based on the data from the gradient of the surface plot (see Equations below).

The 'interp2' function was very useful for this; however, it had the issue of generating NaN values for the E field when the electrons hit the perimeter. This small error cascaded into generating NaN values for all the other parameters that were calculated from the E field (force -> acceleration -> velocity -> position). In order to prevent this error from stopping the simulation of all particles after hitting the perimeter, the 'isnan' function was used which would essentially reset the E field values after a NaN was produced, preventing the cascade of NaNs (see Equations below).

```
Ex_p= interp2(LX,LY,Ex.',Ppx, Ppy);
Ey_p= interp2(LX,LY,Ey.',Ppx, Ppy);
Ex_p(isnan(Ex_p)) = 0;
Ey_p(isnan(Ey_p)) = 0;

Fx = Ex_p*q;
Fy = Ey_p*q;
ax = Fx/mn;
ay = Fy/mn;
```

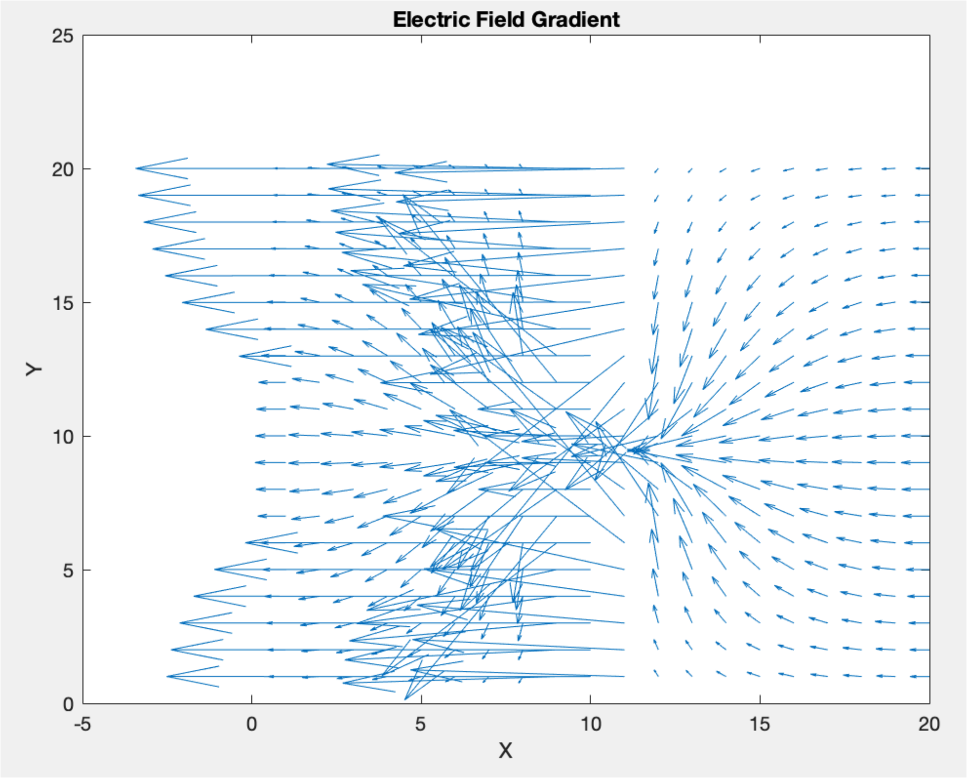


Figure 6: The electric field gradient across the bottle-neck device.

c)

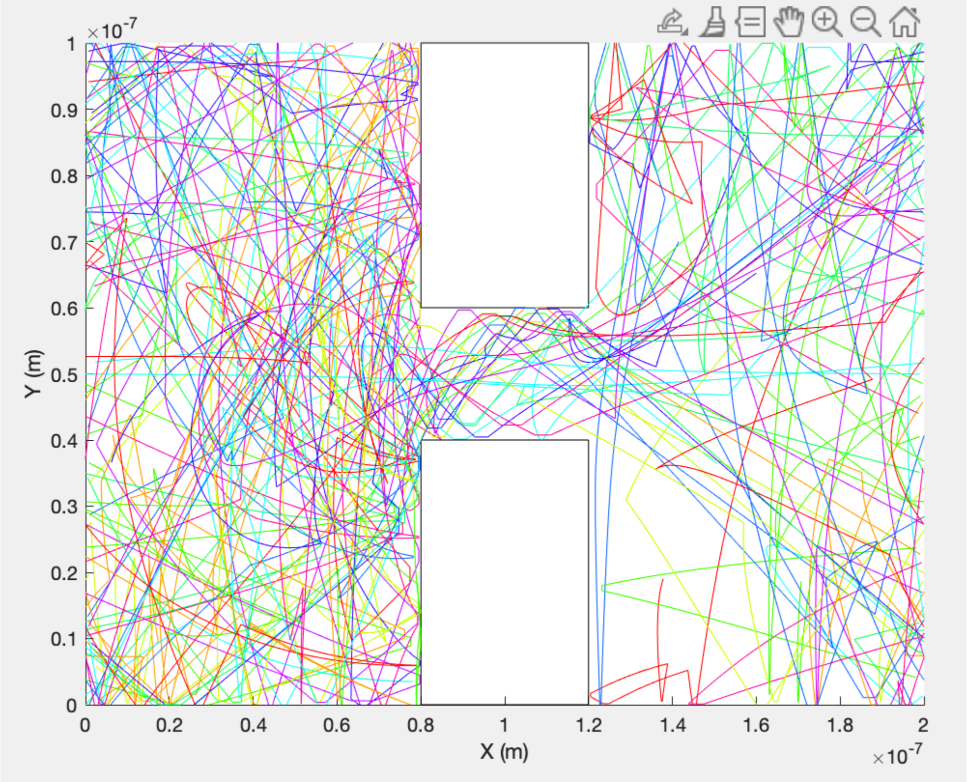


Figure 7: A sample 2-D plot of the particle trajectories with scattering and with the bottleneck. A voltage of 0.5 V was applied across the x-direction of the semiconductor (to enhance curving).

3. Device Investigation & Parameter Extraction

a)

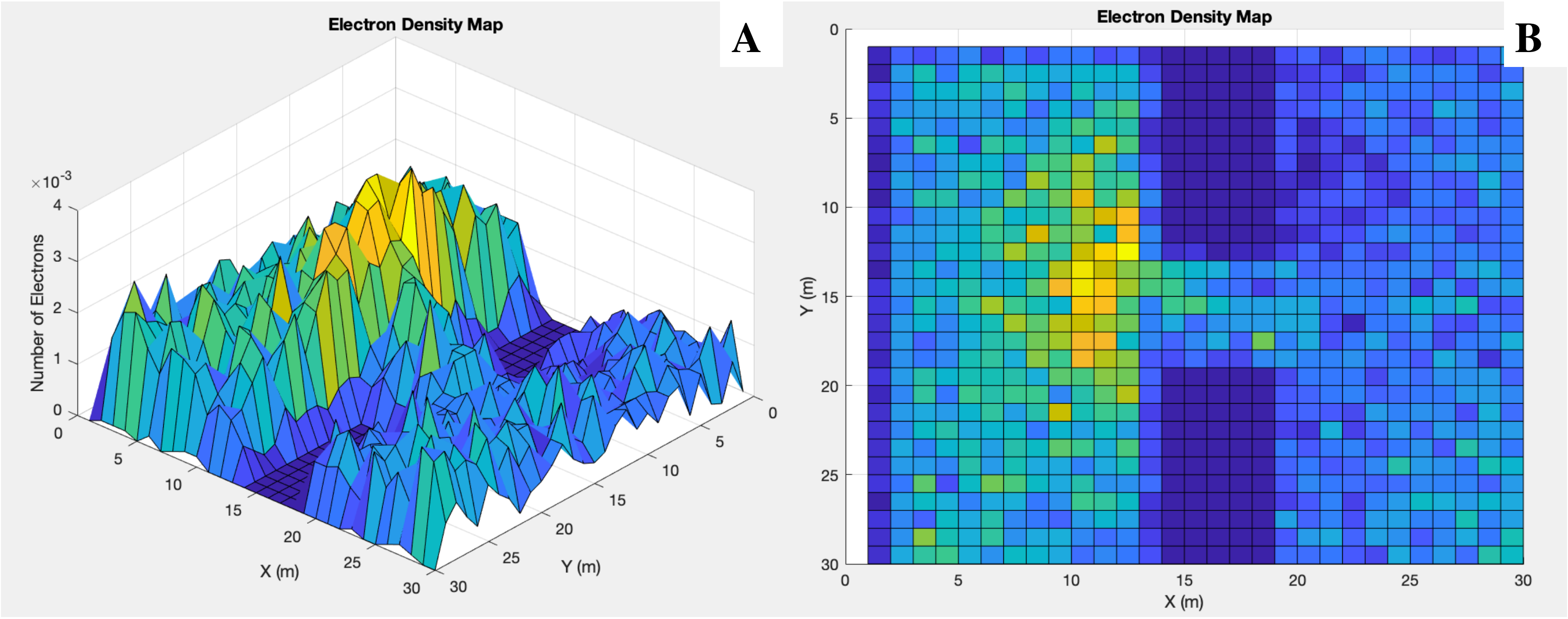


Figure 8: The electron density map after 1000 time steps, with the bottle-neck (Vx = 0.8V), from the side (A) and above (B).

The electron density map shows many prominent trends, one of the most notable being the absence of any electrons within the boxes. This confirms the the code was running properly and no electrons started in the boxes or went through them at any point. Another trend seen is the electron density peak just to the left of the boxes & bottle-neck. This can be explained by the large number of electrons bouncing back off of the left side of the boxes', then continuously being pulled back towards the boxes (in the positive x-direction) by the electric field. Another phenomenon observed was the overall greater electron density in the left half of the area compared to the right half. This is also due to similar reasons as the previous trend. The electric field is inducing a net drift current in the positive x-direction (to the right of the screen). Therefore, once electrons get to the right of the boxes, there is nothing stopping them from simply wrapping back around to the left. However, once on the left side of the boxes, they go back and forth between being pulled to the right and then being bounced back to the left by the boxes. This results in the electrons spending more time on the left half of the area, therefore leading to a higher electron density on that side.

b)

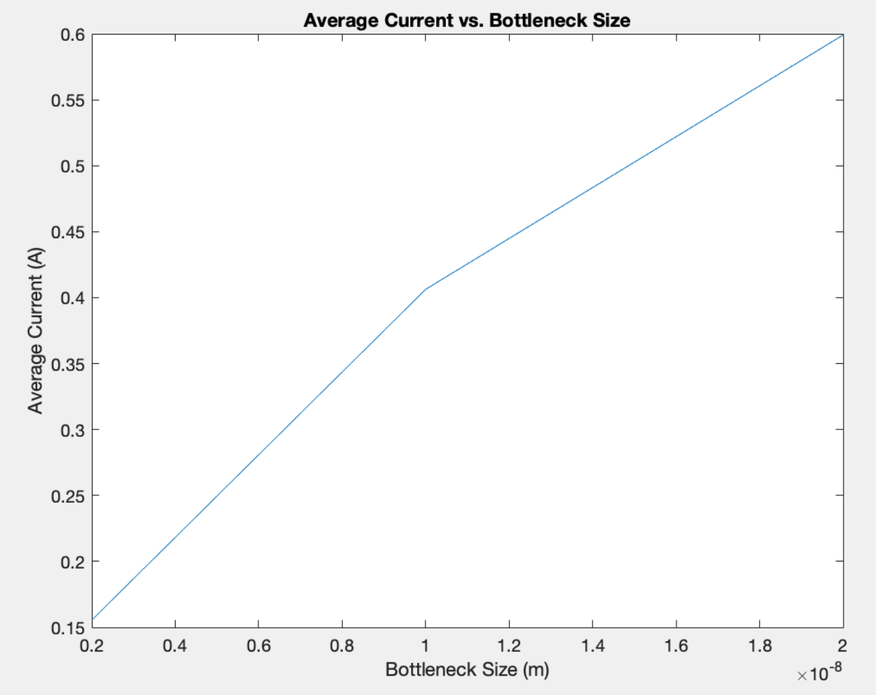


Figure 9: The average current of the electrons against the size of the bottle-neck of the device.

The plot above shows the relationship between the average electron drift current and the size of the bottle-neck. Unsurprisingly, the current was found to increase along with the size of the bottle-neck. This direct linear trend can be explained by the electrons being able to reach a higher steady state drift current when there were less obstacles blocking the direction of the current.

c) During assignments 1 & 3, it was noticed that a greater number of particles and a longer simulation time lead to more accurate results. Once the correct code is implemented, changing these 2 parameters can be fairly simple to do. In addition, it was noted in assignment 2 that the smaller the mesh sizes were, the more accurate the model was.
In terms of simulating a semiconductor, this model could be much more accurate by creating more 'boxes' within the area to more closely represent the size, shape and conductivities of each component of a semiconductor. In addition, adding a second carrier and also including its concentration in each area of the device could allow for a more accurate representation of a semiconductor.