Megan McEwen

Student # 101003509

ELEC 4700

15 March 2020

ASSIGNMENT #3: MONTE CARLO/FINITE DIFFERENCE MODEL

INTRODUCTION

This assignment combines two models developed in previous assignments: the Monte-Carlo simulation with a bottleneck, and the same bottleneck system using finite difference method. Section 1 looks at how particle trajectories change when the Monte Carlo method is applied to a static electric field, as well as electron density and temperature mapping under these conditions. Section 2 uses the finite difference method to generate inputs for the Monte Carlo simulation, and observes how the particles flow in the presence of the bottleneck. Finally, Section 3 uses the combined model to consider the device and extract density and current parameters.

1.0 MONTE CARLO WITH STATIC ELECTRIC FIELD

The first part of this assignment looks at the Monte Carlo method with a static electric field. To start, a voltage of 0.1V is applied across the x dimension. The electric field in the x direction is then calculated as follows:

```
%calculate electric field
voltX = 0.1; %voltage in x dimension
voltY = 0; %voltage in y dimension

%set up EField
EFieldX = voltX/xrange
```

Where xrange is 2e-7m, the length of the device area. Using this calculation, for a voltage of 0.1V, the electric field in the x direction is **500kV/m.**

Next, the force on each electron is determined. Since the electric field in the Y direction is still 0 for this section, we only need to consider the force in the X direction:

```
%calculate force on electrons due to field
Fx = EFieldX*q;
```

Where q is the electron charge ($1.6*10^{-19}$). Using this, Fx is calculated to be **0.08pN.**

The acceleration of each particle is then updated in each direction as follows:

```
%calculate acceleration on electrons
AccX = Fx/m0;
AccY = Fy/m0;
```

Since this part uses a static electric field, the acceleration is a constant number and does not need to be updated in the loop.

The final map of particle trajectories (25 particles, 100 iterations) is shown in Figure 1 below. Note that the curve in the trajectory is the response to the electric field pushing the electrons.

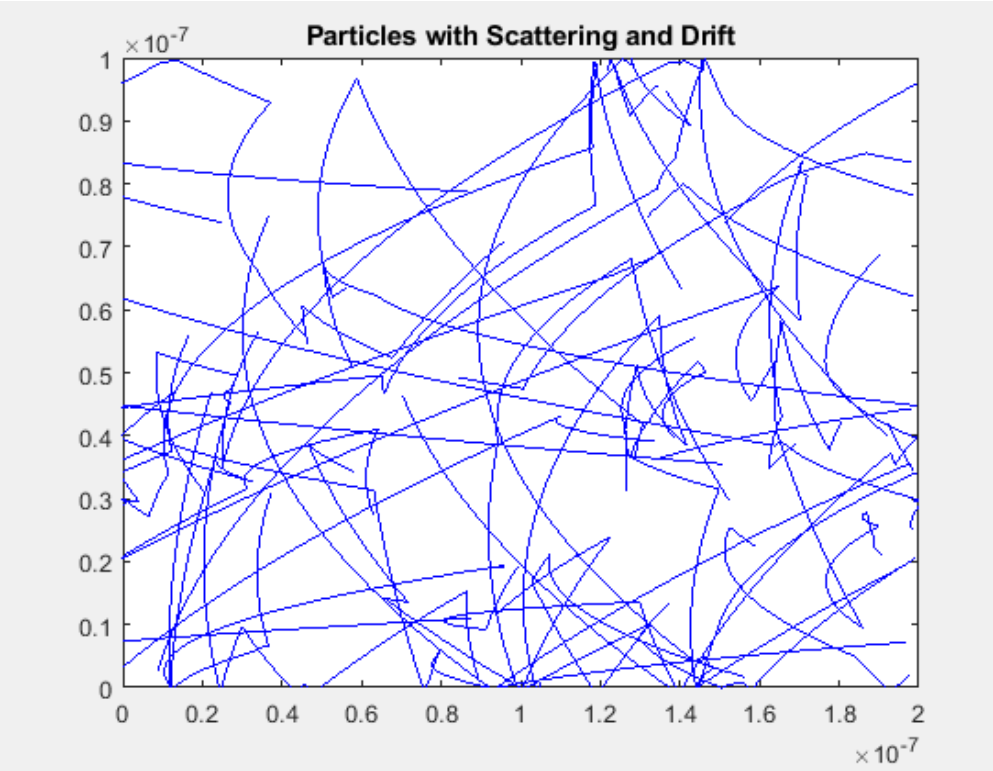

Figure 1: Particle Map with Static Electric Field

Next, the relationship between the electron drift current density and the average carrier velocity is explored. The formula for current is shown below:

```
%calculate current
   mu = VAvg/EFieldX;
   I(i) = q*mu*EFieldX*eConc/(xrange*yrange);
```

Where q is the electron charge, mu is the average velocity divided by the electric field, and eConc is the electron concentration. This is plotted over time in the x direction. Figure 2 below shows current over time in the x direction for 100 particles and 100 iterations:

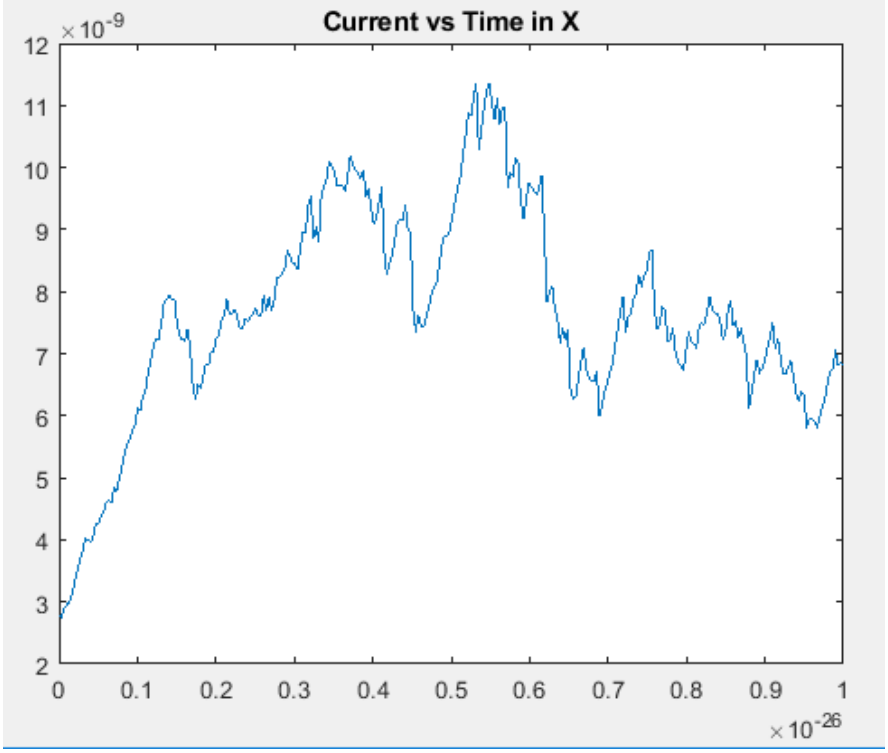

Figure 2: Current vs Time in X Direction

As the figure shows, the current climbs over time initially, then stabilizes.

Electron density and temperature plots are also generated, using the code below:

```
%temperature and electron density maps
x_region = linspace(0,xrange,10);
y_region = linspace(0,yrange,10);

x_bin = discretize(Px,x_region);
y_bin = discretize(Py,y_region);

temp_bin = zeros(10,10);
e_bin = zeros(10,10);

for i = 1:10
    for j  = 1:10
        inI = x_bin == i;
        inJ = y_bin == j;
        inBin = inI & inJ;

        sum_i = sum(Vx(inBin))/(dt/vTH);
        sum_j = sum(Vy(inBin))/(dt/vTH);

        sum_e = sum(inBin);

        avg = sqrt((sum_i)^2 + (sum_j)^2);

        temp_bin(i,j) = (m*0.5*avg.^2)/(k*2);
```

```
        e_bin(i,j) = sum_e;
    end
end

figure(3)
surf(e_bin);
title(['Electron Density Map after ', num2str(iter),' iterations for '
,num2str(n),' particles']);
colorbar

figure(4)
surf(temp_bin);
title(['Temperature Map for ',num2str(n),' particles']);
colorbar
```

The results of this for 100 particles and 300 iterations can be seen in Figures 3 and 4:

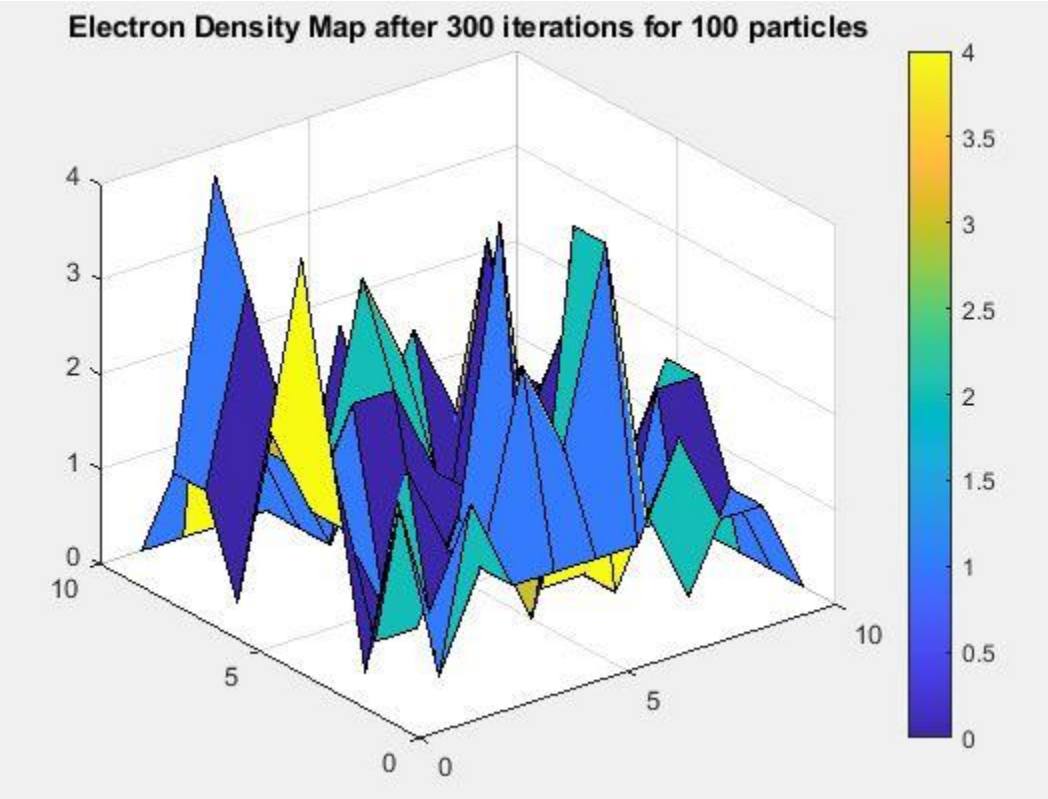

Figure 3: Electron Density Map

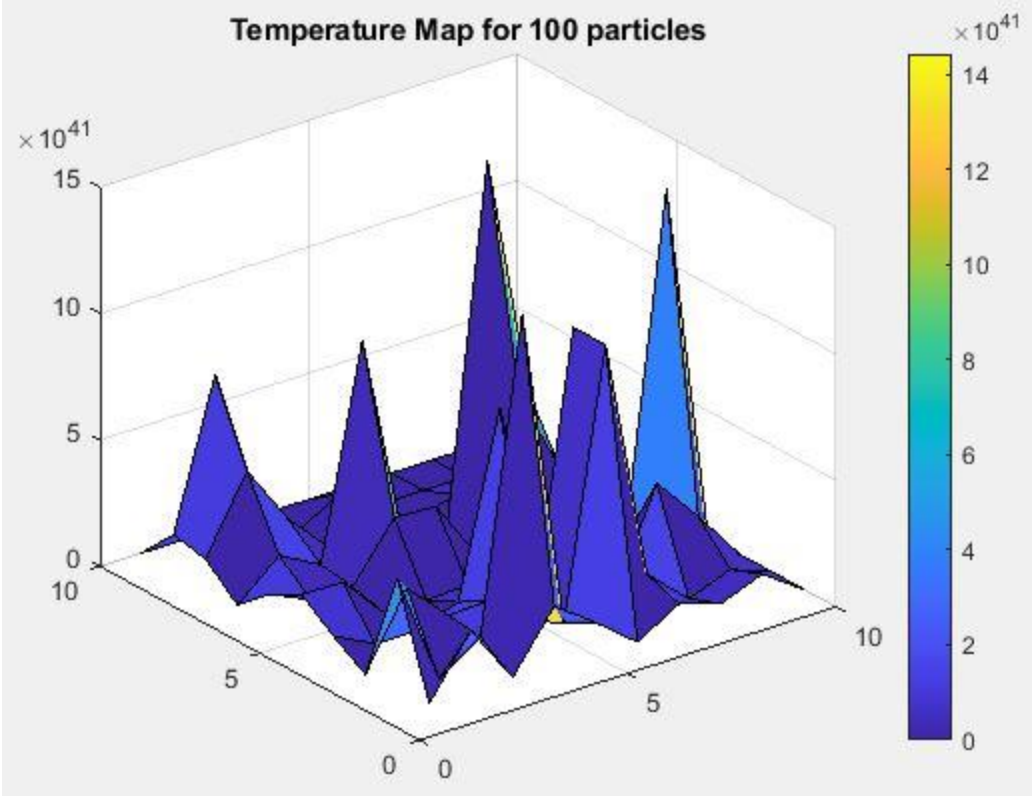

Figure 4: Temperature Map

2.0 FINITE DIFFERENCE METHOD WITH MONTE CARLO

This section re-uses the code from Assignment 2 to generate an electric field using the Finite Difference Method. The bottleneck is also added to this part of the code.

The Assignment 2 report contains details on how the G matrix is set up and the finite difference solution is generated. The bottleneck is inserted by creating contacts with a lower conductivity within the grid space. The resulting voltage potential map can be seen in Figure 5 below:

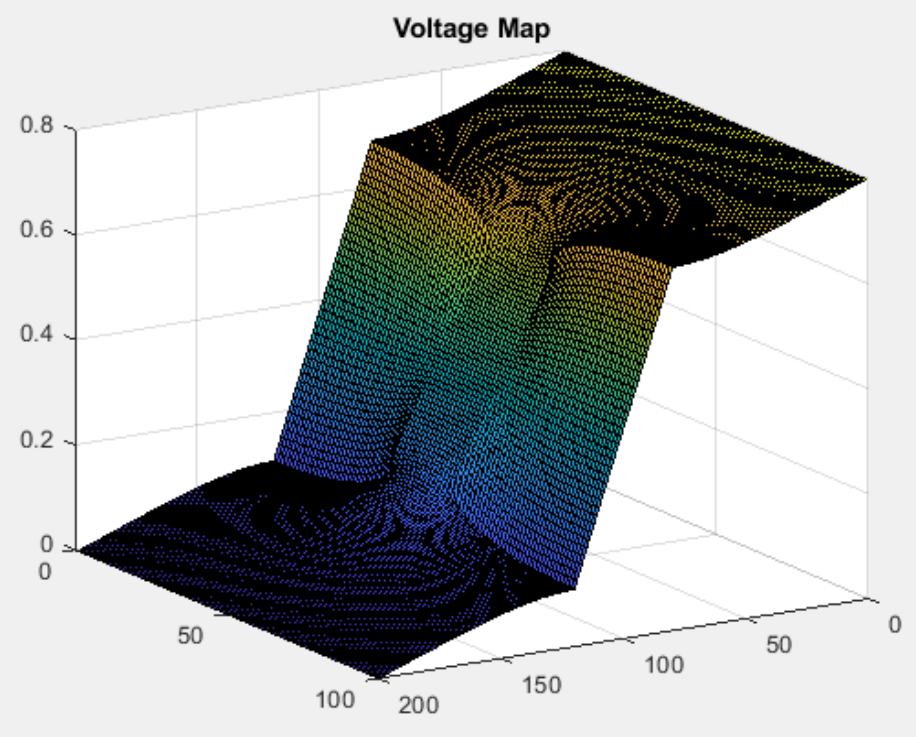

Figure 5: Voltage Map with Bottleneck

The electric field is then calculated from the potential as shown, using 'quiver' to generate the electric field vectors:

```
%solve for E
[Ex_FDM1, Ey_FDM1] = gradient(-VMatrix*1e9);

figure(3)
quiver(Ex_FDM1,Ey_FDM1)
title('Electric Field Plot');
view(0,90);
```

The electric field plot is shown in Figure 6:

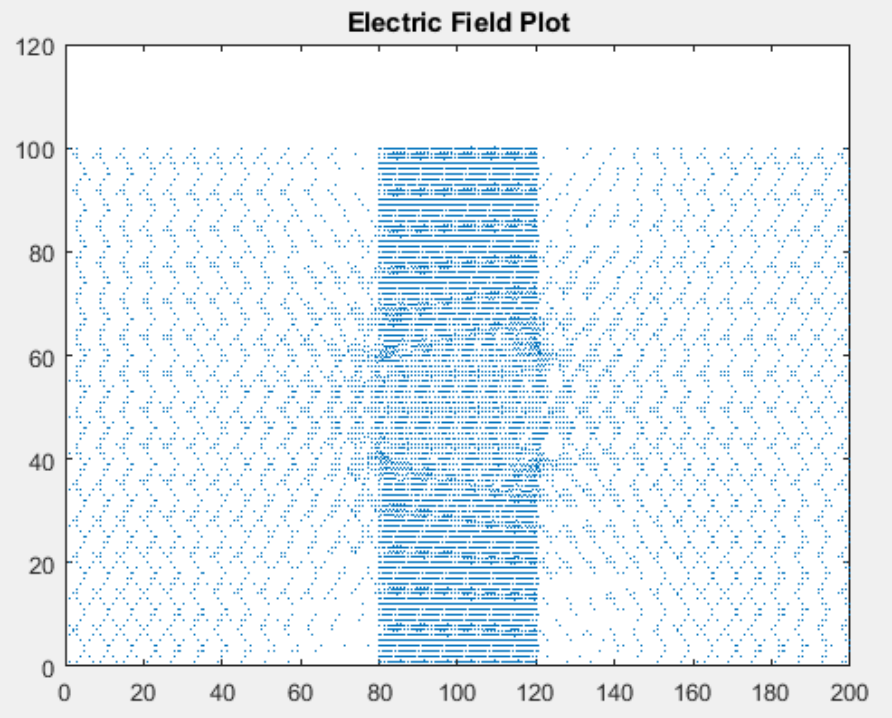

Figure 6: Electric Field Plot

Finally, this code is combined with the Monte Carlo code. The finite difference method is used to generate an electric field, which is then used as an input to determine acceleration for particles in the Monte Carlo simulation. The electron trajectories are then plotted as before. Figure 7 shows a map with 50 particles over 200 iterations:

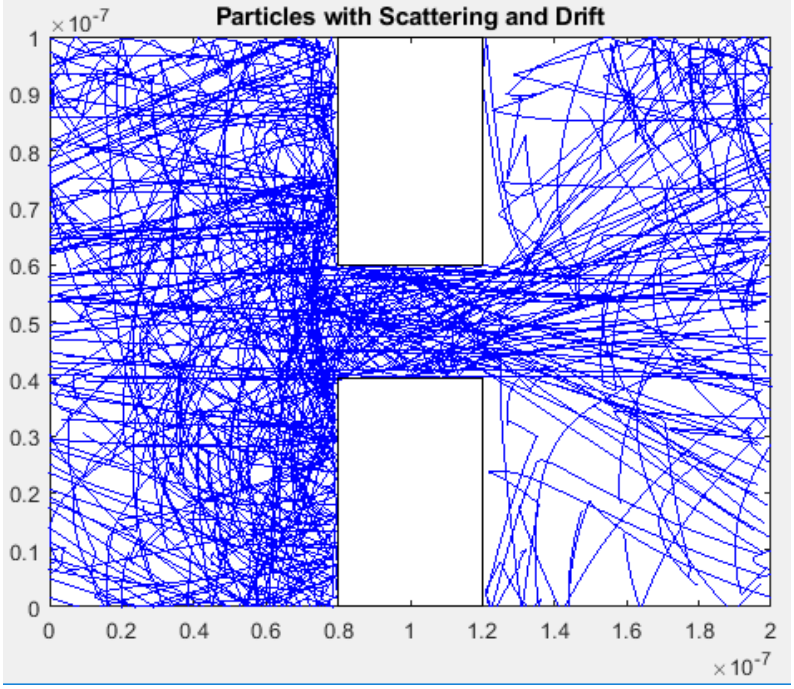

Figure 7: FDM and Monte Carlo Particle Map (50 particles, 200 iterations)

As expected, the particles are pushed toward the right due to the drift current. As particles arrive on the left side, they become trapped by the bottleneck, with fewer particles getting through to the other side.

3.0 DENSITY AND CURRENT MODELLING WITH COMBINED MODELS

This section investigates the device created with the finite difference method/Monte Carlo and extracts some simple parameters.

First, the field is set with V = 0.8V, and the electron density map is generated. Figure 8 shows the electron density map for 500 particles after 100 iterations:

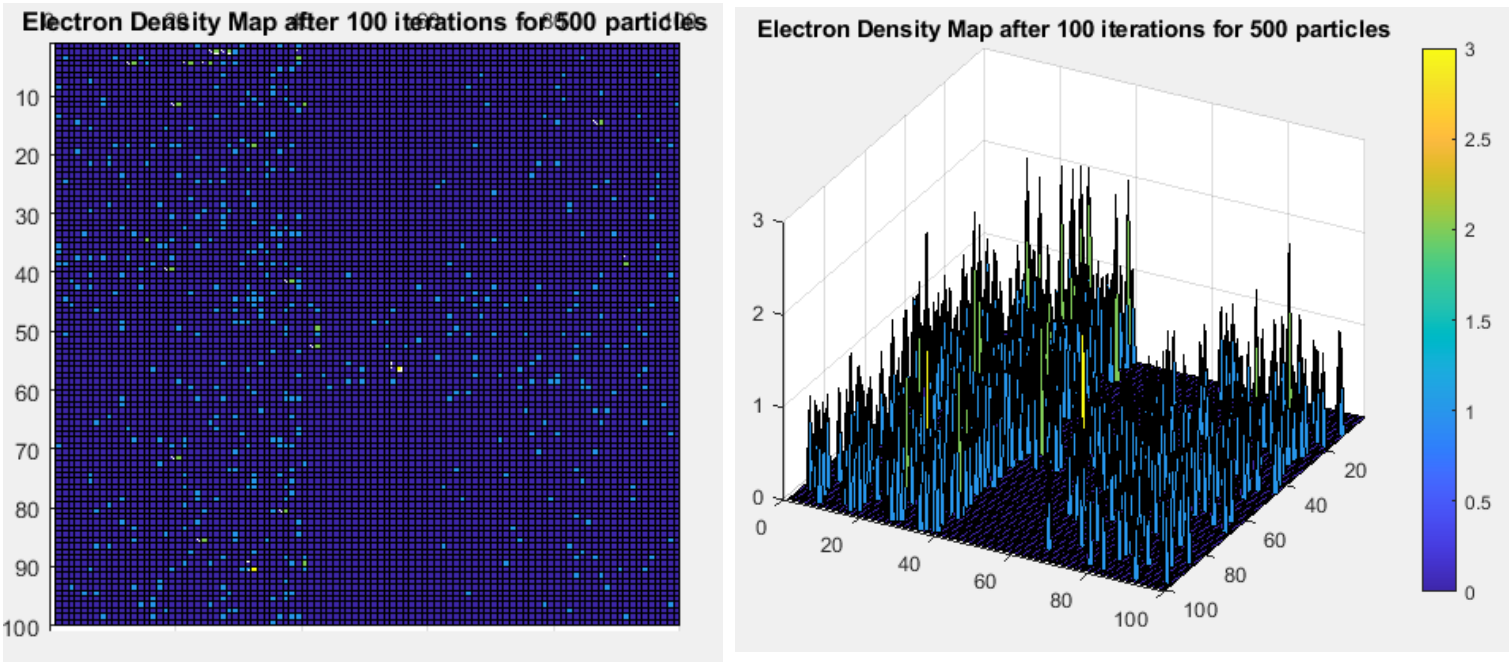

Figure 8: Electron Density Map (V = 0.8V, 500 particles, 100 iterations)

As expected from the voltage map plot in Part 2, there is a significantly higher electron density on the left side of the bottleneck. As the particles move, few make it through the bottleneck and end up on the other side, and the ones that do drift back to the left side over time. No particles are within the bottleneck.

To make this simulation more accurate, the number of electrons/number of iterations can be increased. Increasing the mesh density also improves accuracy.