1.0 Monte-Carlo Simulator

The electric field was calculated using a constant voltage of 0.5 V. In turn, we can calculate force and acceleration using the code below:

```
%ASSINGMENT 3 CODE
voltage = 0.5;
efield = voltage / xlimit; % efield = 2500000
force  = efield * q; % force = 4.0050e-13
a = force/mn; % a = 1.6909e+18
```

Now we can add acceleration each time step of the simulation using the code below.

```
init(:,5) = (init(:,5) + a*timestep); %Adding effect of an Ex field
```

The resulting simulation for 5 particles is shown below with 2% scattering potential.

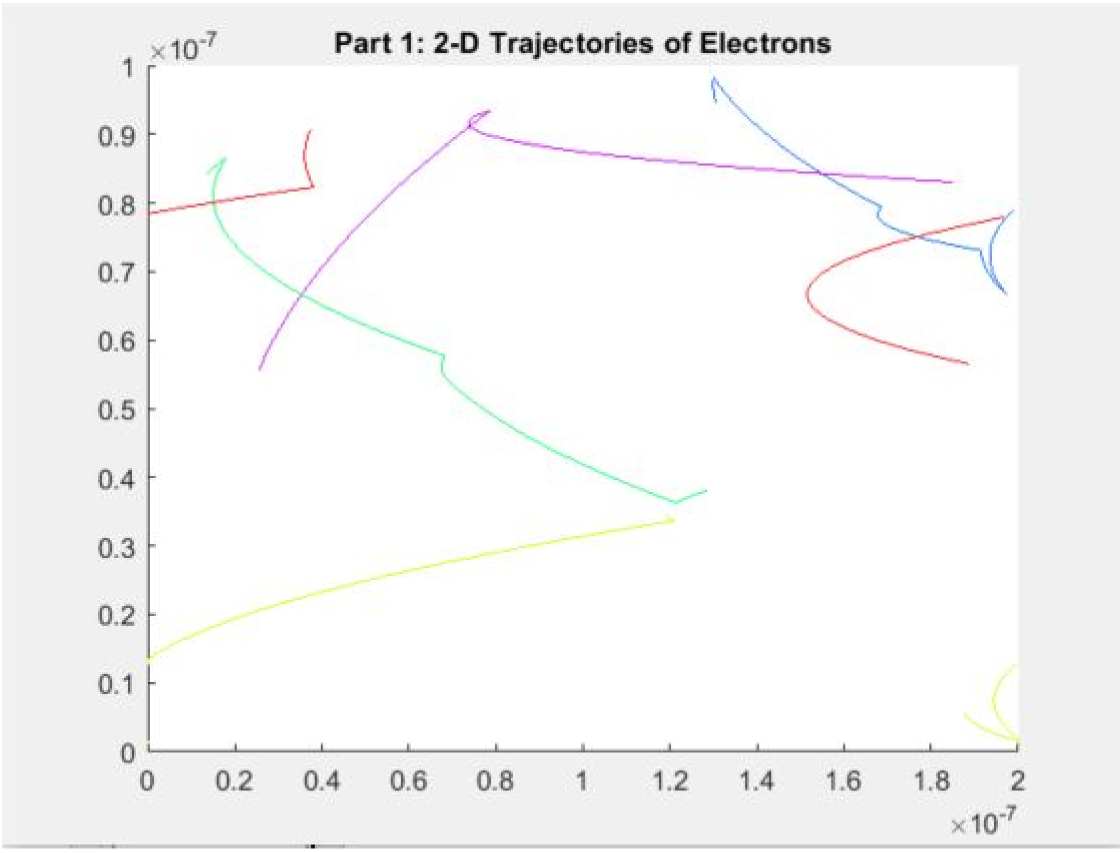

Figure 1: Monte-Carlo Simulation with Efield

From this we can extract the temperature plot for the whole simulation which is shown in figure 2.

```
    for g=1:numpart
    temp = ((init(g,5).^2+init(g,6).^2) *mn )/(2* kb);
    temp = temp + temp;
    end
       average = temp/numpart;
       T(1,count) = average;
       time = time + timestep;
       timeaxis(1,count) = time;

   count = count +1;

end

    figure(3);
    plot(timeaxis,T);
    title 'Temperature Plot over time';
    ylabel 'Temperature (K)';
    xlabel 'time (s)';
```

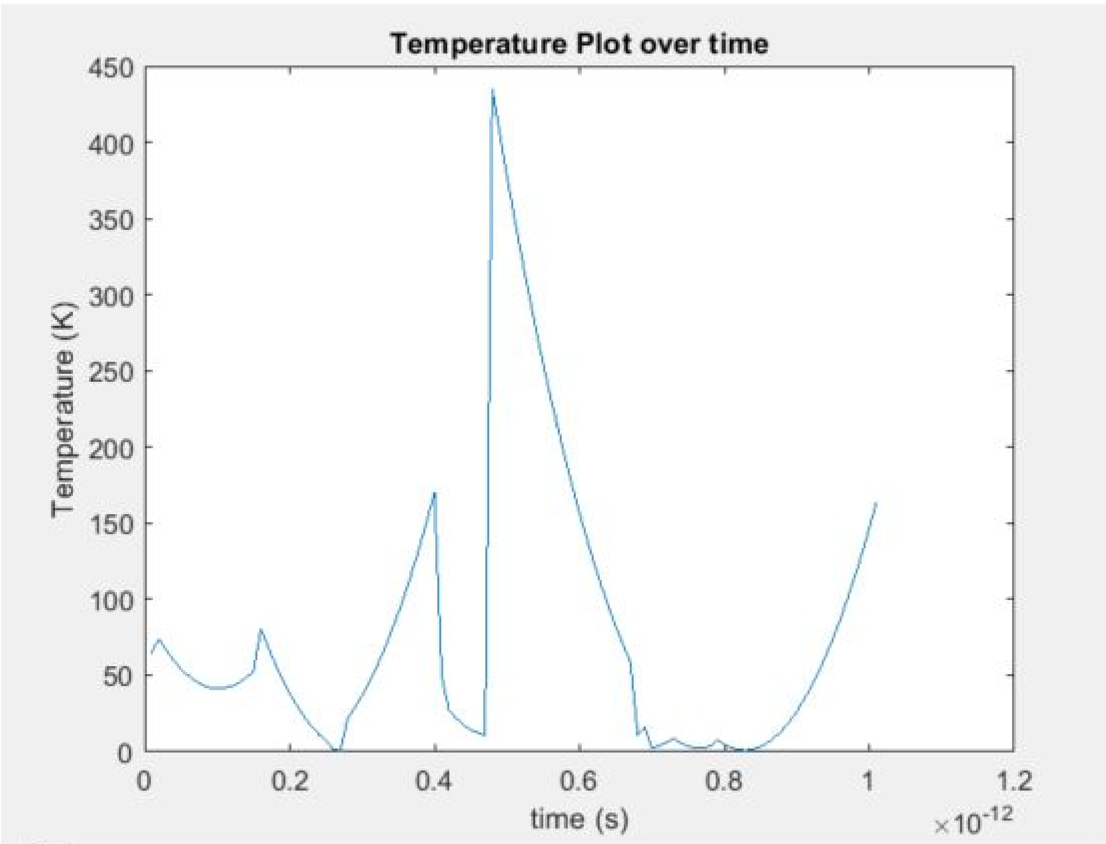

Figure 2: Temperature Plot over Time

2.0 Finite Difference Method

Shown in the following figures is the surface plot of V(x,y) and the 2D electric field plot. The G matrix was used to calculate the potential of a conductivity map. The results below are displayed and were as expected.

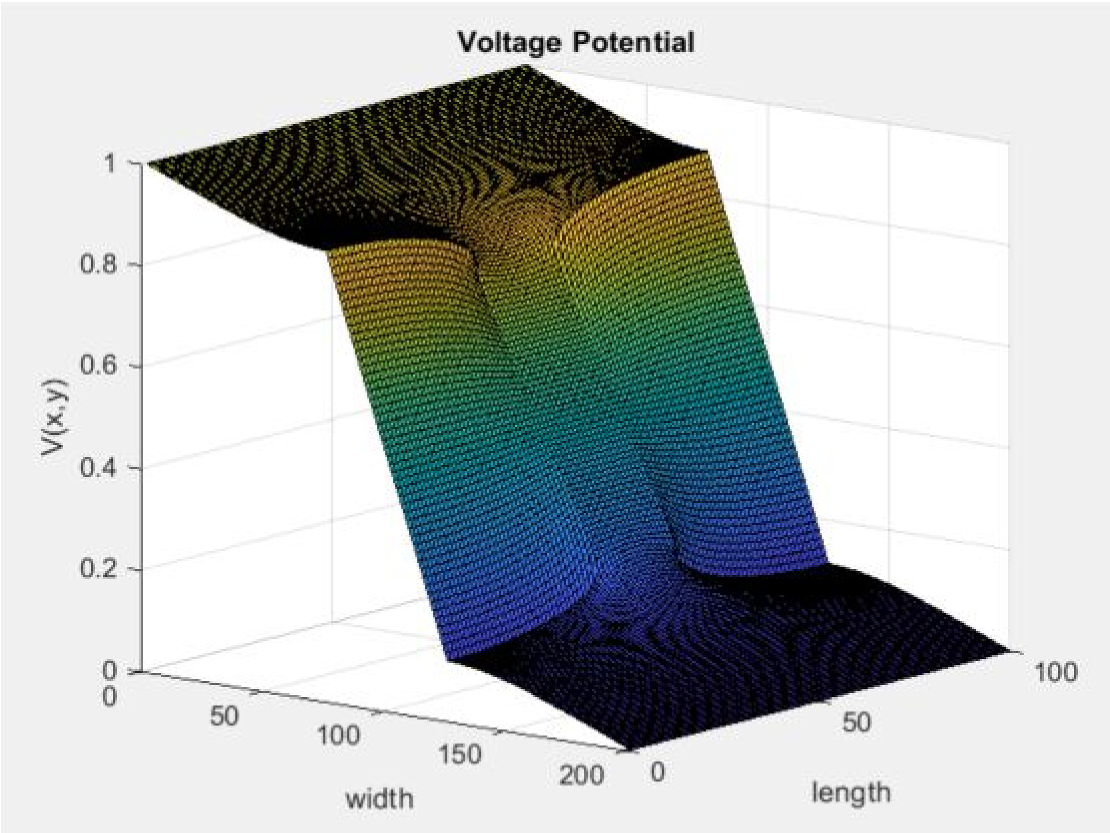

Figure 3: Voltage Potential with Bottle Neck

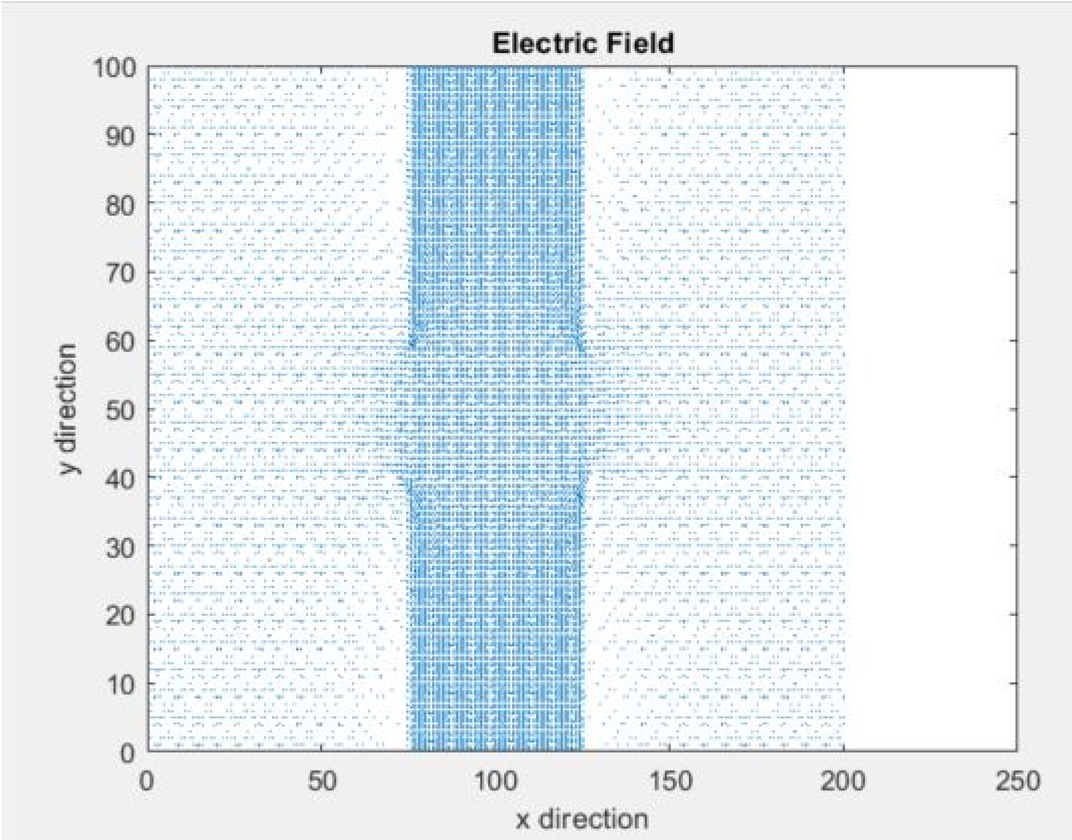

Figure 4: Electric Field Plot

3.0 Monte-Carlo and Finite Difference Method

Below is the plot for the simulation with a constant electric field applied across with a 2% scatter. The exact same approach was used as in question 1.

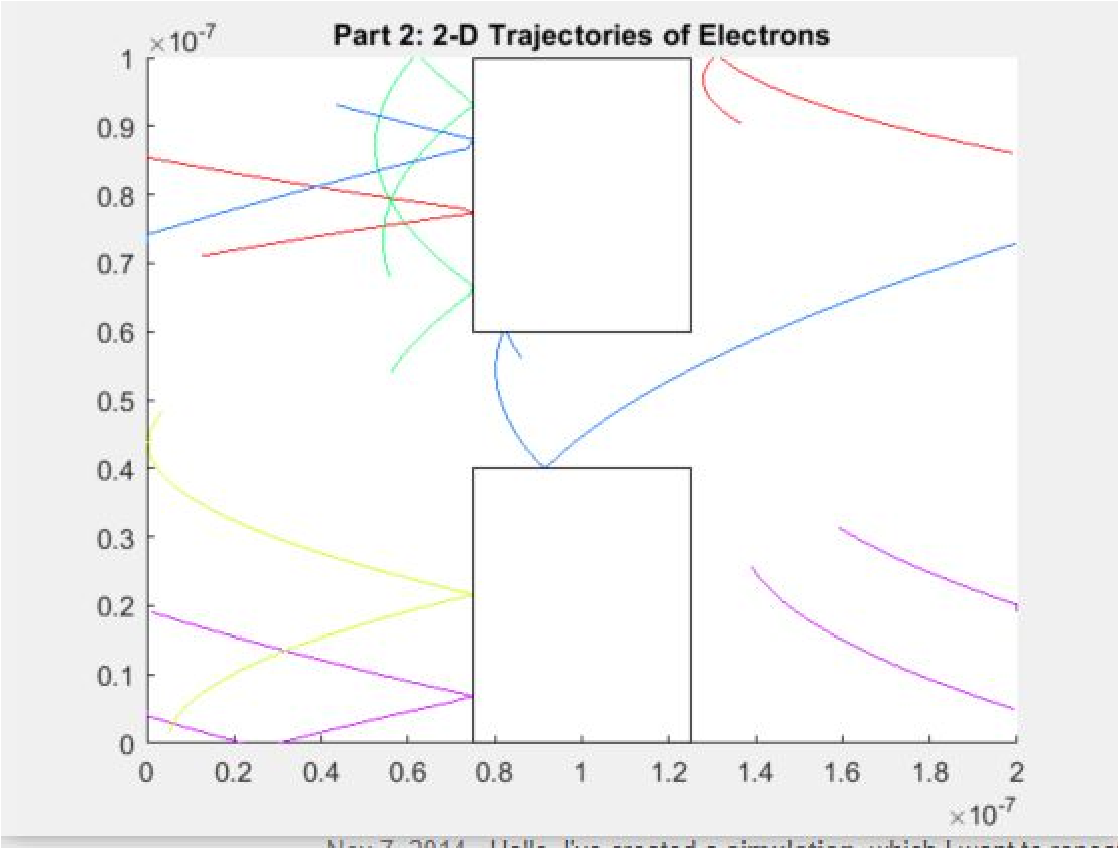

Figure 5: Monte-Carlo simulation with constant Efield

Next, instead of applying a constant electric field across the simulation, we now need to apply the electric field that was found in question 2 to the simulation. Unfortunately, I could not get the simulation to work but I will explain the process I used to attempt the problem.

First, we need to generate a vector for the electric field which is shown below in the following code:

```
%ASSINGMENT 3 PART 3 CODE
voltage = 0.5;
[Ex,Ey] = gradient(X);
forcex  = Ex * q;
forcey = Ey *q;
ax = forcex/mn;
ay = forcey/mn;
```

Now we have a vector for acceleration in the x and y directions. Now instead of applying the acceleration to only the x and it being constant, the acceleration will have an x and y component and be different depending on the position of the particles in the simulation. Now the accelerations are now two separate vectors of values that vary depending on the location. What I decided to do was locate the position of each particle, and translate that position on the simulation to a specific index in the acceleration vectors. The values of acceleration should be added to the x and y components of velocity. This is shown in the following code attempt below.

```
for m=1:1:numpart
roundx = round(init(m,1)*10^9); %round the x position to get an index
roundy = round(init(m,2)*10^9); %round the y position to get an index


init(m,5) = (init(m,5) + ax(roundx,roundy).*timestep);
init(m,6) = (init(m,6) + ay(roundx,roundy).*timestep);
end
```

So there we can see that according to the position on the simulation, the particle will have an according x and y component of acceleration which can be located using the following code.