Assignment 3: Month-Carlo/Finite Difference Method

```
% Connor Warden
% 10107819
```

Part 1

Question A

The electric field imposed on the electrons was determined using the following lines of code.

```
% ASSIGNMENT 3
% 1A - E field
vol_x = 0.1;    % 0.1V applied across the x dimension
vol_y = 0;      % 0V applied in the y direction

% E field is V/d for a constant field
E_x = vol_x/width; % voltage/width of field
E_y = vol_y/height; % voltage/height of field

E_val = sprintf(num2str(E_x)); % turns the reslt into a string
fprintf('The electric field of x with a voltage of 0.1 is %s V/m\n',E_val)
```

This printed the result in the command window, shown below.

```
"The electric field of x with a voltage of 0.1 is 500000 V/m"
```

Question B

The force on each electron was determined using the electric field calculated in question A. The following code was used.

```
% 1B - Force on each electron

q = 1.6e-19; % Charge on electron
F_x = q*E_x; % Force is Charge * Electric Field
F_y = q*E_y; % Force is Charge * Electric Field

F_val = sprintf(num2str(F_x)); % Turns to string for output
fprintf('The force of each electron is %s N\n',F_val)
```

The result of this is shown below.

```
"The force of each electron is 8e-14 N"
```

Question C

Acceleration was determined using the force of each electron, shown below.

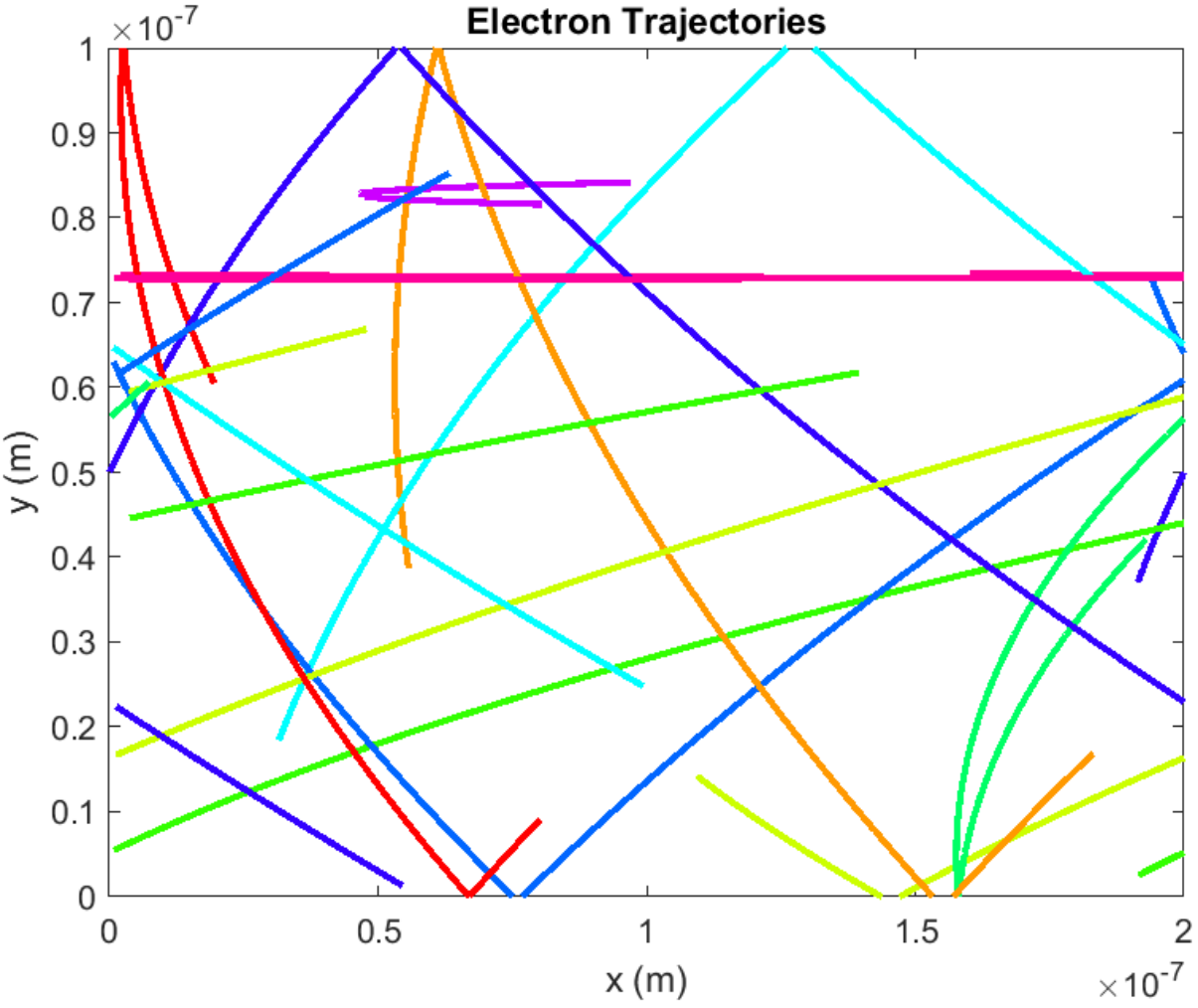
```
% 1C - Acceleration
```

```
a_x = ones(num_elec,1).*(F_x/m_n);    % We need the accelartion of each electron
a_y = ones(num_elec,1).*(F_y/m_n);    % for future calculation within the loop
```

This was then implemented within the solution loop, shown below.

```
for i = 1:step % Step is the number of iterations

    vx = vx + a_x*dt;   % accounts for acceleration
    vy = vy + a_y*dt;   % accounts for acceleration


    dx = vx*dt; % change in x

    dy = vy*dt; % change in y

    x = xp + dx; % updates the new x position
    y = yp + dy; % updates the new y position

    v_avg = sqrt(vx.*vx + vy.*vy);  % total avg velocity
```

This results in the following trajectory plot, shown after 100 steps.

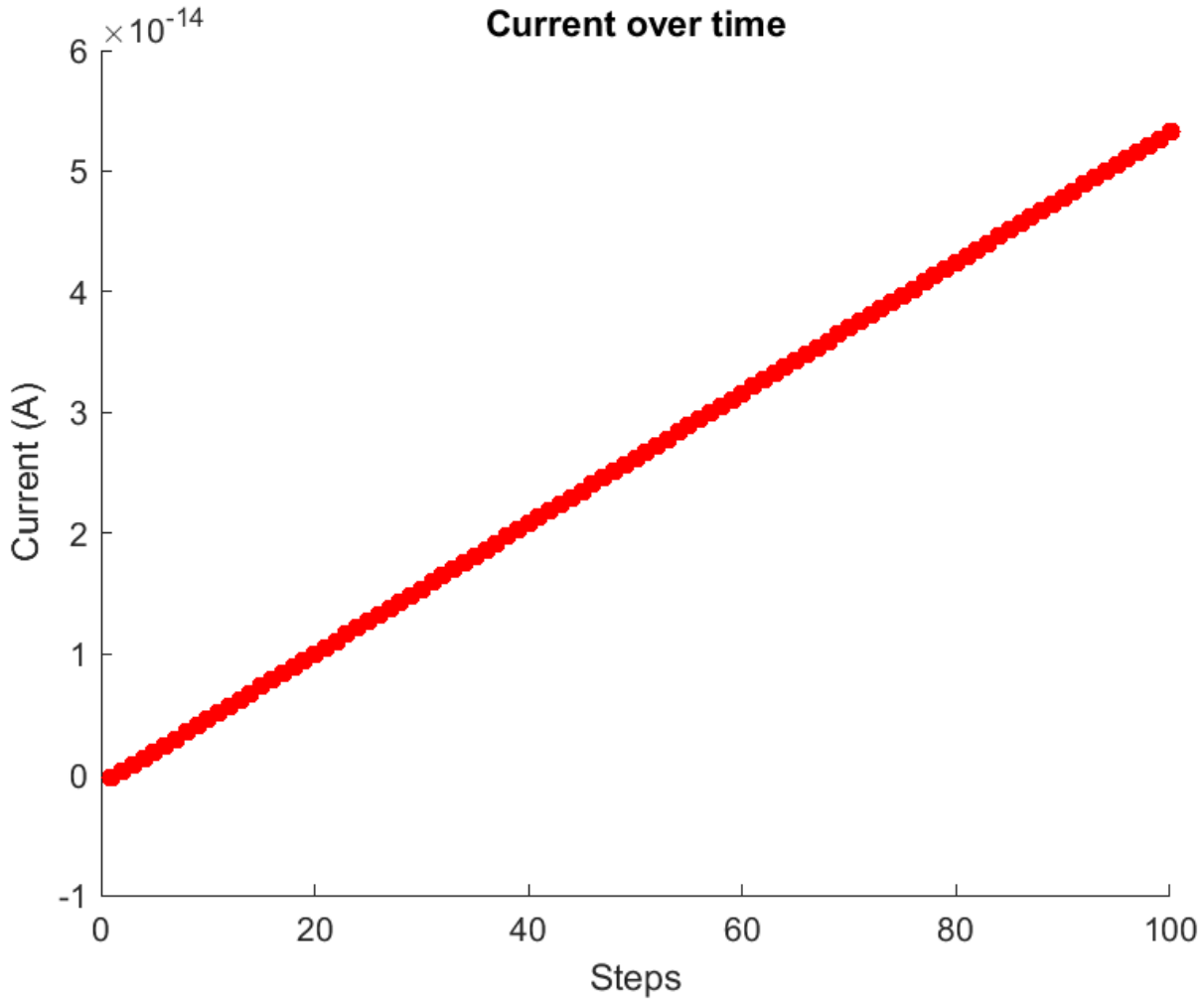


Question D

Current was determined using the following code.

```
%plot of current over time
% Current density = Q * Flux
% or               = enVdx = conductivity*q*vx
% Flux = number of particles crossing A/ area*time interval
e_con = 10^15*1000; % given
vd_x = sum(vx)/1000;
J = q*vd_x*e_con; % Current density = Q * Flux
I = J/e_con;
figure(6)
scatter(i,I,400,'r','.')
title('Current over time')
hold on
```

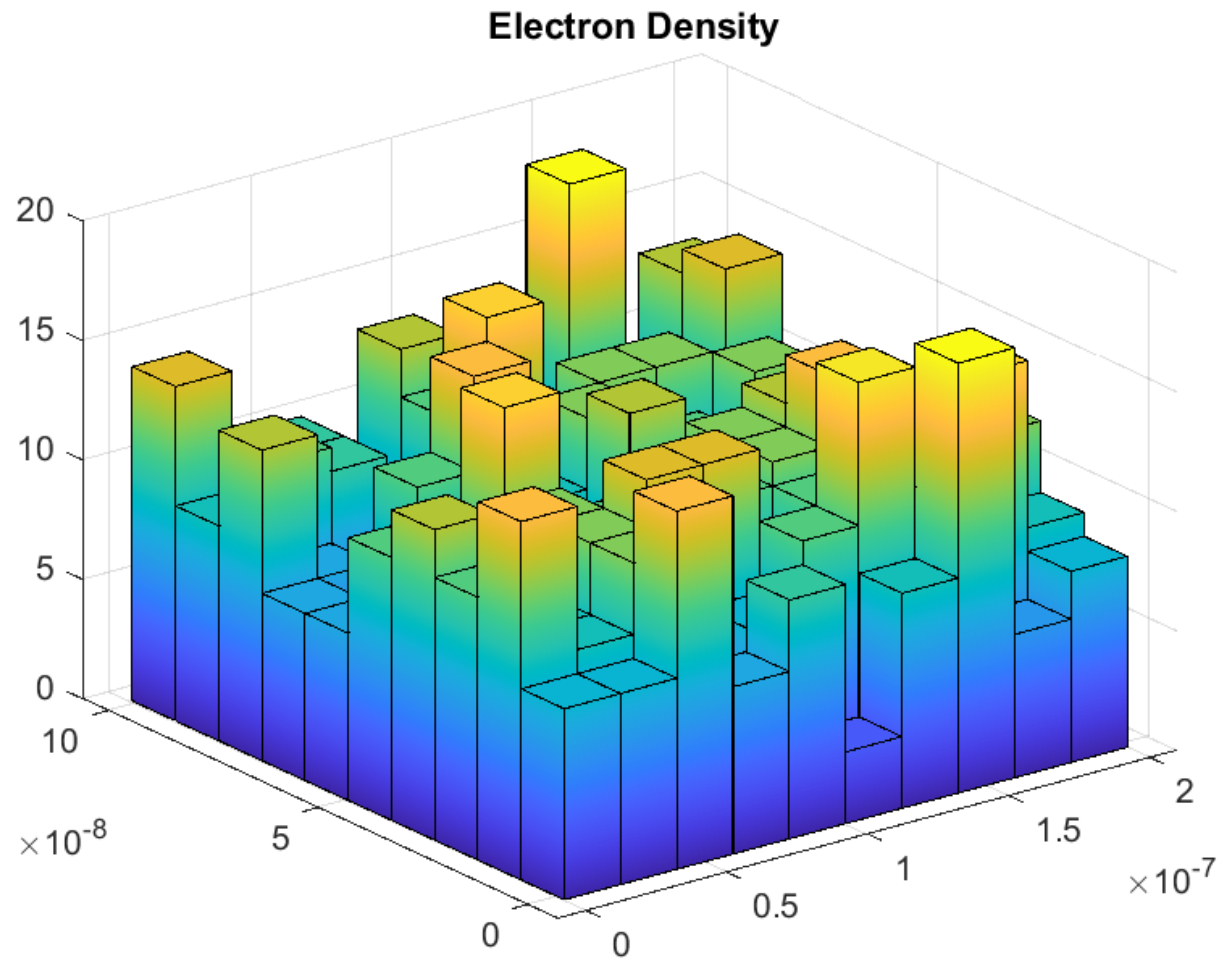
As acceleration is constant in this instance, and velocity is also constant, the current is expected to increase in a linear fashion. This is seen in the following plot, generated using the scatter function in the above code.

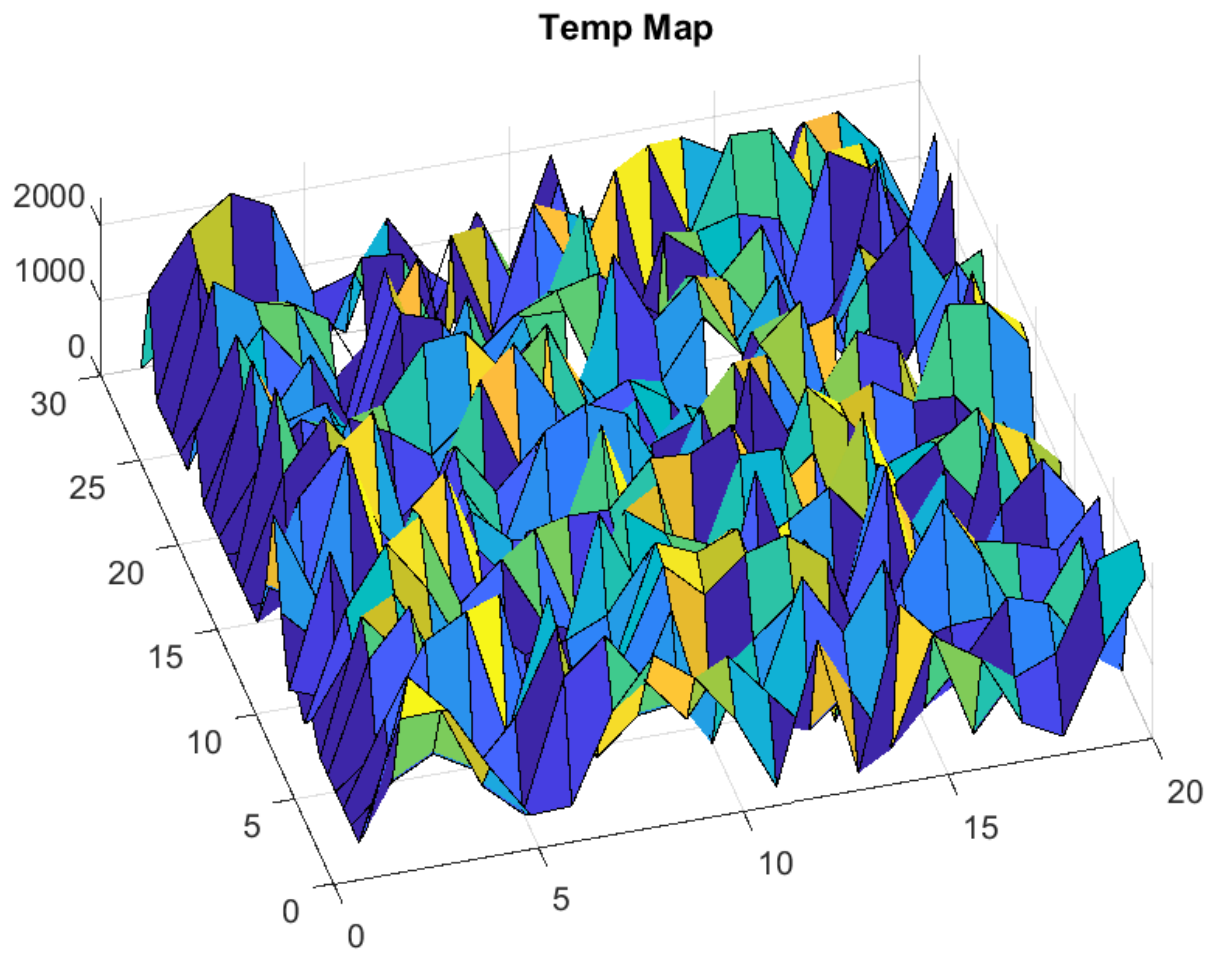


Question E

Density and temp plots were generated using the following code.

```
% Plots the Electron Density Map
figure(4)
hist3([x, y], 'CDataMode','auto','FaceColor','interp')
title(' Electron Density')

% Temp map
n_bins_y = 20;
n_bins_x = 30;

[Y, E_1] = discretize(y,n_bins_y);
[X, E_2] = discretize(x,n_bins_x);
temp_map = zeros(n_bins_x, n_bins_y);
for i = 1:n_bins_x
    for j = 1: n_bins_y
        temp_map(i,j) = nanmean(new_temp((X==i) & (Y==j)));
        if isnan(temp_map(i,j))
            temp_map(i,j) = 0;
        end
    end
end

figure(5)
surf(temp_map)
title("Temp Map")
```



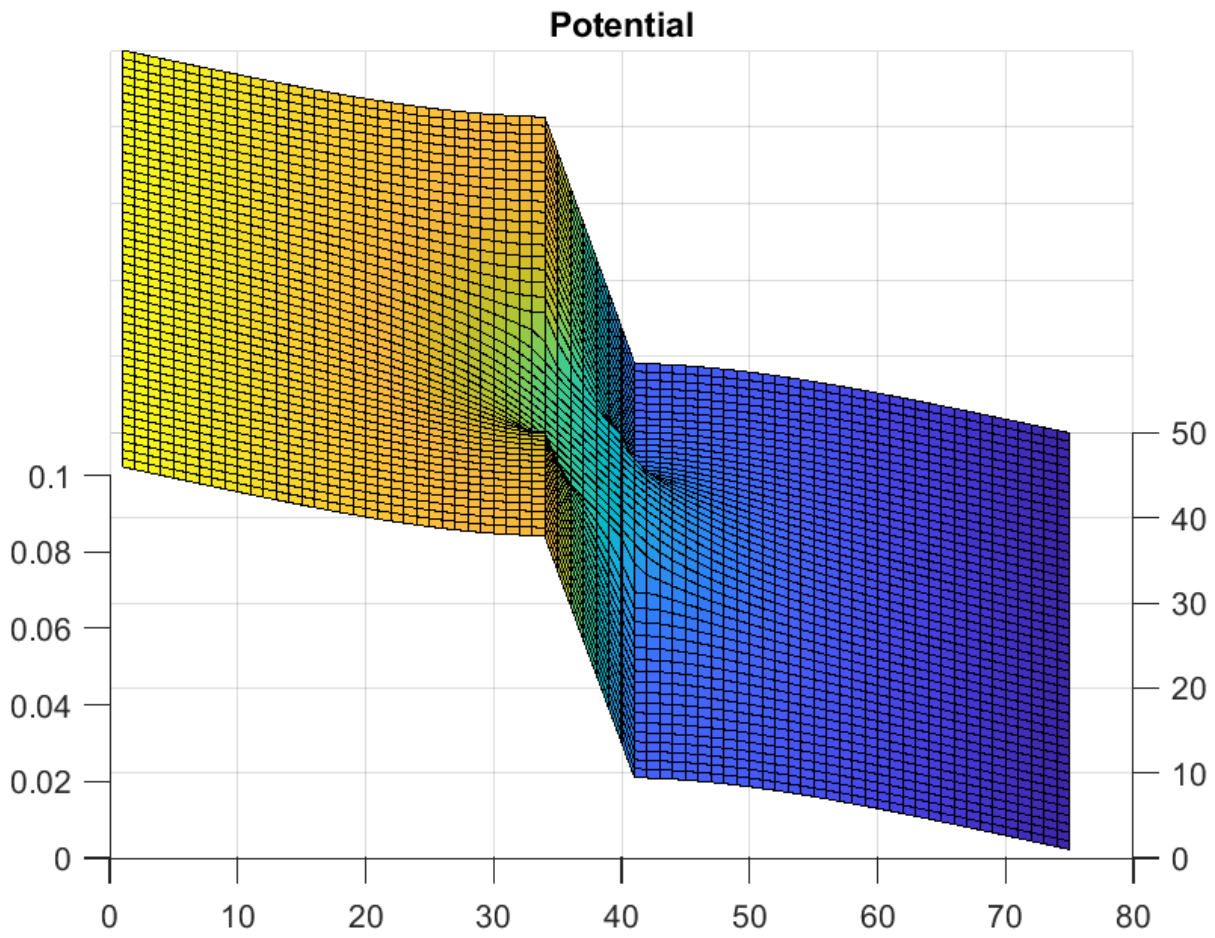


Part 2

Question A

Only one line of code was changed to determine the potential with 0.1V. This is shown below, along with the associated plot.

```
for i = 1:(nx)
    for j = 1:(ny)

        n = j + (i-1)*ny;

        if i == 1
           g(n,n) = 1;
           f(n) = 0.1; % This line was changed to 0.1V
```

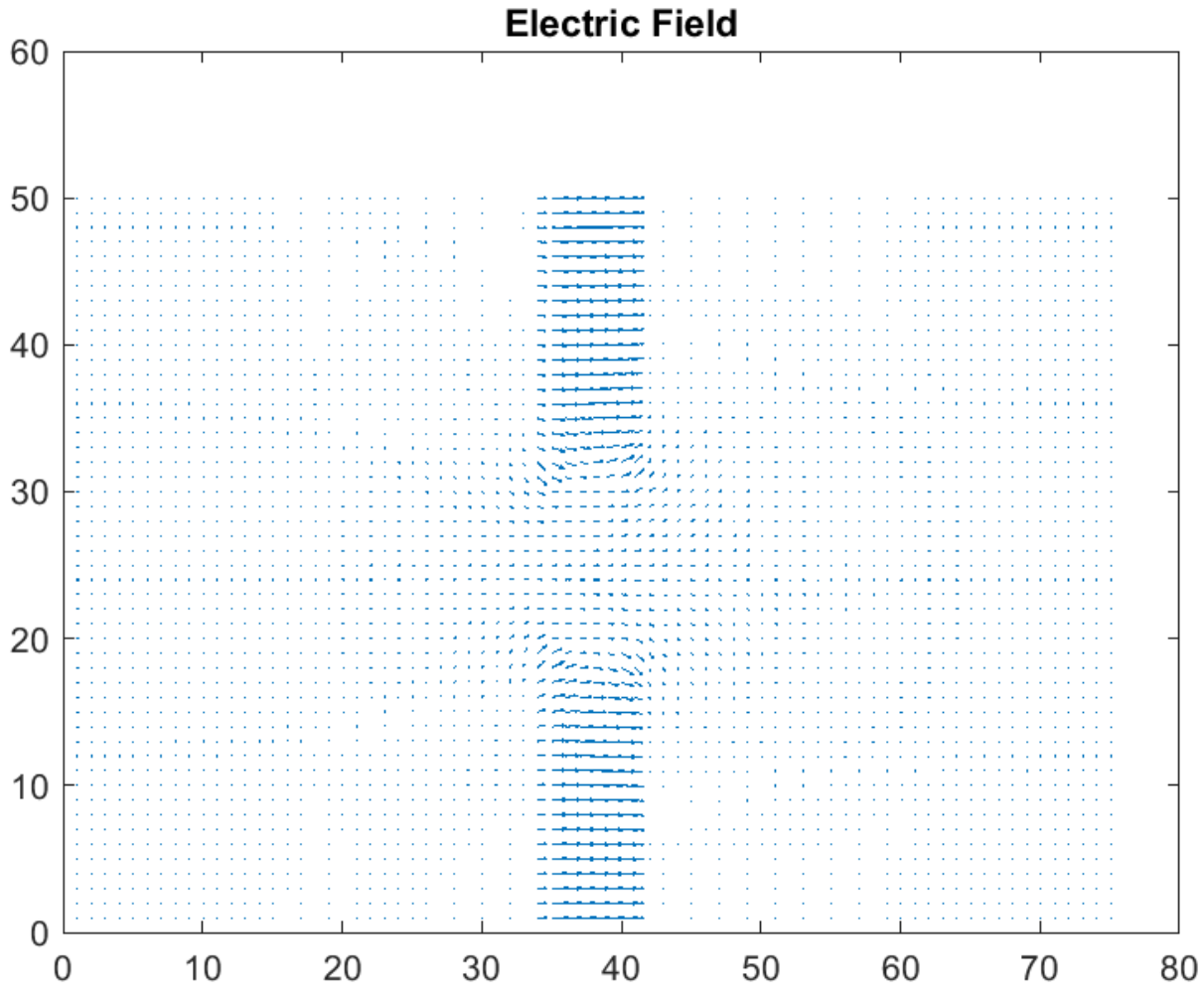


Question B

Nothing was changed from assignment 2 in order to calculate this. The following code is what is used to determine the E field and plot it.

```
vmap = zeros(nx,ny);
for i = 1:(nx)
    for j = 1:(ny)
        n = j + (i-1) * ny;
        vmap(i, j) = v(n);
    end
end

for i = 1:(nx)
    for j = 1:(ny)
        if i == 1
            Ex(i,j) = vmap(i+1,j) - vmap(i,j);
        elseif i == nx
            Ex(i,j) = vmap(i,j) - vmap(i-1,j);
        else
            Ex(i,j) = (vmap(i+1,j) - vmap(i-1,j))*(0.5);
        end
        if j == 1
            Ey(i,j) = vmap(i,j+1) - vmap(i,j);
        elseif j == ny
            Ey(i,j) = vmap(i,j) - vmap(i,j-1);
```

```
        else
            Ey(i,j) = (vmap(i,j+1) - vmap(i,j-1))*(0.5);
        end
    end
end

Ex = -Ex;
Ey = -Ey;

figure(3)
quiver(Ex', Ey'); % Quiver plots E field with respect to both directions of the field
title("Electric Field")
```



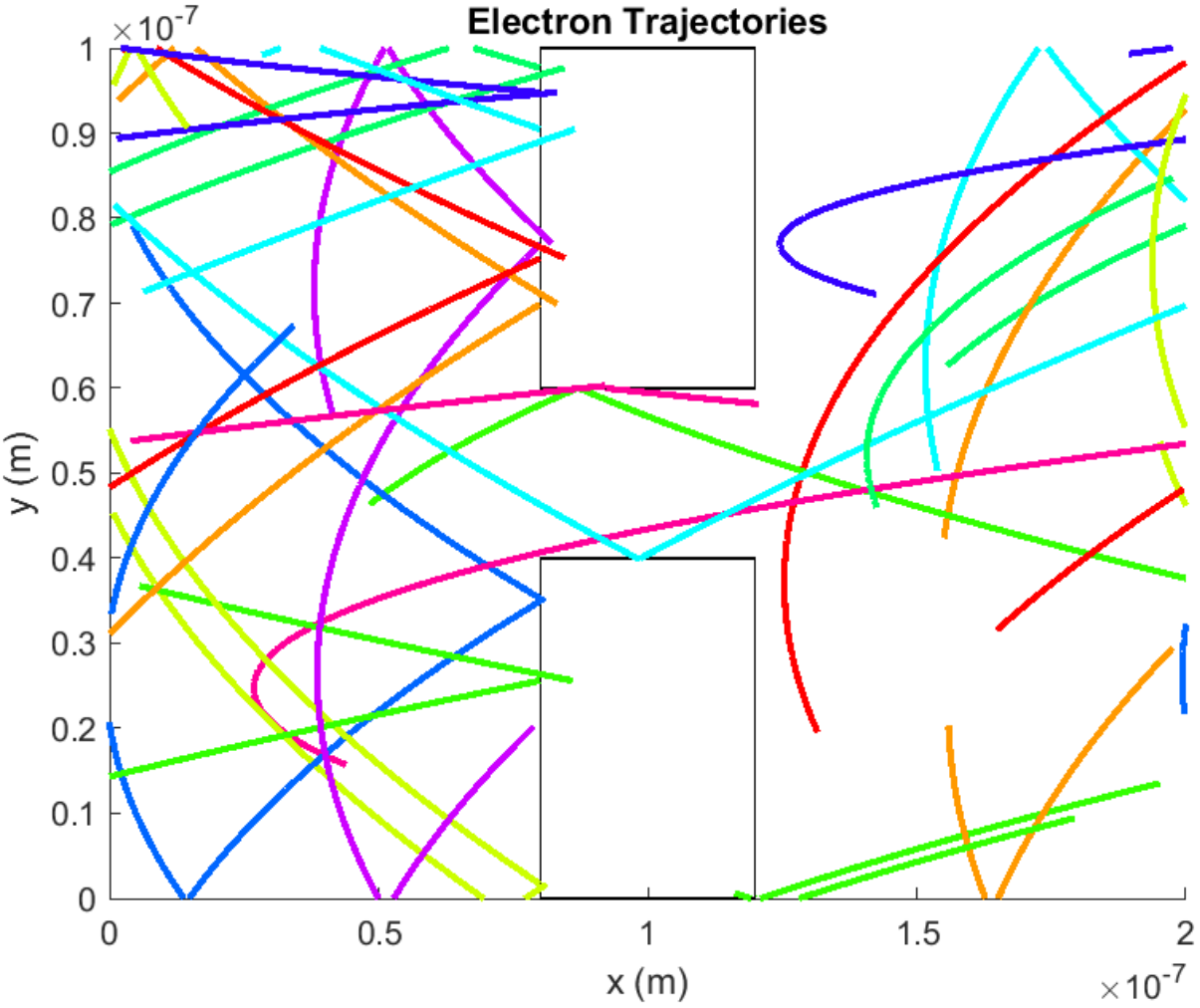
Question C

This question combined elements from part 1 and 2. Assignment 2 code was not changed in any way, and assignment 1 was only changed in determining the force on each electron. The E field from assignment 2 was used as an input to the force equations for assignment 1, which in turn are used to determine the acceleration and in turn velocity of each electron. The code changes are shown below.

```
% 1B - Force on each electron
q = 1.6e-19; % Charge on electron

F_x = q*(mean(mean(E_x))); % Force is Charge * Electric Field
```

```
F_y = q*(mean(mean(E_y)));

% Acceleration

a_x = ones(num_elec,1).*(F_x/m_n);
a_y = ones(num_elec,1).*(F_y/m_n);
```

You can see in the following plot that the electron trajectory is still curved. There are some errors due to the velocity of the electrons causing them to occasionally skip over boundaries, but overall the plot is correct.



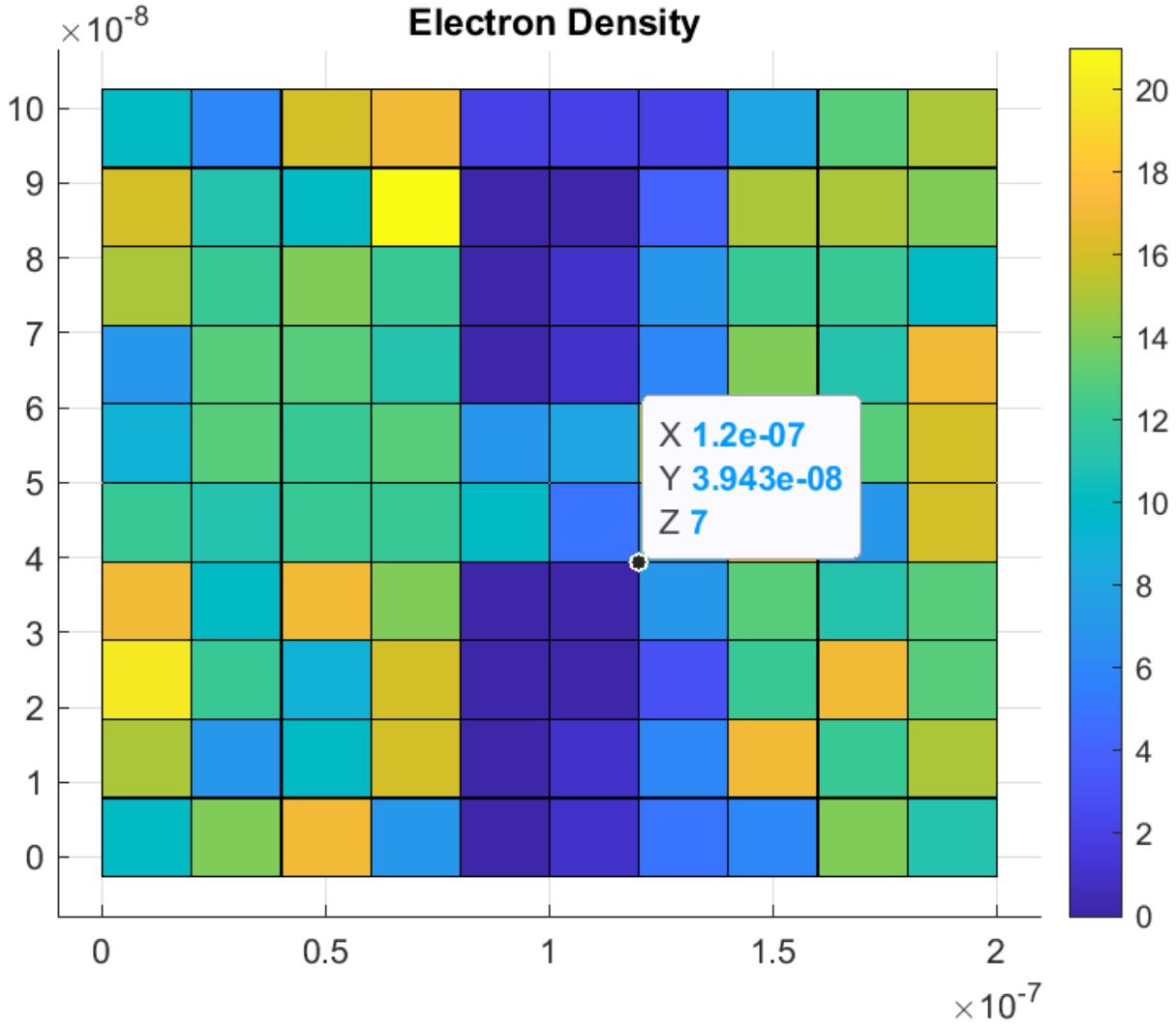
Part 3

Question A

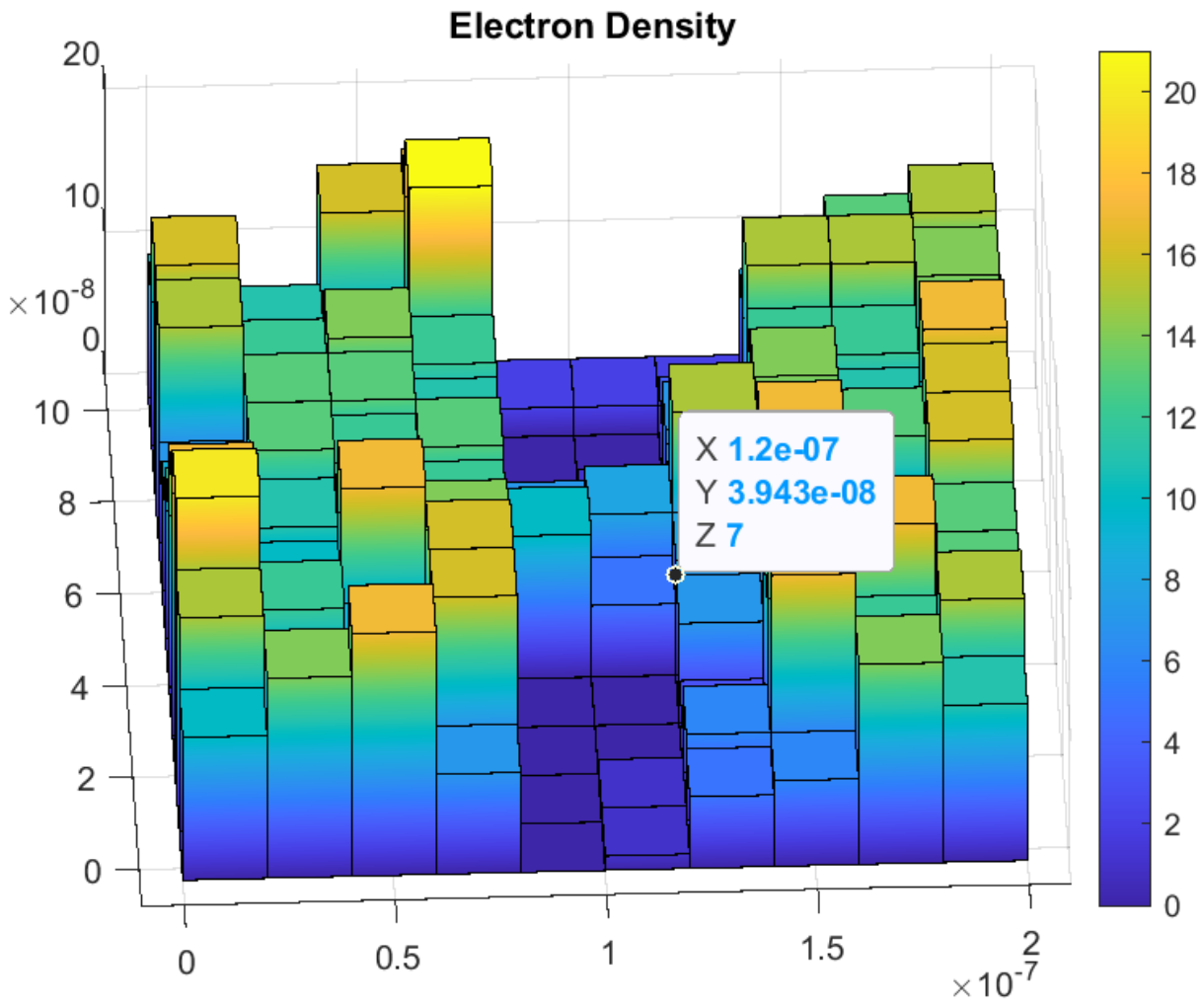
The density plot for 0.8v potential was determined using the following code.

```
figure(4)
hist3([x, y], 'CDataMode','auto','FaceColor','interp')
title(' Electron Density')
colorbar
view(2)
```

The plot generated is shown below. The same issue was present for this as was present for part 2 question c, where the velocity was too great so the electrons would occasionally pass through the boundaries before reflecting.



Another view is shown below. It can be seen that the electrons were mainly trapped on either side of the graph due to the increase in potential causing them to move more laterally than previously. Despite this, the electron density is greater than that of the density plot from part 1. This is due to the increased movement of the electrons causing them to cover more locations within the area, leading to a greater density.



Question B

The average current was determined using the following lines of code. This was implemented within the assignment 1 for loop, so the number of currents measured were equal to the number of steps.

```
e_con = 10^-15*0.0001;
vd_x = sum(vx)/1000;
J = q*vd_x*e_con; % Current density = Q * Flux
I = J/e_con;

current_current(i) = I;
```

Outside of the for loop the following line was used to determine the average current.

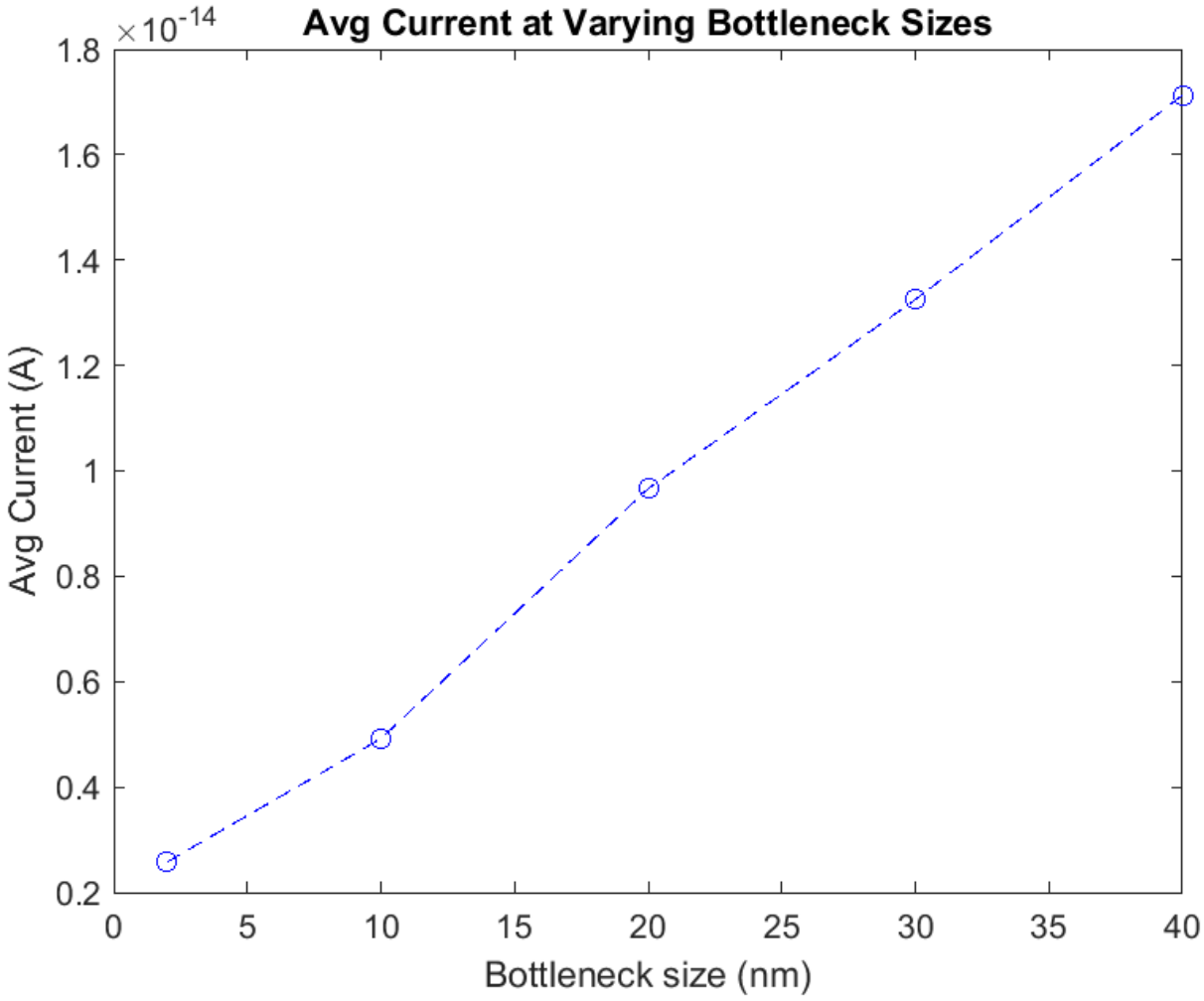
```
current_mean = mean(current_current);
```

The simulation was ran 5 different times with 5 different bottleneck sizes, and was then plotted using the following code.

```
bottleneck = 1.0e-13 * [0.1713    0.1325    0.0966    0.0492    0.0258]; % bottleneck
average current from sims

bottleneck_size = [40 30 20 10 2]; % bottleneck sizes in nano meters
```

```
figure(1)
plot(bottleneck_size, bottleneck,'b--o')
title("Avg Current at Varying Bottleneck Sizes")
xlabel("Bottleneck size (nm)")
ylabel("Avg Current (A)")
```

This resulted in the following plot. As can be seen, as the bottleneck increased so did the average current.



Question C

Several things can be done to increase the accuracy of the simulation. These are listed below.

1: Increase the number of iterations done.

2: Increase the number of electrons simulated.

3: Decrease the mesh size.

4: Improve the functionality of boundary conditions so no crossover occurs.