Table of Contents

Question 1

In this Assignment, we will add an electric field to the simulation from assignment 1, starting with a uniform electric field via a voltage (0.1V) applied over the box. The electric field on the electrons is output as EfieldX. The force on the electrons is output as ForceX. The acceleration of each electron is output as accX. the force and acceleration being negative reflects the fact that the particles move to the left(electrons have negative charge so a positive electric field results in a negative force) to find the current in the box, we use the following formula:

$$current = v_d * e * w * con_e$$

where v_d is the average drift velocity of the electrons, e is the charge of an electron, con_e is the concentration of the electrons, in this case $10^{19} electrons/m^2$, and w is the width of the box (width instead of cross sectional area since we are in 2D not 3D). As the applied voltage causes the electrons to accelerate, the current increases before leveling off at roughly 10mA due to the scattering. The trajectories of 10 the electrons is shown in figure 1. the current over time is shown in figure 2, a map of the final particle density is shown in figure 3, and the final temperature map is shown in figure 4.

```
clear
close all
hold off
%Constants
k=1.38E-23;
e_mass=9.109E-31;
e_charge=1.602e-19;
T_init=300; %Kelvin
Vth=sqrt(k*T_init/(0.26*e_mass));
%initialization
numAtoms = 10000;
plotted=10;
numsteps=1000;
Xmax=200E-9;
Ymax=100E-9;
stepn=1;
col=hsv(plotted);
Px=rand(1,numAtoms)*Xmax;
Py=rand(1,numAtoms)*Ymax;
Vx=Vth.*randn(1,numAtoms);
Vy=Vth.*randn(1,numAtoms);
figure(1)
hold on
xlim([0 Xmax])
ylim([0 Ymax])

VoltageX=0.1;
EfieldX=VoltageX/Xmax
```

```
ForceX=-EfieldX*e_charge
accX=ForceX/(0.26*e_mass)

VoltageY=0;
EfieldY=VoltageY/Ymax;
ForceY=EfieldY*e_charge;
accY=ForceY/(0.26*e_mass);

step = max(Ymax,Xmax)/(500*Vth);
tmax=numsteps*step;
t=0;
P_Scatter=1-exp(-step/0.2E-12);
current=zeros(1,numsteps);
t_arr=zeros(1,numsteps);
%main loop
while t<tmax
    PPx=Px;
    PPy=Py;
    Vx=Vx+accX*step;
    Vy=Vy+accY*step;
    Px=Px+Vx*step;
    Py=Py+Vy*step;
    for i=1:numAtoms
        %bounderies
        if Px(i)>Xmax
            Px(i)=Px(i)-Xmax;
            PPx(i)=0;
        end
        if Px(i)<0
            Px(i)=Px(i)+Xmax;
            PPx(i)=Xmax;
        end
        if Py(i)>Ymax || Py(i)<0
            Vy(i)=-Vy(i);
        end
        %random scattering
        if rand()<P_Scatter %random scattering
            Vx(i)=Vth.*randn();
            Vy(i)=Vth.*randn();
        end
    end

    Plotx=[PPx;Px];
    Ploty=[PPy;Py];
    figure(1)
    for i=1:plotted
        plot(Plotx(:,i),Ploty(:,i),'color',col(i,:))
        xlim([0 Xmax])
        ylim([0 Ymax])
        title('Figure 1: Particle Trajectories')
        hold on
    end
   t_arr(stepn)=t;
   driftV=mean(Vx);
```

```
    current(stepn)=-driftV*e_charge*10^19*Ymax;
    figure(2)
    plot(t_arr(1:stepn),current(1:stepn))
    title('Figure 2: Current over time')
    xlabel('Time (s)')
    ylabel('Current Density (A)')

    pause(0.01)
    t = t+step;
    stepn=1+stepn;

end
Z=zeros(50);
Vx_Z=zeros(50);
Vy_Z=zeros(50);
Temp_Z=zeros(50);
for x=1:50
    for y=1:50
        for i=1:numAtoms
            if
 Px(i)>=(((x-1)*Xmax)/50)&&Px(i)<(x*Xmax/50)&&Py(i)>=(((y-1)*Ymax)/50)&&Py(i)<(y*Y
                Z(y,x)=Z(y,x)+1;
                Vx_Z(y,x)=Vx_Z(y,x)+Vx(i)^2;
                Vy_Z(y,x)=Vy_Z(y,x)+Vy(i)^2;
            end

        end
        if Z(y,x)~=0
            Temp_Z(y,x)=0.26*e_mass*(Vx_Z(y,x)/Z(y,x)+Vy_Z(y,x)/
Z(y,x))/(2*k);
        end
    end
end
figure(3)
surf(linspace(0,Xmax,50),linspace(0,Ymax,50),Z)
title('Figure 3: Electron Density Map')

figure(4)
surf(linspace(0,Xmax,50),linspace(0,Ymax,50),Temp_Z)
title('Figure 4: Temperature map')


EfieldX =

   5.0000e+05


ForceX =

  -8.0100e-14


accX =
```

```
-3.3821e+17
```

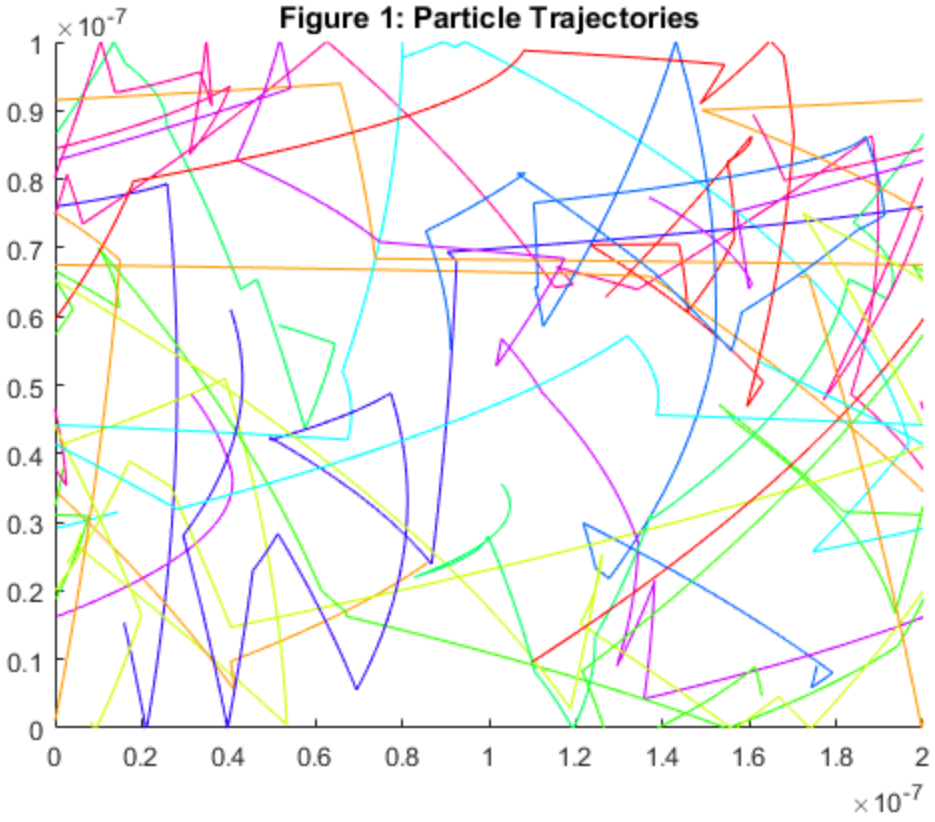


Figure 1: Particle Trajectories
×10^-7
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
×10^-7

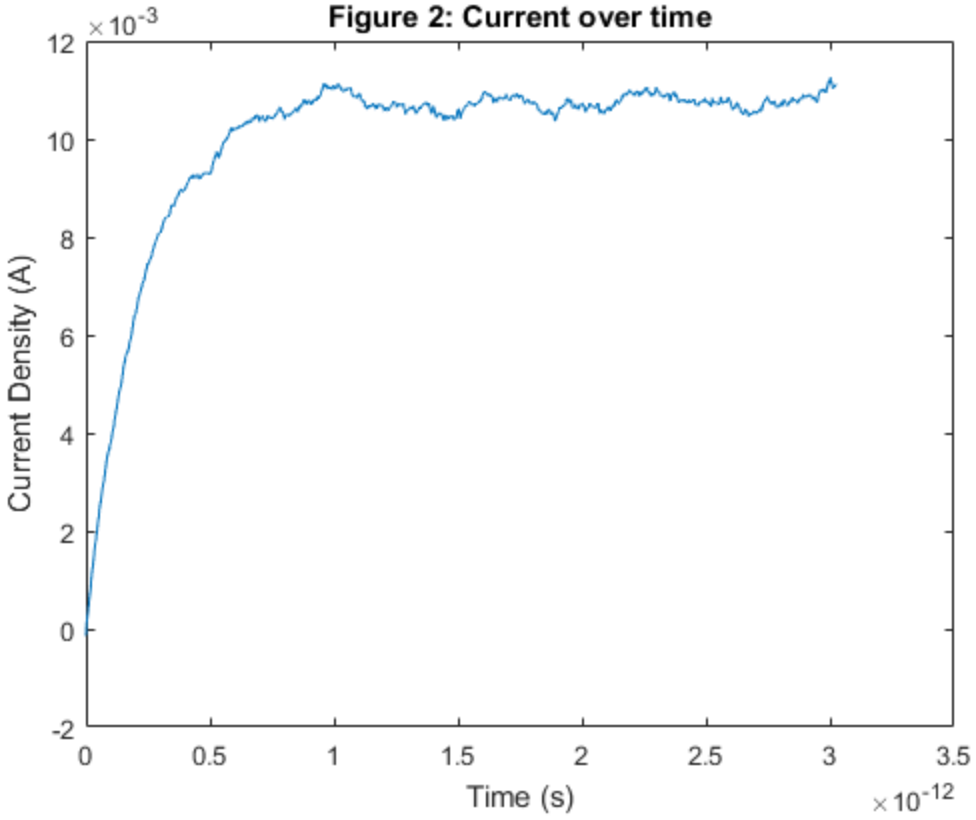


Figure 2: Current over time
×10^-3
12
10
8
6
4
2
0
-2
Current Density (A)
0
0.5
1
1.5
2
2.5
3
3.5
Time (s)
×10^-12

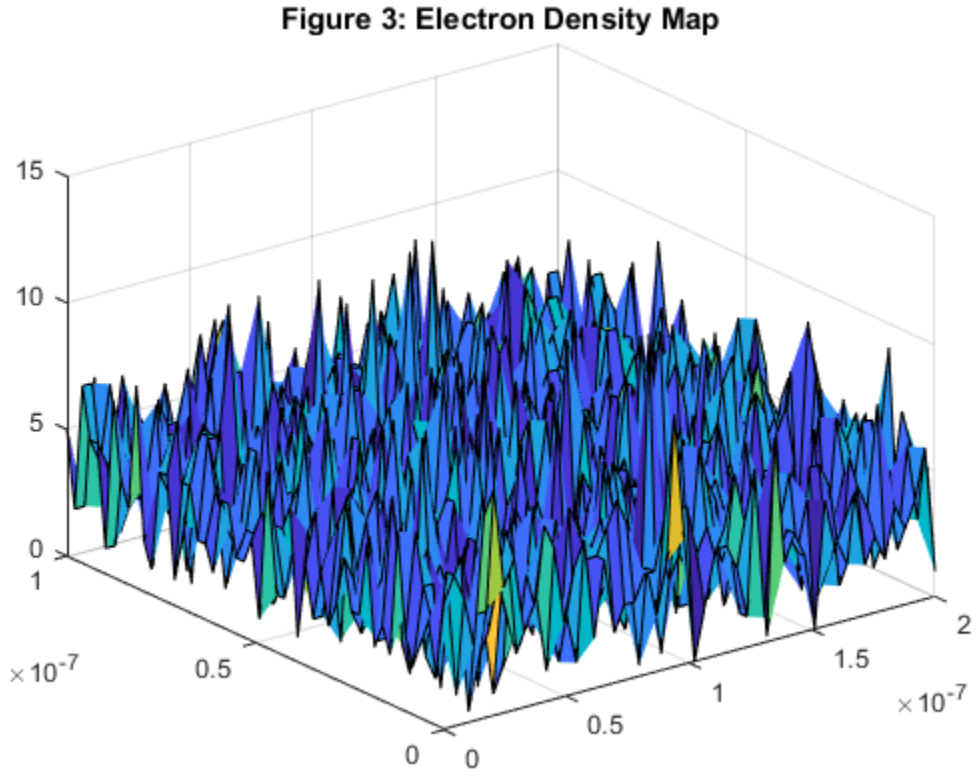
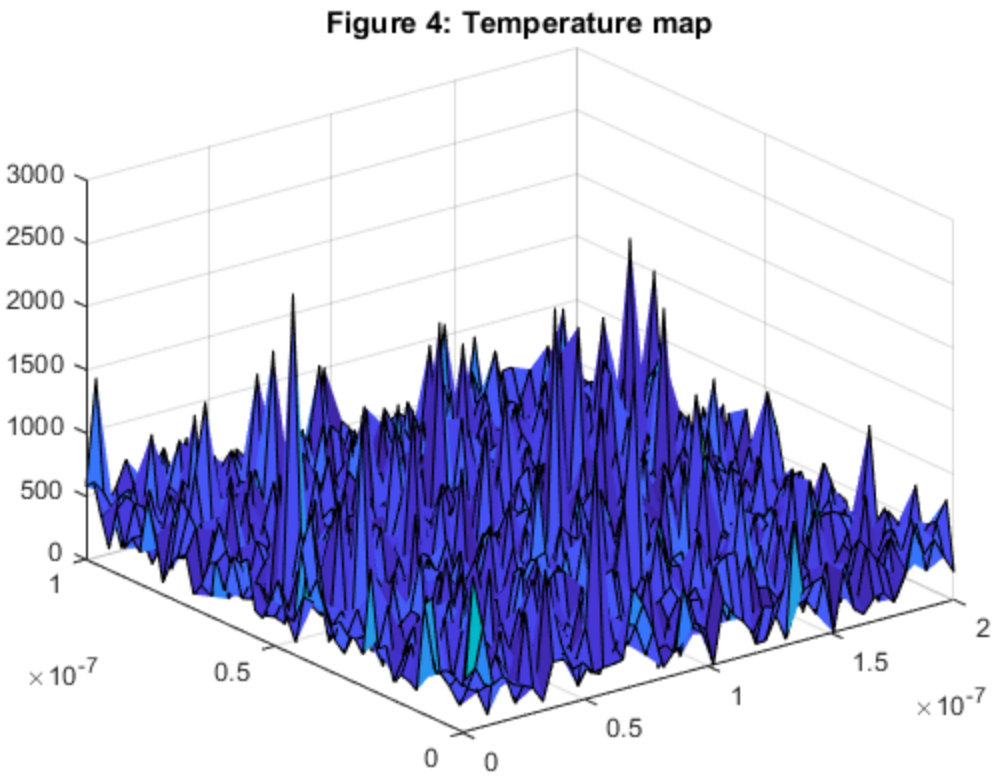


Figure 3: Electron Density Map
15
10
5
0
1
×10^-7
0.5
0
0
0.5
1
1.5
2
×10^-7



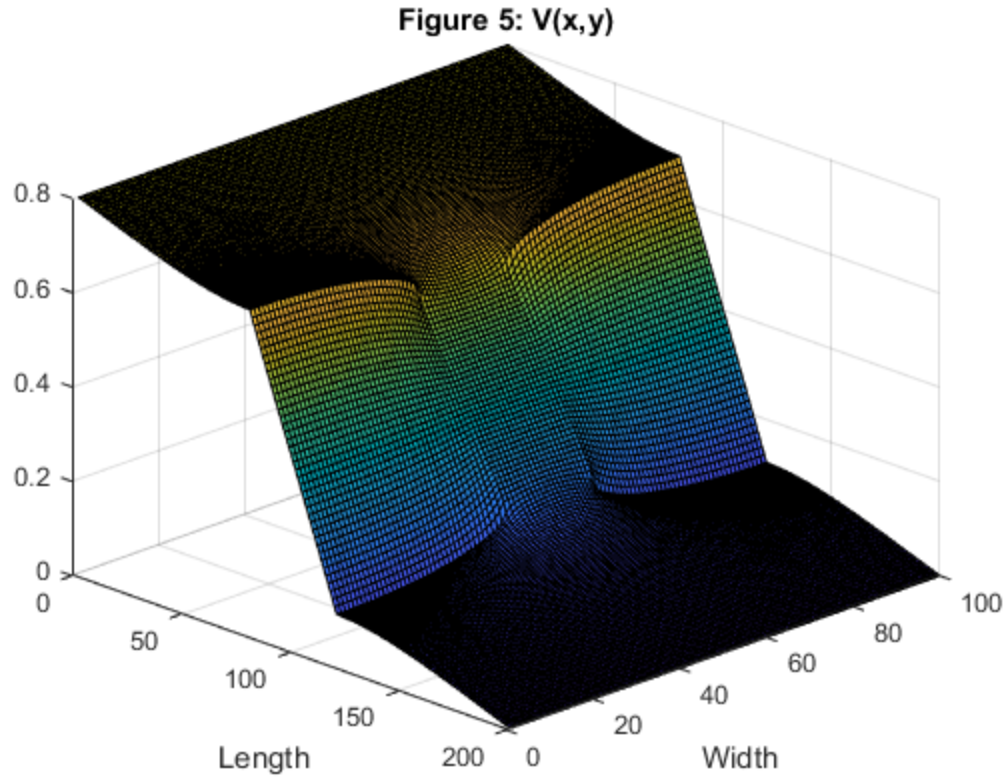
Question 2

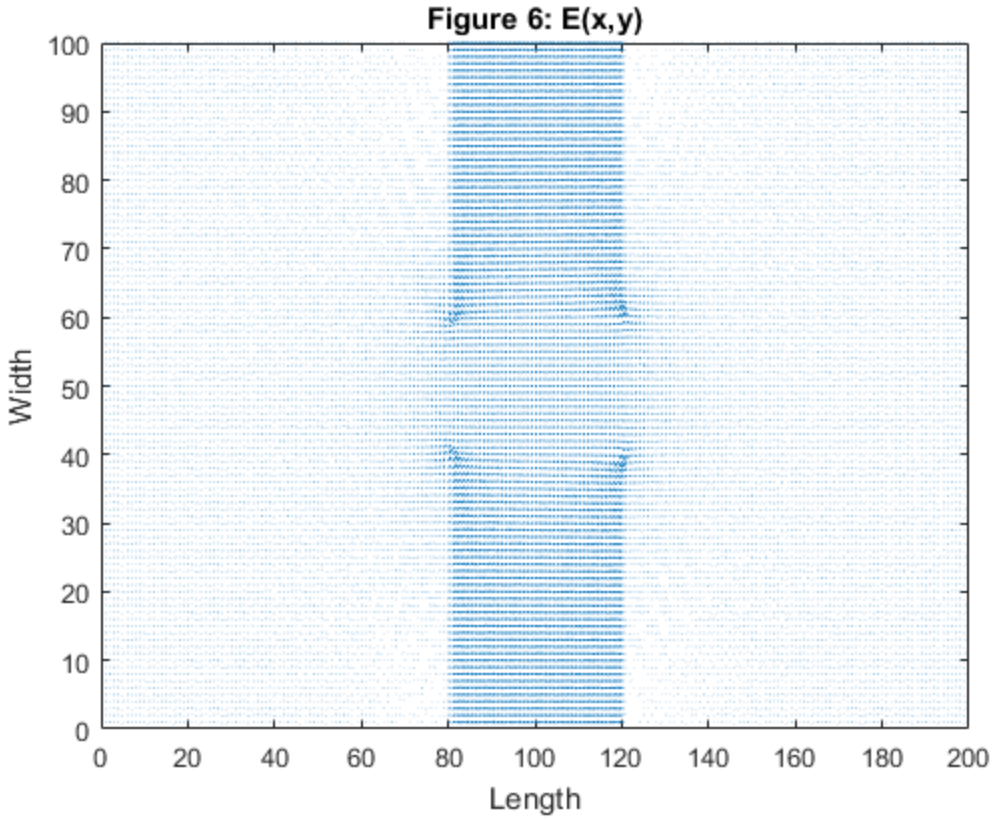
When the bottleneck is added, the electric field caused by the applied voltage will no longer be uniform. to find the electric field at each point, a finite difference method is used. The electric potential at each point is shown in figure 5 and the electric field at each point is shown in figure 6.

```
close all
clear
W=100;
L=200;
Wb=[40 60];
Lb=[80 120];
sigi=0.01;
sigo=1;
Voltage=0.8;%Volts

G=sparse(L*W);
B=zeros(1,L*W);
sigmatrix=sigo*ones(W,L);
for i=Lb(1):Lb(2)
    for j=1:Wb(1)
        sigmatrix(j,i)=sigi;
    end
    for j=Wb(2):W
        sigmatrix(j,i)=sigi;
    end
```

```
end
for i =1:L
    for j=1:W
        n=j+(i-1)*W;
        if i==1
            G(n,:)=0;
            G(n,n)=1;
            B(n)=Voltage;
        elseif i==L
            G(n,:)=0;
            G(n,n)=1;
            B(n)=0;
        elseif j==1
            Rup=(sigmatrix(j,i)+sigmatrix(j+1,i))/2;
            Rleft=(sigmatrix(j,i)+sigmatrix(j,i-1))/2;
            Rright=(sigmatrix(j,i)+sigmatrix(j,i+1))/2;
            G(n,n)=-(Rup+Rleft+Rright);
            G(n,n+1)=Rup;
            G(n,n+W)=Rright;
            G(n,n-W)=Rleft;
        elseif j==W
            Rdown=(sigmatrix(j,i)+sigmatrix(j-1,i))/2;
            Rleft=(sigmatrix(j,i)+sigmatrix(j,i-1))/2;
            Rright=(sigmatrix(j,i)+sigmatrix(j,i+1))/2;
            G(n,n)=-(Rdown+Rleft+Rright);
            G(n,n-1)=Rdown;
            G(n,n+W)=Rright;
            G(n,n-W)=Rleft;
        else
            Rup=(sigmatrix(j,i)+sigmatrix(j+1,i))/2;
            Rdown=(sigmatrix(j,i)+sigmatrix(j-1,i))/2;
            Rleft=(sigmatrix(j,i)+sigmatrix(j,i-1))/2;
            Rright=(sigmatrix(j,i)+sigmatrix(j,i+1))/2;
            G(n,n)=-(Rup+Rdown+Rleft+Rright);
            G(n,n+1)=Rup;
            G(n,n-1)=Rdown;
            G(n,n+W)=Rright;
            G(n,n-W)=Rleft;
        end
    end
end
V=G\B';
Vmatrix=zeros(W,L);
for i =1:L
    for j=1:W
        n=j+(i-1)*W;
        Vmatrix(j,i)=V(n);
    end
end


[Ex,Ey]=gradient(Vmatrix);
Ex=-Ex;
Ey=-Ey;
```

```
figure (5)
surf(Vmatrix)
title('Figure 5: V(x,y)')
xlabel('Length')
ylabel('Width')
view(45,30)

figure (6)
quiver(Ex,Ey)
title('Figure 6: E(x,y)')
xlabel('Length')
ylabel('Width')
xlim([0 L])
ylim([0 W])
```





Question 3

Now that the electric field is known, the trajectories of the particles can be simulated as in part 1, but with the bottlneck. the particle trajectories are shown in figure 7. The final density map of the particles is shown in figure 8. From the density map, it is clear that the bottleneck is causing a buildup of electrons on the right side of it, as electrons are blocked from reaching the other side. in addition, we see a lack of electrons on the left side, especially in the top and bottom corners, as particles are accelerated to the left, and fewer are able to reach those corners.

The simulaion could be made more accurate in a number of ways. One such method could be to increase the number of electrons simulated, or the length of the simulation. The simulation could also be made more realistic, by removing the periodic boundary condition, and replacing it with an injection of particles as would be the case for a device in a circuit with other components.

```
close all
hold off
%Constants
k=1.38E-23;
e_mass=9.109E-31;
e_charge=1.602e-19;
T_init=300; %Kelvin
Vth=sqrt(k*T_init/(0.26*e_mass));
plotted=10;
col=hsv(plotted);
%initialization
```

```
rethermalize=0; %sets whether or not particles are rethermalized when
 they bounce off a box
numAtoms = 10000;
numsteps=1000;
Xmax=200E-9;
Ymax=100E-9;
stepn=1;
Px=rand(1,numAtoms)*Xmax;
Py=rand(1,numAtoms)*Ymax;

Fx=-e_charge*Ex;
Fy=-e_charge*Ey;
accx=Fx/(0.26*e_mass*10^-9);
accy=Fy/(0.26*e_mass*10^-9);

for i=1:numAtoms %makes sure particles don't start in the box
    while (Py(i)<=40E-9&&Px(i)>=80E-9&&Px(i)<=120E-9)||
(Py(i)>=60E-9&&Px(i)>=80E-9&&Px(i)<=120E-9)
        Px(i)=rand()*Xmax;
        Py(i)=rand()*Ymax;
    end
end
Vx=Vth.*randn(1,numAtoms);
Vy=Vth.*randn(1,numAtoms);
figure(7)
hold on
xlim([0 Xmax])
ylim([0 Ymax])
rectangle('Position',[80E-9,0,40E-9,40E-9])
rectangle('Position',[80E-9,60E-9,40E-9,40E-9])

step = max(Ymax,Xmax)/(500*Vth);
Tmax=numsteps*step;
t=0;
P_Scatter=1-exp(-step/0.2E-12);

%main loop
while t<Tmax
    PPx=Px;
    PPy=Py;
    for i=1:numAtoms
    Vx(i)=Vx(i)+accx(ceil(Py(i)/10^-9),ceil(Px(i)/10^-9))*step;
    Vy(i)=Vy(i)+accy(ceil(Py(i)/10^-9),ceil(Px(i)/10^-9))*step;
    end
    Px=Px+Vx*step;
    Py=Py+Vy*step;
    for i=1:numAtoms
        %bounderies
        if Px(i)>Xmax
            Px(i)=Px(i)-Xmax;
            PPx(i)=0;
        end
        if Px(i)<0
            Px(i)=Px(i)+Xmax;
```

```
            PPx(i)=Xmax;
        end
        if Py(i)>Ymax
            Vy(i)=-Vy(i);
            Py(i)=Py(i)-2*abs(Py(i)-Ymax);
        end
        if Py(i)<0
            Vy(i)=-Vy(i);
            Py(i)=-Py(i);
        end
        %random scattering
        if rand()<P_Scatter %random scattering
            Vx(i)=Vth.*randn();
            Vy(i)=Vth.*randn();
        end
        %boxes
            if Py(i)<40E-9 && Px(i)>=80E-9 && PPx(i)<=80E-9
                if rethermalize
                    Vy(i)=Vth*randn();
                    Vx(i)=-abs(Vth*randn());
                else
                Vx(i)=-Vx(i);
                end
                Px(i)=Px(i)-2*abs(Px(i)-80E-9);
            end
            if Py(i)<40E-9&&Px(i)<=120E-9&&PPx(i)>=120E-9
                if rethermalize
                    Vy(i)=Vth*randn();
                    Vx(i)=abs(Vth*randn());
                else
                Vx(i)=-Vx(i);
                end
                Px(i)=Px(i)+2*abs(Px(i)-120E-9);
            end
            if
Py(i)<=40E-9&&PPy(i)>=40E-9&&Px(i)>=80E-9&&Px(i)<=120E-9
                if rethermalize
                    Vx(i)=Vth*randn();
                    Vy(i)=abs(Vth*randn());
                else
                Vy(i)=-Vy(i);
                end
                Py(i)=Py(i)+2*abs(Py(i)-40E-9);
            end
            if Py(i)>60E-9&&Px(i)>=80E-9&&PPx(i)<=80E-9
                if rethermalize
                    Vy(i)=Vth*randn();
                    Vx(i)=-abs(Vth*randn());
                else
                Vx(i)=-Vx(i);
                end
                Px(i)=Px(i)-2*abs(Px(i)-80E-9);
            end
            if Py(i)>=60E-9&&Px(i)<=120E-9&&PPx(i)>=120E-9
```

```
                if rethermalize
                    Vy(i)=Vth*randn();
                    Vx(i)=abs(Vth*randn());
                else
                Vx(i)=-Vx(i);
                end
                Px(i)=Px(i)+2*abs(Px(i)-120E-9);
            end
            if
 Py(i)>=60E-9&&PPy(i)<=60E-9&&Px(i)>=80E-9&&Px(i)<=120E-9
                if rethermalize
                    Vx(i)=Vth*randn();
                    Vy(i)=-abs(Vth*randn());
                else
                Vy(i)=-Vy(i);
                end
                Py(i)=Py(i)-2*abs(Py(i)-60E-9);
            end
    end

    Plotx=[PPx;Px];
    Ploty=[PPy;Py];
    figure(7)
    for i=1:plotted
        plot(Plotx(:,i),Ploty(:,i),'color',col(i,:))
        xlim([0 Xmax])
        ylim([0 Ymax])
        title('Figure 7: Particle Trajectories')
        hold on
    end
    pause(0.01)
    t = t+step;
    stepn=1+stepn;

end
%end plots
Z=zeros(50);
for x=1:50
    for y=1:50
        for i=1:numAtoms
            if
 Px(i)>=(((x-1)*Xmax)/50)&&Px(i)<(x*Xmax/50)&&Py(i)>=(((y-1)*Ymax)/50)&&Py(i)<(y*Y
                Z(y,x)=Z(y,x)+1;
            end
        end
    end
end
figure(8)
surf(linspace(0,Xmax,50),linspace(0,Ymax,50),Z)
title('Figure 8: Electron Density Map')
view(25,60)
```

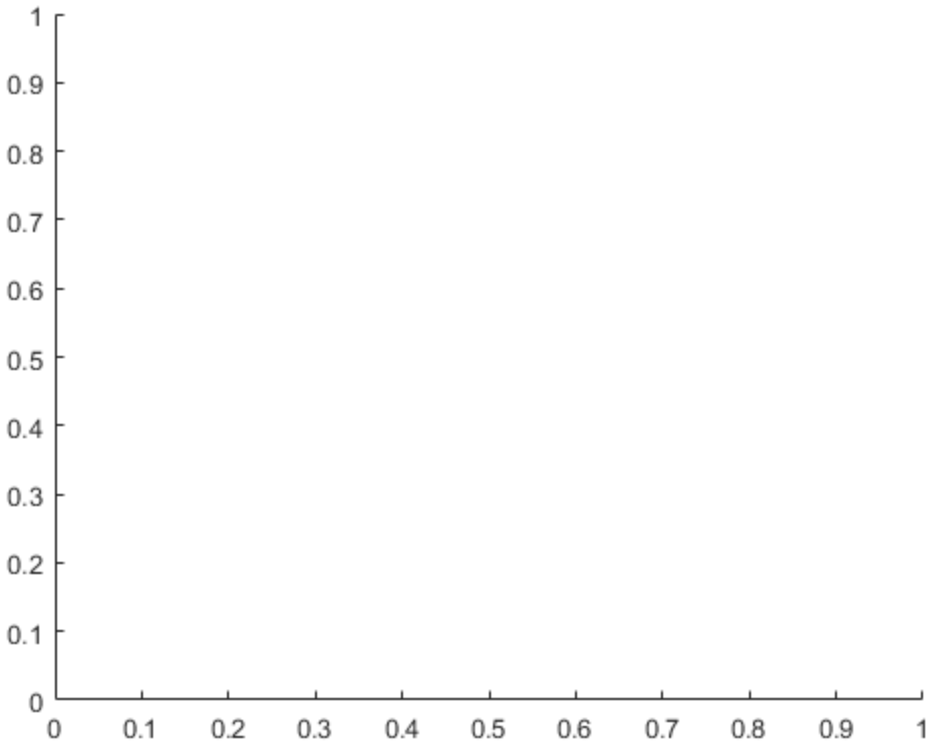
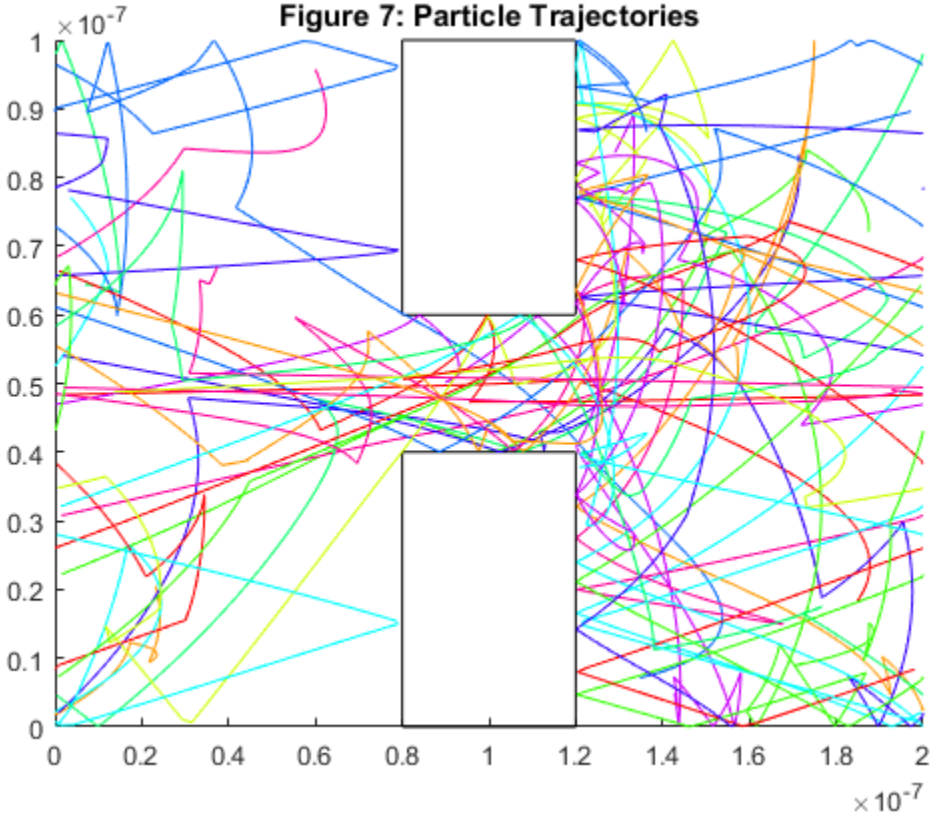
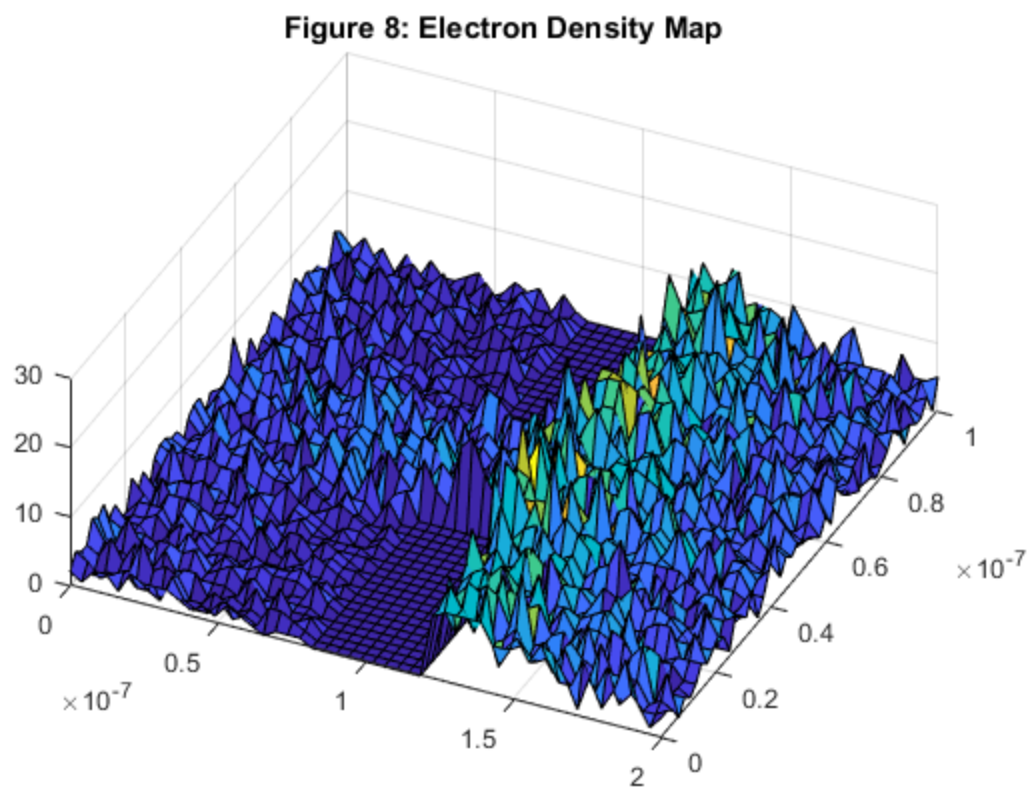


Figure 7: Particle Trajectories





Published with MATLAB® R2017a