# Table of Contents

Question 1: Electron Modelling ..... 1
Question 2: Collisions with Mean Free Path (MFP) ..... 4
Question 3: Enhancements ..... 8

# Question 1: Electron Modelling

The termal velocity $v_{th}$ is

$$v_{th} = \sqrt{k_b T / 0.26 m_0}$$

where T is the temperature in kelvin, $m_0$ is the rest mass of

electron and $k_b$ is boltzmans constant. For a temperature of 300K

$$v_{th} = 1.3221 \times 10^5 \ m/s$$

If the mean time between collisions is $\tau_{mn} = 0.2ps$ then the mean free path is

$$MFP = v_{th} \times \tau_{mn} = 26.443nm$$

This script models the motion of 10000 electrons in a 200nm by 100nm box. The top and bottom bounderies of the box reflect electrons, while the left and right bounderies are periodic. Each electron is started at a random location in the box, and given an initial velocity equal to $v_{th}$ in a random direction. The trajectories of ten of these electrons are plotted in figure 1. The temperature of the box over time is displayed in figure 2.

```
close all
hold off
%Constants
k=1.38E-23;
e_mass=9.109E-31;
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;
Px=rand(1,numAtoms)*Xmax;
Py=rand(1,numAtoms)*Ymax;
V_angle=2*pi*rand(1,numAtoms);
Vx=Vth*cos(V_angle);
Vy=Vth*sin(V_angle);
figure(1)
xlim([0 Xmax])
ylim([0 Ymax])
step = max(Ymax,Xmax)/(500*Vth);
Tmax=numsteps*step;
t=0;
```

```
Temp=zeros(1,numsteps);
t_arr=zeros(1,numsteps);
col=hsv(plotted);
%main loop
while t<Tmax
    PPx=Px;
    PPy=Py;
    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
    end

    Plotx=[PPx;Px];
    Ploty=[PPy;Py];
    figure(1)
    for i=1:plotted
        plot(Plotx(:,i),Ploty(:,i),'color',col(i,:))
        title('Figure 1: Particle Trajectories')
        xlim([0 Xmax])
        ylim([0 Ymax])
        hold on
    end
    Temp(stepn)=0.26*e_mass*(mean(Vx.^2)+mean(Vy.^2))/k;
    t_arr(stepn)=t;
    figure(2)
    plot(t_arr(1:stepn),Temp(1:stepn))
    title('Figure 2: Temperature Plot')
    xlabel('Time (s)')
    ylabel('Temperature (Kelvin)')
    pause(0.01)


    t = t+step;
    stepn=1+stepn;

end
```

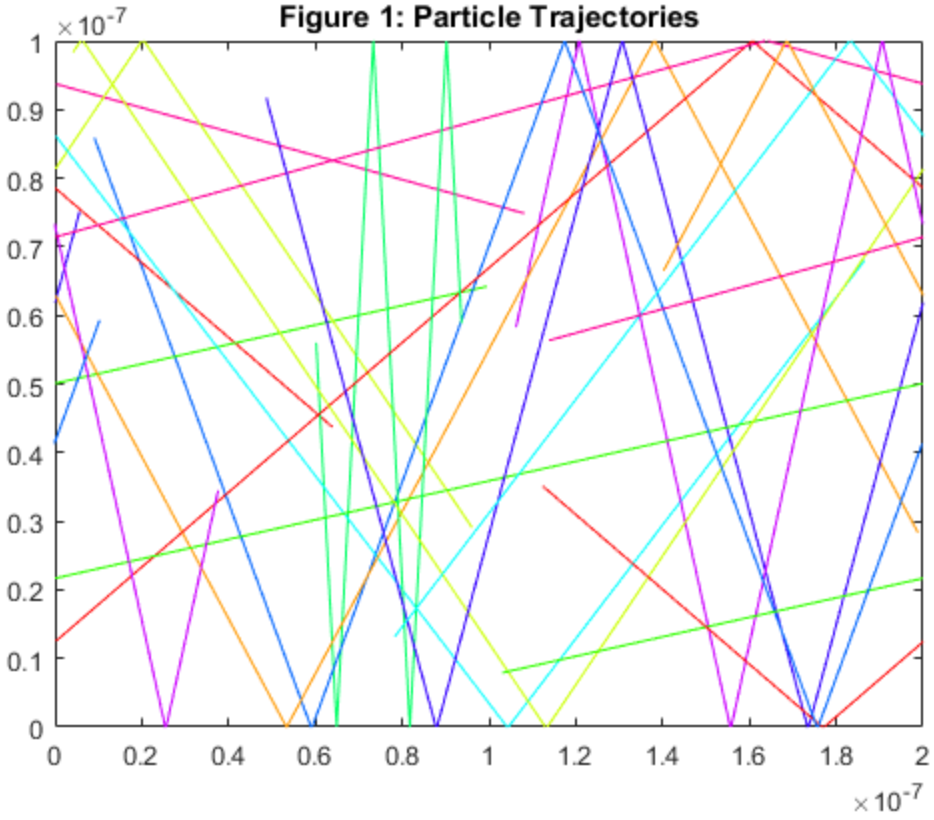


Figure 1: Particle Trajectories
×10^-7
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
×10^-7

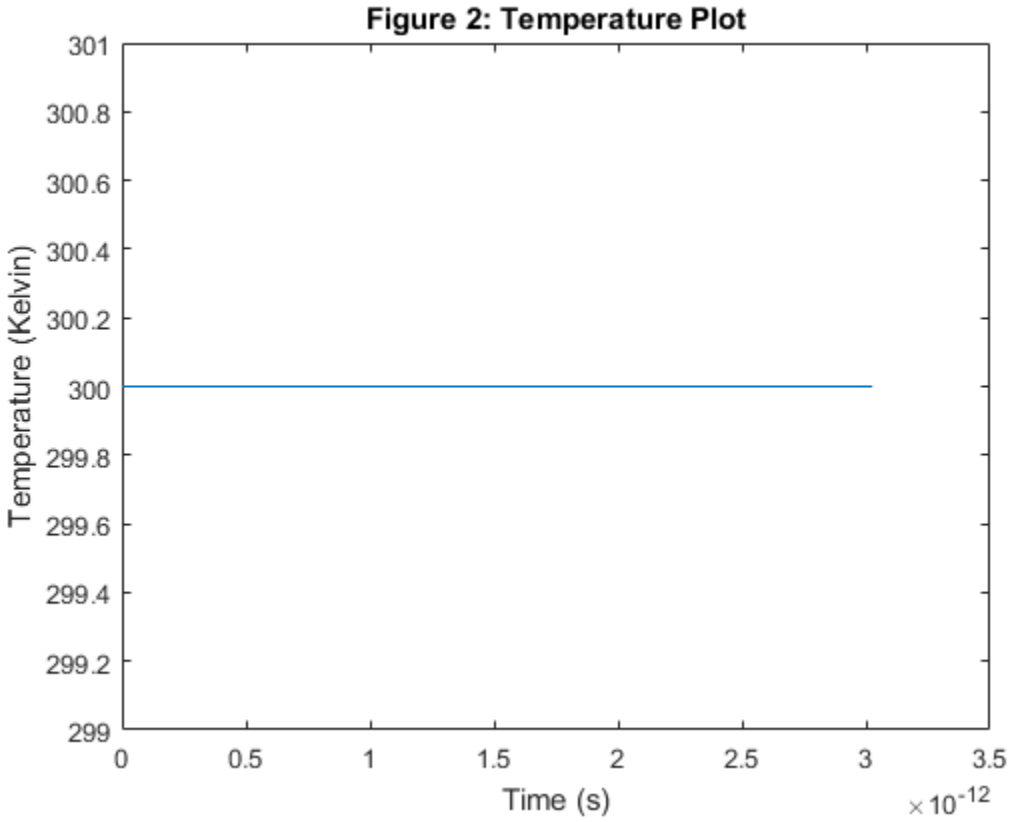


Figure 2: Temperature Plot
Temperature (Kelvin)
301
300.8
300.6
300.4
300.2
300
299.8
299.6
299.4
299.2
299
0
0.5
1
1.5
2
2.5
3
3.5
Time (s)
×10^-12

# Question 2: Collisions with Mean Free Path (MFP)

Now we add the effect of electrons scattering off of each other to the simulation. we do this using a stocastic method, by calculating the probability that an electron will scatter in each time step based on the average time between collsions. we apply this probability to each electron seperately, and it is expressed as

$$P_{scat} = 1 - e^{-dt/\tau_{mn}}$$

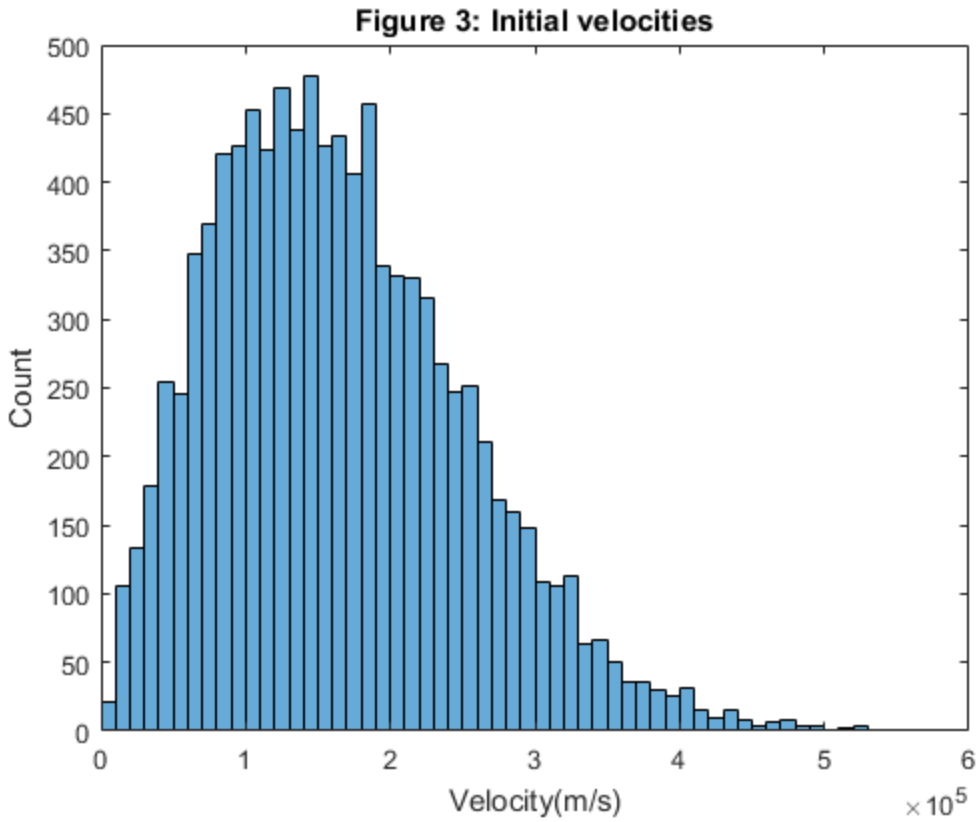
due to this scattering, the average temperature will vary over time, although it will remain close to the 300K. The average temperature over time is displayed in figure 5. We also assigned each particle a random velocity using a Maxwell-Boltzmann distribution such that the average of all the speeds is $v_{th}$. A histogram of the initial velocities is displayed in figure 3. The actual mean free path and mean time between collisions is calculated once the simulation is complete, and is outputted into the console. The trajectories of ten of the particles are shown in figure 4.

```
close all
hold off
%Constants
k=1.38E-23;
e_mass=9.109E-31;
T_init=300; %Kelvin
Vth=sqrt(k*T_init/(0.26*e_mass));
col=hsv(plotted);
%initialization
numAtoms = 10000;
plotted=10;
numsteps=1000;
Xmax=200E-9;
Ymax=100E-9;
stepn=1;
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])

step = max(Ymax,Xmax)/(500*Vth);
Tmax=numsteps*step;
t=0;
P_Scatter=1-exp(-step/0.2E-12);
Temp=zeros(1,numsteps);
t_arr=zeros(1,numsteps);
t_col=zeros(1,numAtoms);
Tau=0;
Path=0;
%velocity histogram
figure(3)
histogram(sqrt(Vx.^2+Vy.^2))
```

```
title('Figure 3: Initial velocities')
xlabel('Velocity(m/s)')
ylabel('Count')
%main loop
while t<Tmax
    PPx=Px;
    PPy=Py;
    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
        t_col(i)=step+t_col(i);
        if rand()<P_Scatter %random scattering
            Vx(i)=Vth.*randn();
            Vy(i)=Vth.*randn();
            Tau=[Tau,t_col(i)];
            Path=[Path,t_col(i)*sqrt((Vx(i)^2+Vy(i)^2))];
            t_col(i)=0;
        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 4: Particle Trajectories')
        hold on
    end
    Temp(stepn)=0.26*e_mass*(mean(Vx.^2)+mean(Vy.^2))/(2*k);
    t_arr(stepn)=t;
    figure(2)
    plot(t_arr(1:stepn),Temp(1:stepn))
    title('Figure 5: Average Temperature over Time')
    xlabel('Time (s)')
    ylabel('Temperature (Kelvin)')
    pause(0.01)


    t = t+step;
```

```
    stepn=1+stepn;

end
%end statistics
Tau_mn=mean(Tau(2:length(Tau)))
MFP=mean(Path(2:length(Path)))
```

```
Tau_mn =

   1.8776e-13


MFP =

   3.1010e-08
```



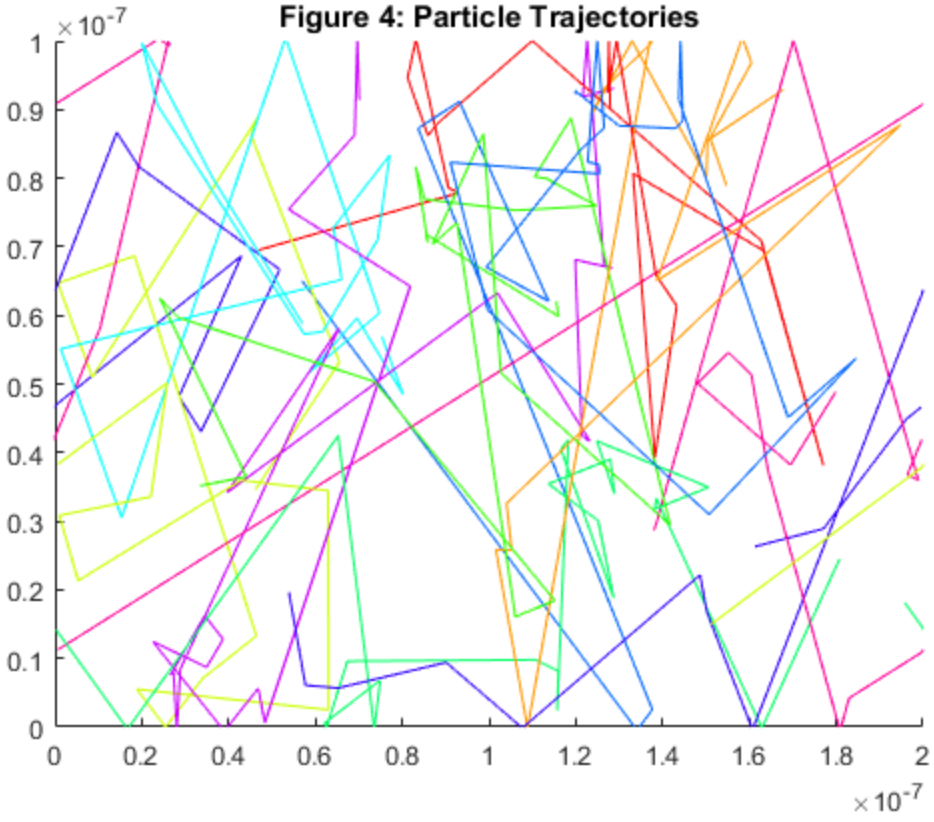


Figure 4: Particle Trajectories
×10^-7
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
×10^-7

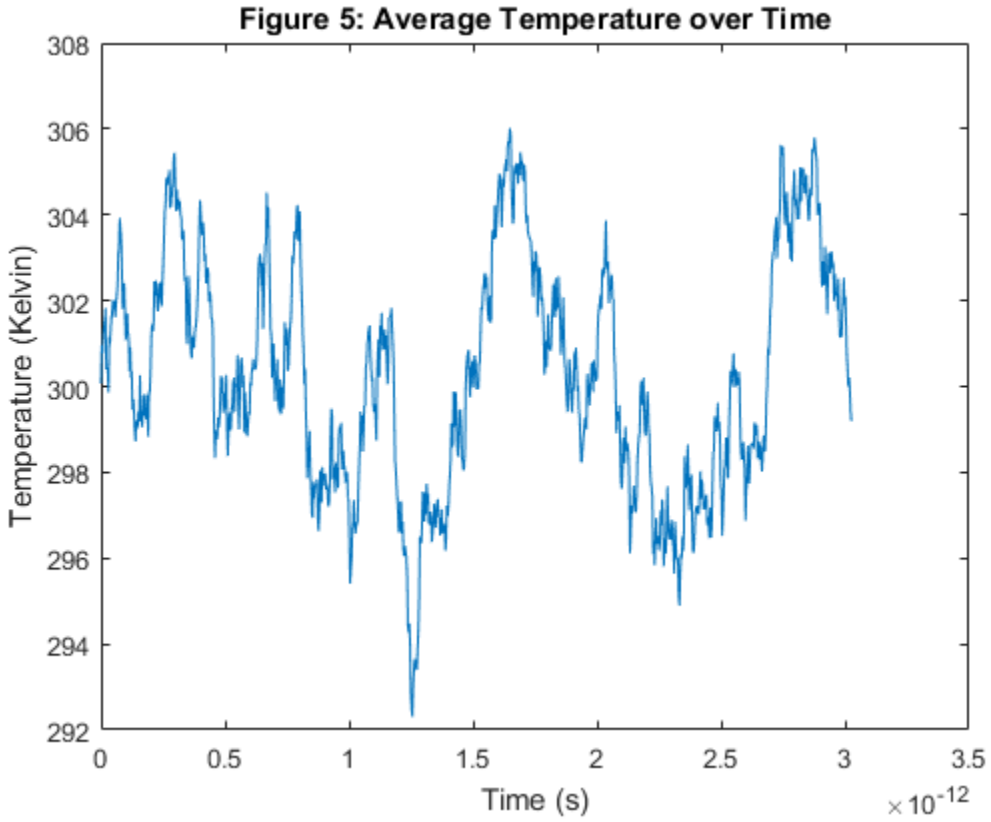


Figure 5: Average Temperature over Time
Temperature (Kelvin)
308
306
304
302
300
298
296
294
292
0
0.5
1
1.5
2
2.5
3
3.5
Time (s)
×10^-12

# Question 3: Enhancements

in this section, regions where electrons are not allowed are added to create a bottleneck effect. The boundaries of these regions reflect the electrons, either specularly or diffusely(i.e. re-thermalizing the electron) depending on a setting at the beggining of the code. Some code was also added to ensure that electrons do not begin the simulation inside the box. The trajectories of ten of the electrons are shown in figure 6. At the end of the simulation, an electron density map, shown in figure 7, and temperature map, shown in figure 8, are generated.

```
close all
hold off
%Constants
k=1.38E-23;
e_mass=9.109E-31;
T_init=300; %Kelvin
Vth=sqrt(k*T_init/(0.26*e_mass));
col=hsv(plotted);
%initialization
rethermalize=1; %sets whether or not particles are rethermalized when
 they bounce off a box
numAtoms = 10000;
plotted=10;
numsteps=1000;
Xmax=200E-9;
Ymax=100E-9;
stepn=1;
Px=rand(1,numAtoms)*Xmax;
Py=rand(1,numAtoms)*Ymax;

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(1)
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;
    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
        t_col(i)=step+t_col(i);
        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(1)
    for i=1:plotted
        plot(Plotx(:,i),Ploty(:,i),'color',col(i,:))
        xlim([0 Xmax])
        ylim([0 Ymax])
        title('Figure 6: Particle Trajectories')
        hold on
    end
    pause(0.01)


    t = t+step;
    stepn=1+stepn;

end
%end plots
Z=zeros(50);
V_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;
```

```
                V_Z(y,x)=V_Z(y,x)+sqrt(Vx(i)^2+Vy(i)^2);
            end

        end
        if V_Z(y,x)~=0
            Temp_Z(y,x)=0.26*e_mass*(V_Z(y,x)/Z(y,x))/(2*k);
        end
    end
end
figure(5)
surf(linspace(0,Xmax,50),linspace(0,Ymax,50),Z)
title('Figure 7: Electron Density Map')

figure(6)
surf(linspace(0,Xmax,50),linspace(0,Ymax,50),Temp_Z)
title('Figure 8: Temperature map')
```

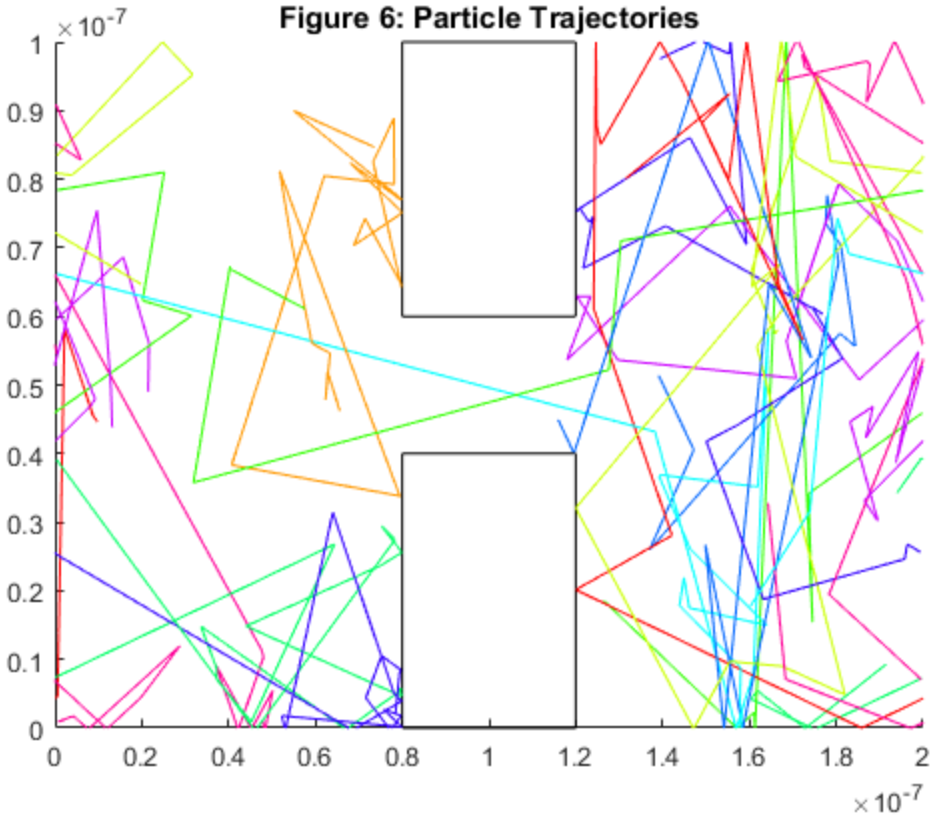


**Figure 7: Electron Density Map**

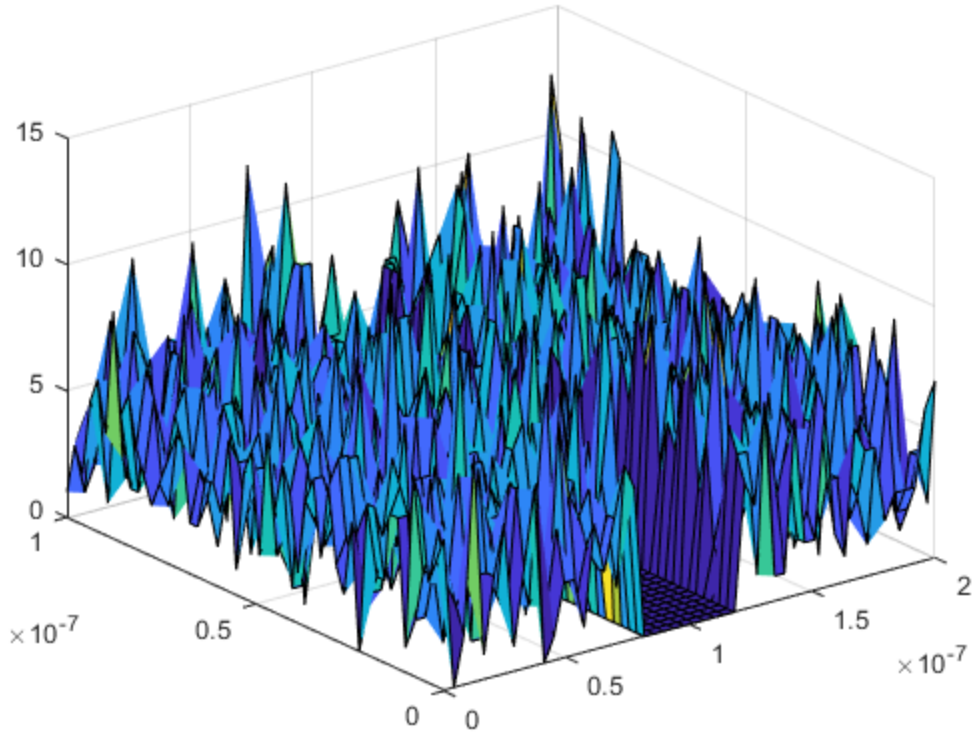
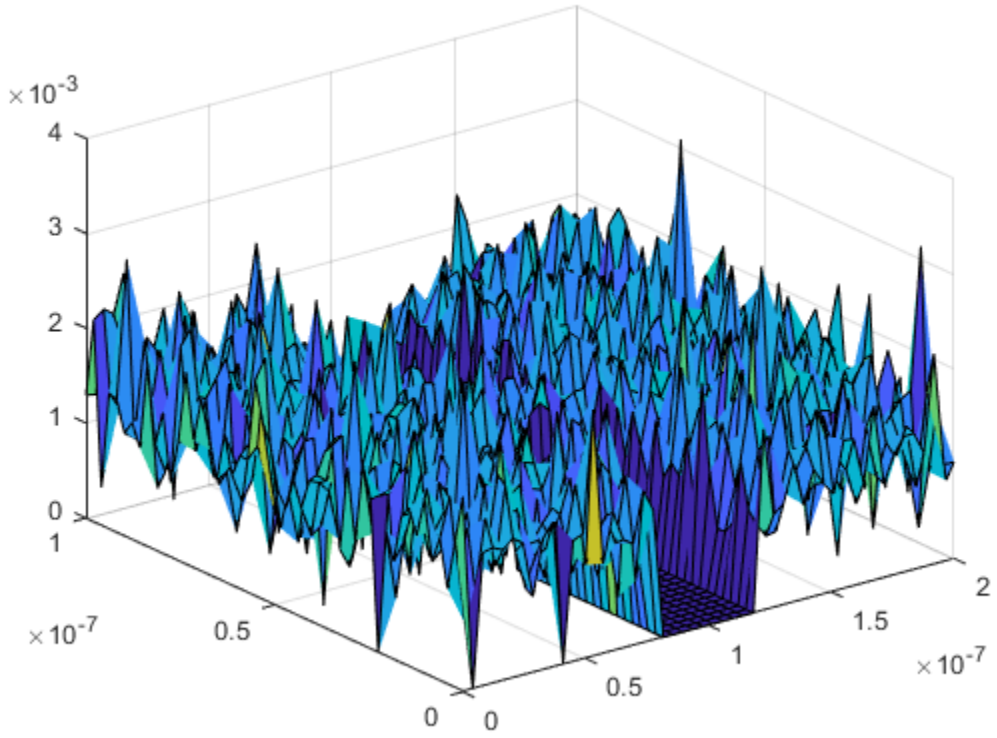



**Figure 8: Temperature map**



*Published with MATLAB® R2017a*