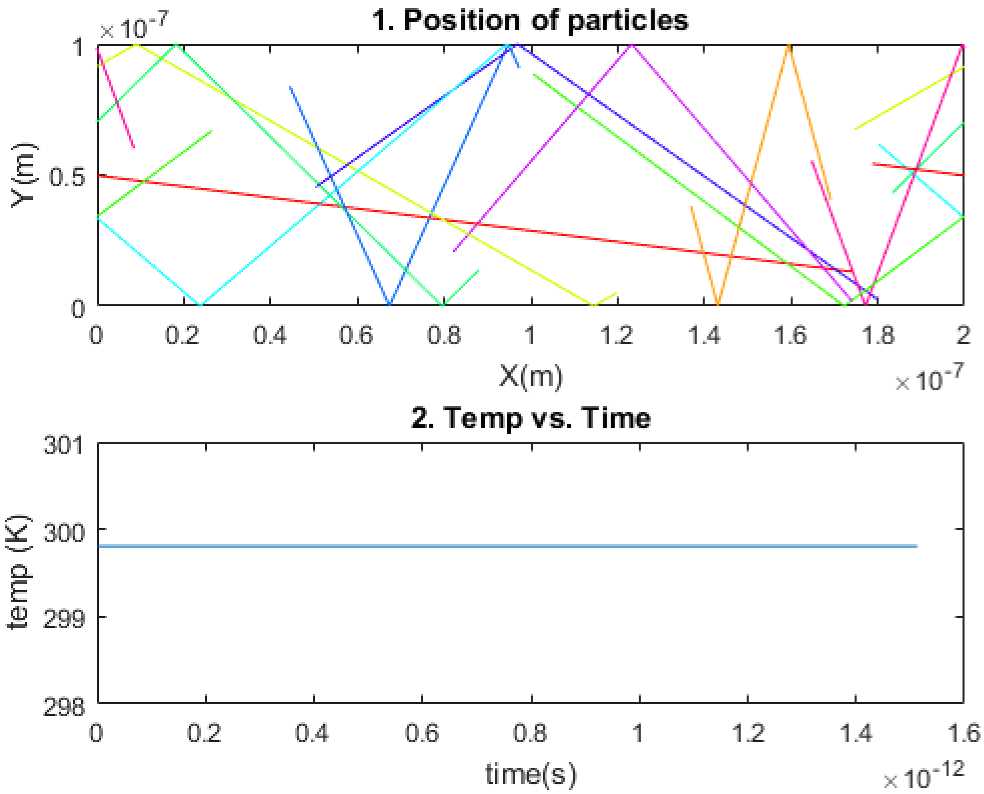
Table of Contents

Question 1

```
%In the first comments of this code, the thermal velocity and mean
 free
%path are found using the equations at the end of question 1.  Then,
 the
%position of particles is randomly assigned with a constant speed in a
%random direction.  This can be seen in figure 1.  The trajectories
 are plotted, and the temperature at
%each time step is plotted. This can be seen in figure 2.


%Part A)
%Thermal Velocity = 132.2e3 m/s
%Part B)
%mean free path =2.644e-8 meters

%variables you can edit
num_e = 10;
x_dim = 200*10^-9;
y_dim = 100e-9;


col=hsv(num_e); %create a colour array for each line in the movie

Temp_arr=300;
vth=132.2e3;

close all
hold off

%generate an initial array of positions and velocities
[x_arr,y_arr, vx_arr,vy_arr] = gen_e(num_e,x_dim,y_dim,1);

%set time constrainsts
t=0 ;
t_step = max(x_dim,y_dim)/(1000*vth);
Tstop=1000*t_step;
time_arr=zeros(1,1001);
for i=1:length(time_arr)
    time_arr(i)=(i-1)*t_step;
end
```

```
%loop over the timeframe
while t< Tstop
    %calculate temp
    Temp_arr = [Temp_arr,1/
(1.3806e-23)*9.109e-31*.26*(mean(vx_arr.^2+vy_arr.^2))];

    %add the time step to the position
    xp_arr=x_arr;
    xg_arr=x_arr;
    yp_arr=y_arr;
    yg_arr=y_arr;
    x_arr=x_arr+vx_arr*t_step;
    y_arr=y_arr+vy_arr*t_step;

    %check boundaries
    for q=1:num_e
       if x_arr(q)<0
           x_arr(q)=x_arr(q)+x_dim;
           xg_arr(q)=x_dim;
       end
       if x_arr(q) > x_dim
           x_arr(q)=x_arr(q)-x_dim;
           xg_arr(q)=0;
       end
       if y_arr(q)>y_dim
           vy_arr(q)=-vy_arr(q);
           y_arr(q)=2*y_dim-y_arr(q);
       end
       if y_arr(q)<0
           vy_arr(q)=-vy_arr(q);
           y_arr(q)=abs(y_arr(q));
       end
    end


    %plot the particle trajectories
    subplot(2,1,1)
    xlabel('X(m)')
    ylabel('Y(m)')
    title('1. Position of particles')
    xlim([0 x_dim])
    ylim([0 y_dim])
    pause(.01)

    for q=1:num_e
        plot([xg_arr(q);x_arr(q)],
[yg_arr(q);y_arr(q)],'color',col(q,:))
        hold on
    end

    t=t+t_step;

    %plot temperature vs. time
```

```
    subplot(2,1,2)
    plot(time_arr(2:length(Temp_arr)),Temp_arr(2:length(Temp_arr)))
    xlabel('time(s)')
    ylabel('temp (K)')
    title('2. Temp vs. Time')
end
```



Equations:

$$Thermal\ Velocity = \sqrt{v^2 * k_b/(0.26 * m_e)}$$

$$Temperature(K) = (mean(V^2) * 0.26 * m_e)/k_b$$

$$Mean\ Free\ Path = Thermal\ Velocity * Mean\ time\ between\ collisions$$

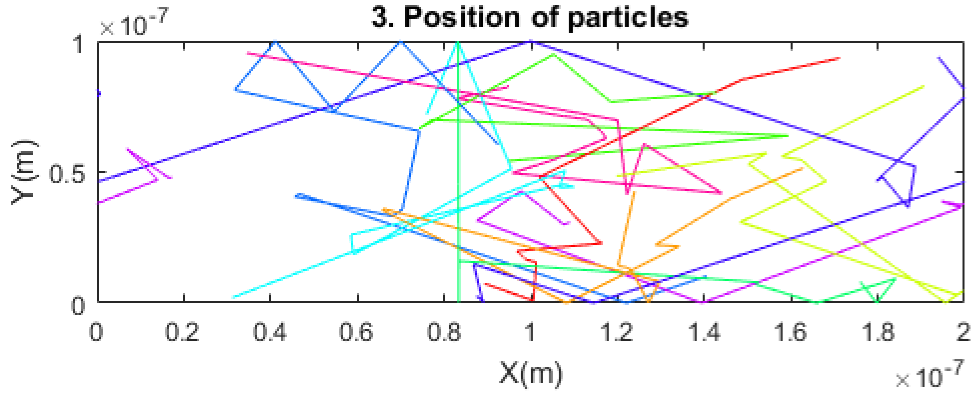
Question 2

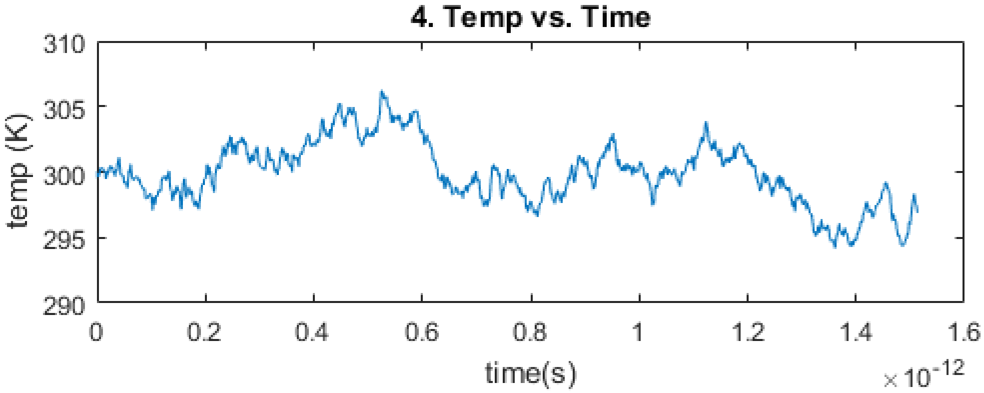
```
%In this section, each particle gets a random velocity.  These
 velocities are normal in the x, and y direction, which produced a
 velocity
%distribution that will follow a maxwell botlzmann distribution.
 This distribution can be seen in figure 5 .At every time step, each
 particle has a small probably of scattering (0.75% with the current
 time step).  The average temperature over time will vary, but
```

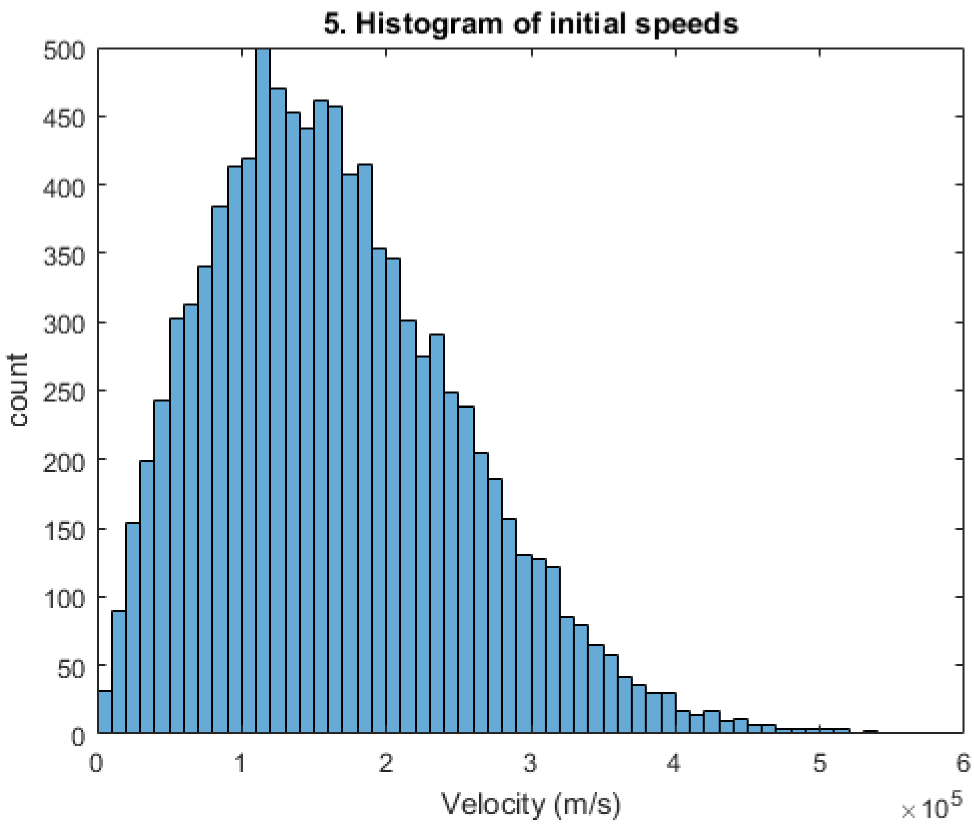
```
%remain close to 300K.  This can be seen in figure 4.  The particles
 trajectories can be seen in figure 3.  Then the mean free path and
 time between the
%collision is found from the simulation

%variables you can edit
num_e = 10000;
x_dim = 200*10^-9;
y_dim = 100e-9;

%colours for plot
col=hsv(10);


Temp_arr=[300];
tau = zeros(1,num_e);
Tau=0;
mfp=0;
count=0;

close all

%get initial positions and velocities
[x_arr,y_arr, vx_arr,vy_arr] = gen_e(num_e,x_dim,y_dim,2);


v = sqrt(vx_arr.*vx_arr + vy_arr.*vy_arr);

vth =132.2e3;

t=0 ;
t_step = max(x_dim,y_dim)/(1000*vth);
Tstop=1000*t_step;
time_arr=zeros(1,10000);
for i=1:length(time_arr)
    time_arr(i)=(i-1)*t_step;
end

%scatter probablity
P_scat=1-exp(-t_step/(.2e-12));

while t< Tstop

    %calculate new velocity if scatter, update mfp, time between
 collisions
    for q= 1:length(x_arr)
        tau(q)=tau(q)+t_step;
        if rand()<P_scat
            Tau=[Tau,tau(q)];
            mfp=[mfp,tau(q)*sqrt(vx_arr(q)^2+vy_arr(q)^2)];
            tau(q)=0;
            vx_arr(q)=132.2e3*randn();
            vy_arr(q)=132.2e3*randn();
```

```
        end
    end

    %calculate temp
    Temp_arr = [Temp_arr,(1/2)/
(1.3806e-23)*9.109e-31*.26*(mean(vx_arr.^2)+mean(vy_arr.^2))];

    %add the time step to the position
    xp_arr=x_arr;
    xg_arr=x_arr;
    yp_arr=y_arr;
    yg_arr=y_arr;
    x_arr=x_arr+vx_arr*t_step;
    y_arr=y_arr+vy_arr*t_step;


    %check to see if anything is out of bounds
    for q=1:num_e
       if x_arr(q)<0
           x_arr(q)=x_arr(q)+x_dim;
           xg_arr(q)=x_dim;
       end
       if x_arr(q) > x_dim
           x_arr(q)=x_arr(q)-x_dim;
           xg_arr(q)=0;
       end
       if y_arr(q)>y_dim
           vy_arr(q)=-vy_arr(q);
           y_arr(q)=2*y_dim-y_arr(q);
       end
       if y_arr(q)<0
           vy_arr(q)=-vy_arr(q);
           y_arr(q)=abs(y_arr(q));
       end
    end

    %position plot
    subplot(2,1,1)
    xlabel('X(m)')
    ylabel('Y(m)')
    title('3. Position of particles')
    xlim([0 x_dim])
    ylim([0 y_dim])
    pause(.01)
    for q=1:10
        plot([xg_arr(q);x_arr(q)],
[yg_arr(q);y_arr(q)],'color',col(q,:))
        hold on
    end


    t=t+t_step;
    count=count+1;
```

```
    %temperature plot
    subplot(2,1,2)
    plot(time_arr(1:length(Temp_arr)),Temp_arr)
    xlabel('time(s)')
    ylabel('temp (K)')
    title('4. Temp vs. Time')


end
figure(2)
histogram(v)
xlabel('Velocity (m/s)')
ylabel('count')
title('5. Histogram of initial speeds')
mean_free_path = mean(mfp(2:length(mfp)))
%meters

time_between_collisions=mean(Tau(2:length(Tau)))
%seconds


mean_free_path =

   2.8901e-08


time_between_collisions =

   1.7407e-13
```


3. Position of particles
Y(m)
X(m)
×10^-7
0
0.5
1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2


4. Temp vs. Time
temp (K)
time(s)
×10^-12
290
295
300
305
310
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6


5. Histogram of initial speeds
count
Velocity (m/s)
×10^5
0
50
100
150
200
250
300
350
400
450
500
0
1
2
3
4
5
6

Equations

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

Question 3

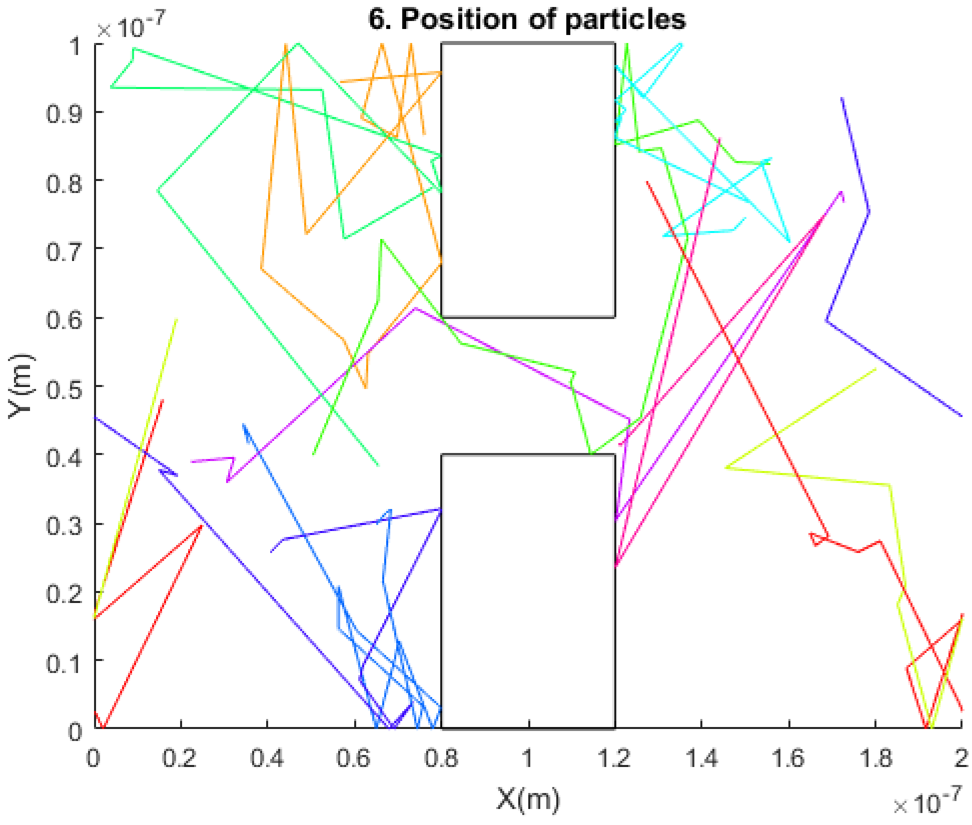
```
%This section of code models the flow of electrons in a 200nm by 100nm
 box
%with two rectangle boundaries.  These boundaries can be specular or
%diffusive (currently set to diffusive).  Every time a particle
 strikes a
%boundary, it gains a new velocity.  The patricles paths can be seen
 in figure 6.This code also produces an electron
%density map seen in figure 7, and a temperature density map seen in
 figure
%8.


%variables you can edit
num_e = 10000;
x_dim = 200*10^-9;
y_dim = 100e-9;
retherm=1; %rethermalize variable.  1 to activate, 0 to deactivate

col=hsv(10);
Temp_arr=[300];
tau = zeros(1,num_e);
Tau=0;
mfp=0;
count=0;

close all
hold off

[x_arr,y_arr, vx_arr,vy_arr] = gen_e(num_e,x_dim,y_dim,3);


vth =132.2e3;
%vth in m/s

t=0 ;
t_step = max(x_dim,y_dim)/(1000*vth);
Tstop=1000*t_step;
time_arr=zeros(1,10000);
for i=1:length(time_arr)
    time_arr(i)=(i-1)*t_step;
end

P_scat=1-exp(-t_step/(.2e-12));
hold off

%define boundary outline in figure
```

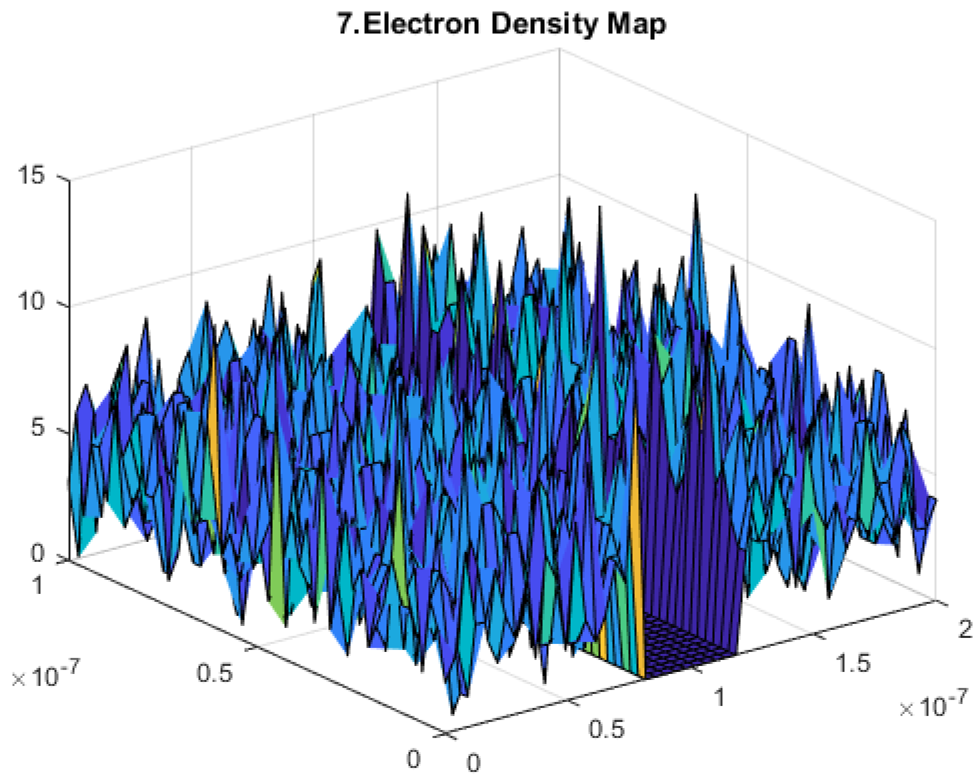
```
figure(1)
hold on
rectangle('position',[0.4*x_dim,0,0.2*x_dim,0.4*y_dim])
rectangle('position',[0.4*x_dim,0.6*y_dim,0.2*x_dim,0.4*y_dim])

while t< Tstop

    %calculate new velocity if scattering occurs

    for q= 1:length(x_arr)
        tau(q)=tau(q)+t_step;
        if rand()<P_scat
            Tau=[Tau,tau(q)];
            mfp=[mfp,tau(q)*sqrt(vx_arr(q)^2+vy_arr(q)^2)];
            tau(q)=0;
            vx_arr(q)=132.2e3*randn();
            vy_arr(q)=132.2e3*randn();

        end
    end

    %calculate temperature

    Temp_arr = [Temp_arr,(1/2)/
(1.3806e-23)*9.109e-31*.26*(mean(vx_arr.^2)+mean(vy_arr.^2))];

    %add the time step to the position
    xp_arr=x_arr;
    xg_arr=x_arr;
    yp_arr=y_arr;
    yg_arr=y_arr;
    x_arr=x_arr+vx_arr*t_step;
    y_arr=y_arr+vy_arr*t_step;


    %check to see if anything is out of bounds
    for q=1:num_e
       if x_arr(q)<0
           x_arr(q)=x_arr(q)+x_dim;
           xg_arr(q)=x_dim;
       end
       if x_arr(q) > x_dim
           x_arr(q)=x_arr(q)-x_dim;
           xg_arr(q)=0;
       end
       if y_arr(q)>y_dim
           vy_arr(q)=-vy_arr(q);
           y_arr(q)=2*y_dim-y_arr(q);
       end
       if y_arr(q)<0
           vy_arr(q)=-vy_arr(q);
           y_arr(q)=abs(y_arr(q));
       end
```

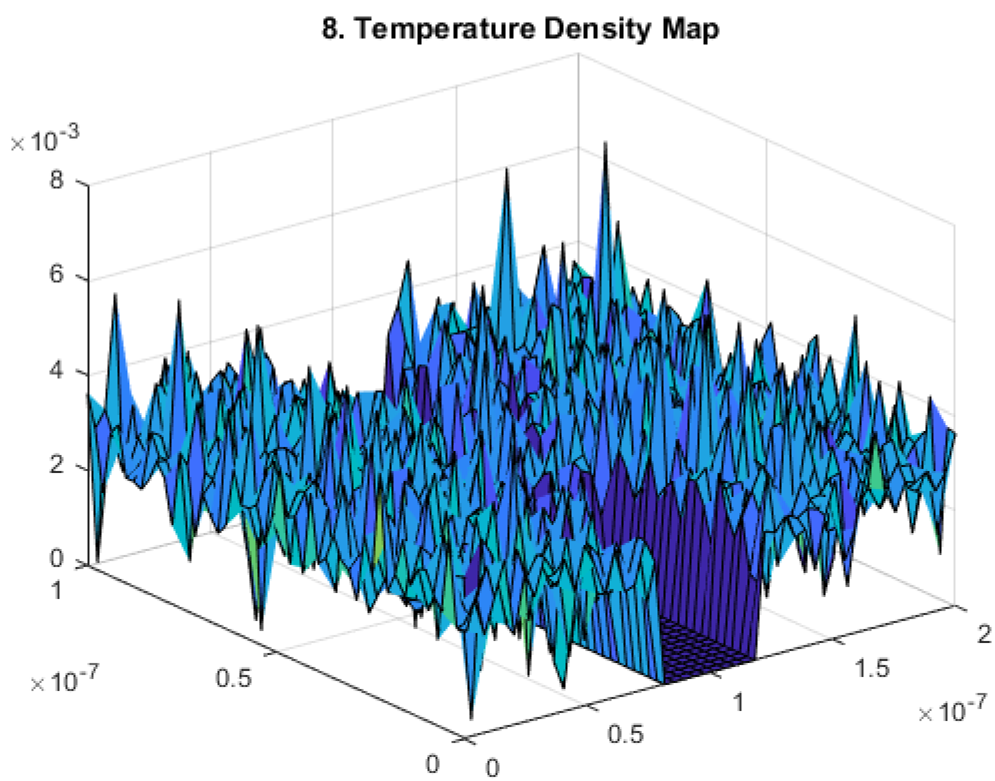
```
        %bot box boundary
        if y_arr(q)<0.4*y_dim && x_arr(q)>0.4*x_dim &&
x_arr(q)<0.6*x_dim
            if y_arr(q)<0.4*y_dim && yp_arr(q)>0.4*y_dim
                y_arr(q)=abs(y_arr(q)-0.4*y_dim)+0.4*y_dim;
                if retherm
                     vy_arr(q)=(132.2e3)*abs(randn(1));
                     vx_arr(q)=132.2e3*randn(1);
                else
                     vy_arr(q)= -vy_arr(q);
                end
            end
            if x_arr(q)>0.4*x_dim && xp_arr(q)<0.4*x_dim
                x_arr(q)=0.4*x_dim-abs(x_arr(q)-0.4*x_dim);
                if retherm
                     vx_arr(q)= -(132.2e3)*abs(randn(1));
                     vy_arr(q)=132.2e3*randn(1);

                else
                     vx_arr(q)= -vx_arr(q);
                end

            end
            if x_arr(q)<0.6*x_dim && xp_arr(q)>0.6*x_dim
                x_arr(q)=abs(x_arr(q)-0.6*x_dim)+0.6*x_dim;
                if retherm
                     vx_arr(q)= (132.2e3)*abs(randn(1));
                     vy_arr(q)=132.2e3*randn(1);
                else
                     vx_arr(q)= -vx_arr(q);
                end
            end
        end

        %top box boundary
        if y_arr(q)>0.6*y_dim && x_arr(q)>0.4*x_dim &&
x_arr(q)<0.6*x_dim
            if y_arr(q)>0.6*y_dim && yp_arr(q)<0.6*y_dim
                if retherm
                    vy_arr(q)=(132.2e3)*(-abs(randn(1)));
                    vx_arr(q)=132.2e3*randn(1);
                else
                    vy_arr(q)= -vy_arr(q);
                end
                y_arr(q)=1.2*(y_dim)-y_arr(q);
            end
            if x_arr(q)>0.4*x_dim && xp_arr(q)<0.4*x_dim
                if retherm
                     vx_arr(q)= -(132.2e3)*abs(randn(1));
                     vy_arr(q)=132.2e3*randn(1);
                else
                     vx_arr(q)=-vx_arr(q);
                end
                x_arr(q)=0.4*x_dim-abs(x_arr(q)-0.4*x_dim);
```

```
            end
            if x_arr(q)<0.6*x_dim && xp_arr(q)>0.6*x_dim
                if retherm
                     vx_arr(q)= (132.2e3)*abs(randn(1));
                     vy_arr(q)=132.2e3*randn(1);
                else
                     vx_arr(q)=-vx_arr(q);
                end
                x_arr(q)=abs(x_arr(q)-0.6*x_dim)+0.6*x_dim;
            end
        end


    end
    %plot positions
    xlabel('X(m)')
    ylabel('Y(m)')
    title('6. Position of particles')
    xlim([0 x_dim])
    ylim([0 y_dim])
    pause(.01)
    for q=1:10
        plot([xg_arr(q);x_arr(q)],
[yg_arr(q);y_arr(q)],'color',col(q,:))
        hold on
    end


    t=t+t_step;
    count=count+1;
end
p=zeros(50);
v=zeros(50);
temp=zeros(50);

%make the density maps
for q=1:50
    for w=1:50
        for n=1:num_e
            if
 x_arr(n)>=(((q-1)*x_dim/50))&&(x_arr(n)<(q*x_dim/50))&&(y_arr(n)>=(w-1)*y_dim/50
                p(w,q)=p(w,q)+1;
                v(w,q)=v(w,q)+sqrt(vx_arr(n)^2+vy_arr(n)^2);
            end
        end
        if p(w,q)==0
            temp(w,q)=0;

        else
            temp(w,q)=0.26*9.109e-31*v(w,q)/p(w,q)/(1.3806e-23);
        end

    end
end
```

```
figure(2)
surf(linspace(0,x_dim,50),linspace(0,y_dim,50),p)
title('7.Electron Density Map')

figure (3)
surf(linspace(0,x_dim,50),linspace(0,y_dim,50),temp)
title('8. Temperature Density Map')
```




7.Electron Density Map
15
10
5
0
1
$\times 10^{-7}$
0.5
0
0
0.5
1
1.5
2
$\times 10^{-7}$


8. Temperature Density Map
$\times 10^{-3}$
8
6
4
2
0
1
$\times 10^{-7}$
0.5
0
0
0.5
1
1.5
2
$\times 10^{-7}$

Published with MATLAB® R2017a