Ovwagbedia Oghenero

Table of Contents

Student no: 101040228

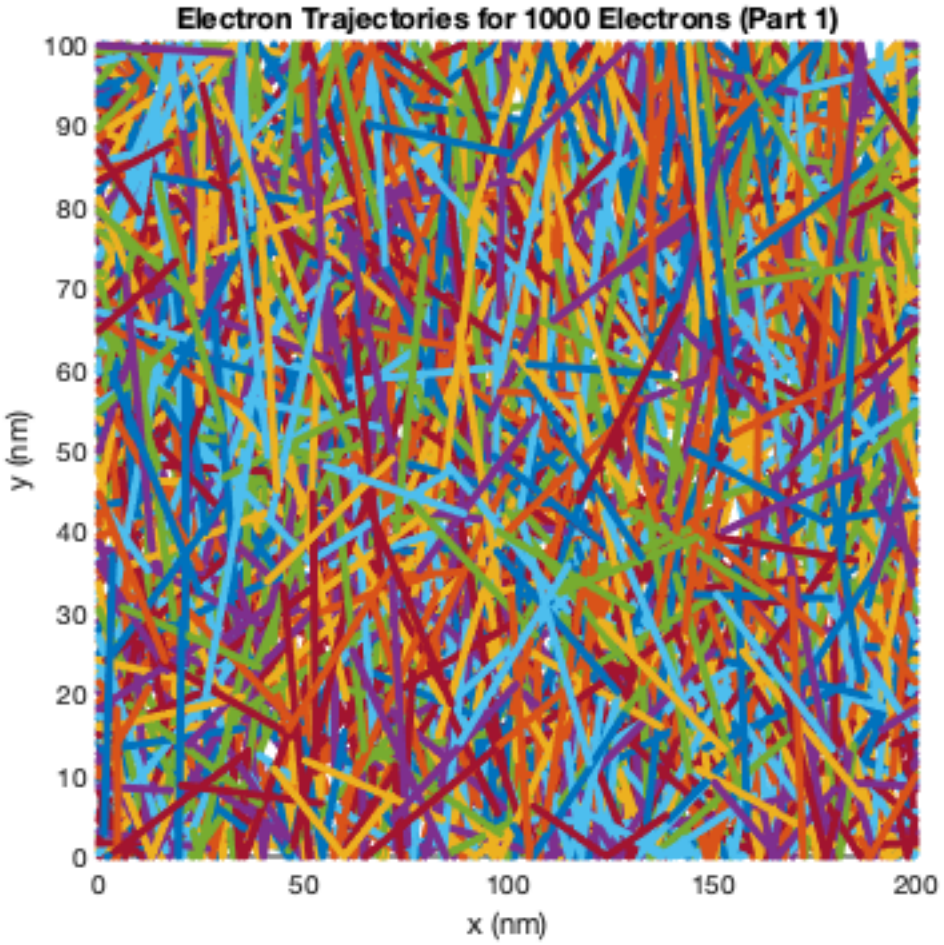
Part 1: Electron Modelling

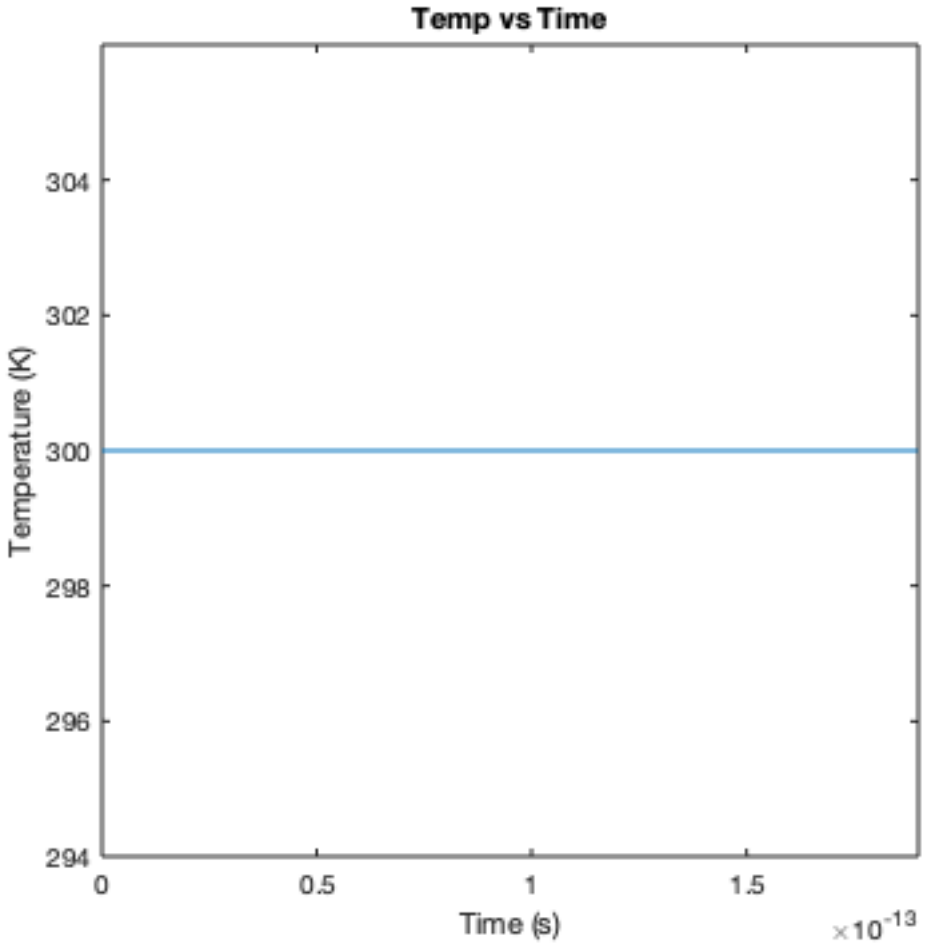
declaration of constants

```
k = 1.28e-23; %J/K??
mo = 9.1e-31; %kg
mn = 0.26 * mo; %effective mass of electrons
T = 300; %K
v_th = sqrt((2*k*T)/mn); %solving for thermal velocity
tmn = 0.2e-12; %seconds(mean time between collisions)
%electrons
Nom =  1000;
step = 1000;
movie = 0;


%Mean free path
meanFP = v_th * tmn

%nominal size of region
width = 200e-9; %metres
height = 100e-9; %metres
area = width * height %the area of the region
%spacial step
t_step0 = 0.01 * area;
%ideal spacial step
t_step = t_step0 - 0.1e-16; % smaller than 1/100 of region


%position and velocity
pos = zeros(Nom,4);
traj = zeros(step,Nom*2);
temp = zeros(step,1);

%initial condition
for i = 1:Nom
```

```
    ang = 2*pi*rand;
    pos(i,:) = [width*rand height*rand v_th*cos(ang) v_th*sin(ang)];
end

% updating position
for i = 1:step
    pos(:,1:2) = pos(:,1:2) + t_step*pos(:,3:4);

    %For side collision
    j = pos(:,1) > width;
    pos(j,1) = pos(j,1) - width;

    j = pos(:,1) < 0;
    pos(j,1) = pos(j,1) + width;

    %For bottom and top colission
    j = pos(:,2) > height;
    pos(j,2) = 2*height - pos(j,2);
    pos(j,4) = -pos(j,4);

    j = pos(:,2) < 0;
    pos(j,2) = -pos(j,2);
    pos(j,4) = -pos(j,4);

    temp(i) = (sum(pos(:,3).^2) + sum(pos(:,4).^2))*mn/k/2/Nom;

    %trajectory
    for j = 1:Nom
        traj(i, (2*j):(2*j+1)) = pos(j, 1:2);
    end

    %update movie after some iterations
    if movie && mod(i,10) == 0
        figure(1);
        hold off;
        plot(pos(1:Nom,1)./1e-9, pos(1:Nom,2)./1e-9, 'o');
        axis([0 width/1e-9 0 height/1e-9]);
        title(sprintf('Trajectories for %d Electrons (Part 1)',...
        Nom));
        xlabel('x (nm)');
        ylabel('y (nm)');

        if i > 1
            figure (2);
            hold off;
            plot(t_step*(0:i-1), temp(1:i));
            axis([0 t_step*step min(temp)*0.98 max(temp)*1.02]);
            title('Semiconductor Temperature');
            xlabel('Time (s)');
            ylabel('Temperature (K)');
        end
        pause(0.05);
    end
```

```
end

 %trajectory after movie
 figure (1);
 title(sprintf('Electron Trajectories for %d Electrons (Part 1)',...
 Nom));
 xlabel('x (nm)');
 ylabel('y (nm)');
 axis([0 width/1e-9 0 height/1e-9]);
 hold on;

 for i=1:Nom
    plot(traj(:,i*2)./1e-9, traj(:,i*2+1)./1e-9, '.');
 end

 figure(2);
 hold off;
 plot(t_step*(0:step-1), temp);
 axis([0 t_step*step min(temp)*0.98 max(temp)*1.02]);
 title('Temp vs Time');
 xlabel('Time (s)');
 ylabel('Temperature (K)');


meanFP =

   3.6033e-08


area =

   2.0000e-14
```


Electron Trajectories for 1000 Electrons (Part 1)
y (nm)
x (nm)
0
10
20
30
40
50
60
70
80
90
100
0
50
100
150
200

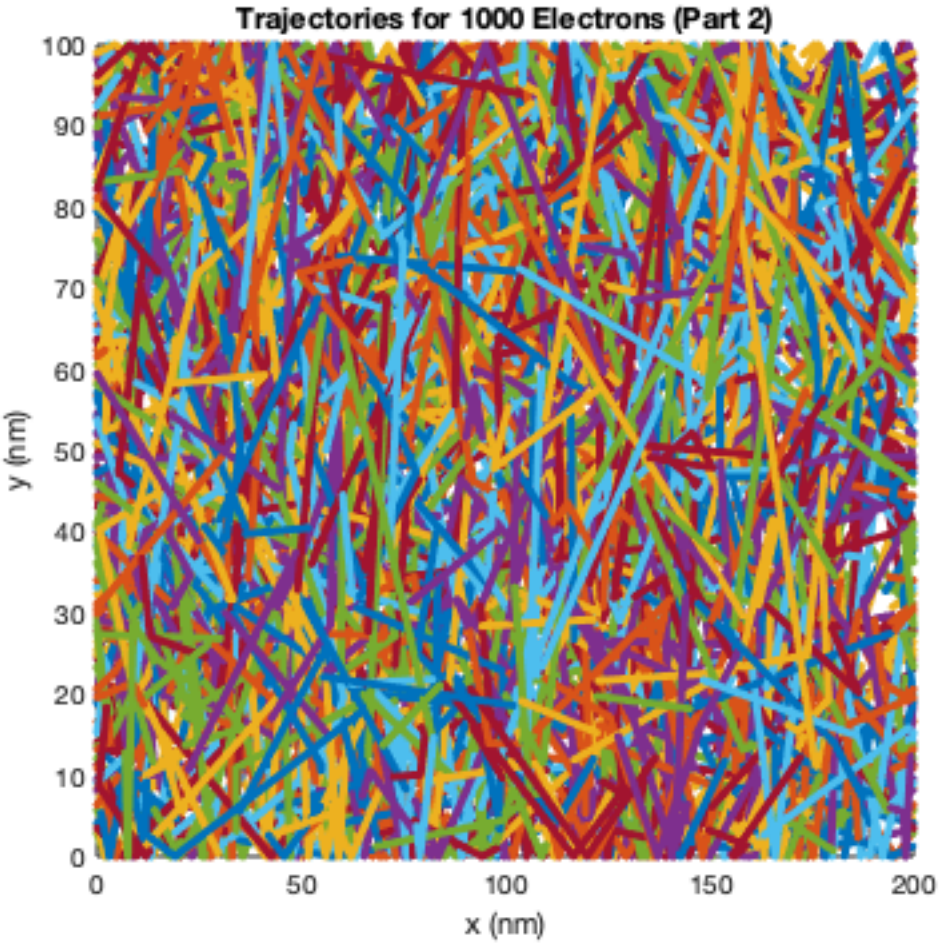


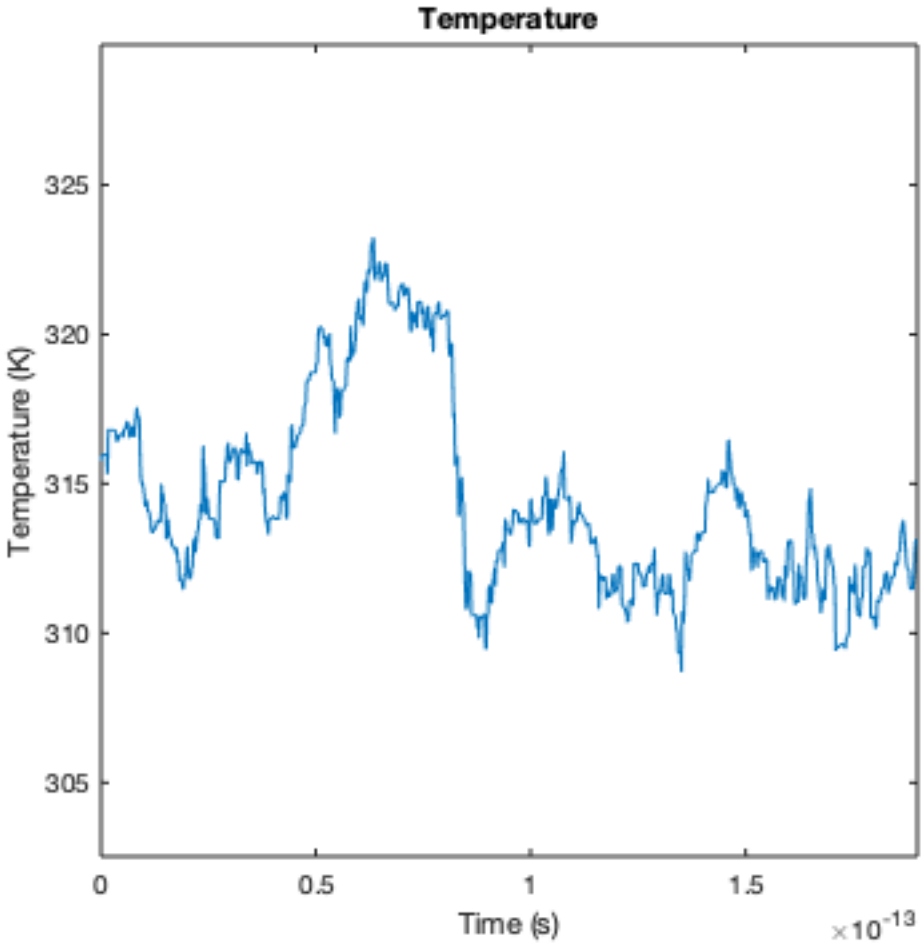
Part 2:Collisions with Mean Free Path

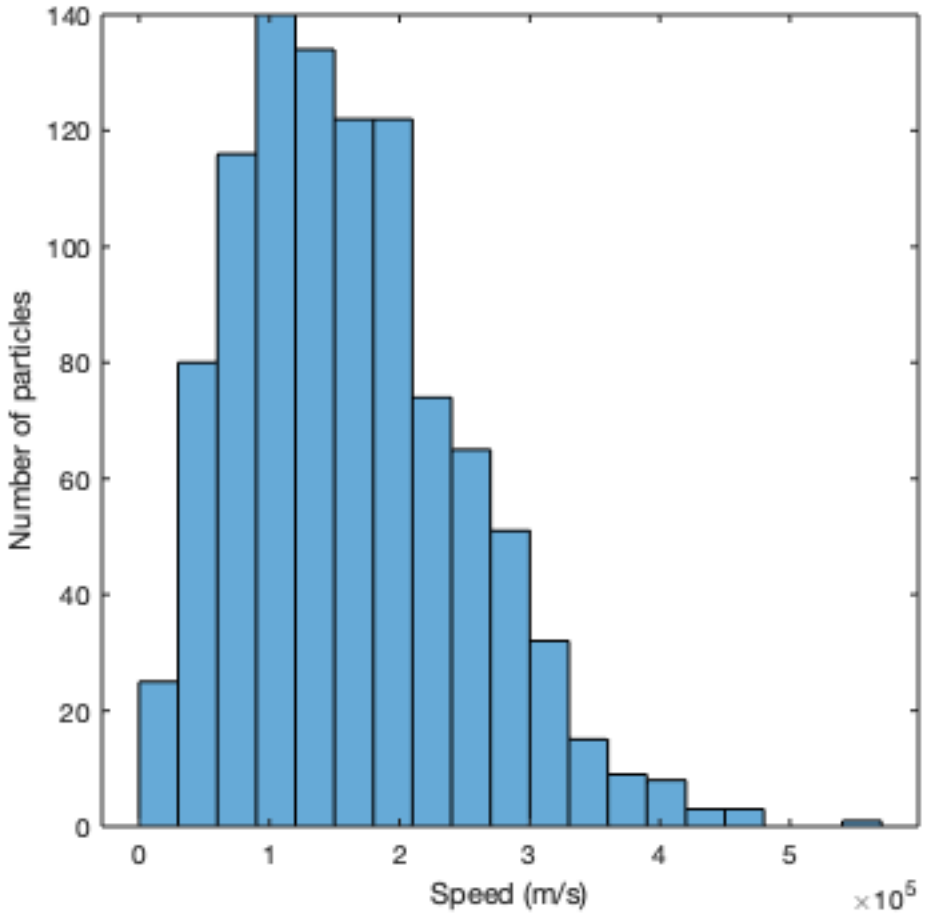
```
 %probability of scattering in a time step
 p_Scat = 1 - exp(-t_step/tmn);

 %Velecity in x and y is gaussian
 %therefore the overall is a Maxwell-Boltzman distribution
 v_o = makedist('Normal', 'mu', 0, 'sigma', sqrt(k*T/mn));

 %initial condition
 for i = 1:Nom
    ang = rand*2*pi;
    pos(i,:) = [width*rand height*rand random(v_o) random(v_o)];
end

%average velocity calc
avg_vel = sqrt(sum(pos(:,3).^2)/Nom + ...
    sum(pos(:,4).^2)/Nom)

% updating position
for i = 1:step
```

```
pos(:,1:2) = pos(:,1:2) + t_step.*pos(:,3:4);

  j = pos(:,1) > width;
  pos(j,1) = pos(j,1) - width;

  j = pos(:,1) < 0;
  pos(j,1) = pos(j,1) + width;

  j = pos(:,2) > height;
  pos(j,2) = 2*height - pos(j,2);
  pos(j,4) = -pos(j,4);

  j = pos(:,2) < 0;
  pos(j,2) = -pos(j,2);
  pos(j,4) = -pos(j,4);

  %scatter
  j = rand(Nom, 1) < p_Scat;
  pos(j,3:4) = random(v_o, [sum(j),2]);

  temp(i) = (sum(pos(:,3).^2) + sum(pos(:,4).^2))*mn/k/2/Nom;

  %Trajectory
  for j=1:Nom
      traj(i, (2*j):(2*j+1)) = pos(j, 1:2);
  end

  %updating movie after certain iterations
  if movie && mod(i,10) == 0
      figure(3);
      hold off;
      plot(pos(1:Nom,1)./1e-9, pos(1:Nom,2)./1e-9, 'o');
      axis([0 width/1e-9 0 height/1e-9]);
      title(sprintf('Trajectories for %d Electrons (Part 2)',...
      Nom));
      xlabel('x (nm)');
      ylabel('y (nm)');
      if i > 1
          figure (4);
          hold off;
          plot(t_step*(0:i-1), temp(1:i));
          axis([0 t_step*step min(temp)*0.98 max(temp)*1.02]);
          title('Temperature');
          xlabel('Time (s)');
          ylabel('Temperature (K)');
      end

      % histogram plot
      figure (5);
      vel = sqrt(pos(:,3).^2 + pos(:,4).^2);
      title('Histogram of Electron Speeds');
      histogram(vel);
      xlabel('Speed (m/s)');
      ylabel('Number of particles');
```

```
        pause(0.05);
    end
end

%trajectory after movie
figure(3);
title(sprintf('Trajectories for %d Electrons (Part 2)',...
    Nom));
xlabel('x (nm)');
ylabel('y (nm)');
axis([0 width/1e-9 0 height/1e-9]);
hold on;

for i=1:Nom
    plot(traj(:,i*2)./1e-9, traj(:,i*2+1)./1e-9, '.');
end

figure (4);
hold off;
plot(t_step*(0:step-1), temp);
axis([0 t_step*step min(temp)*0.98 max(temp)*1.02]);
title('Temperature');
xlabel('Time (s)');
ylabel('Temperature (K)');

%histogram plot
figure (5);
vel = sqrt(pos(:,3).^2 + pos(:,4).^2);
title('Histogram of Electron Speeds');
histogram(vel);
xlabel('Speed (m/s)');
ylabel('Number of particles');


avg_vel =

   1.8473e+05
```


Trajectories for 1000 Electrons (Part 2)
y (nm)
x (nm)
0
10
20
30
40
50
60
70
80
90
100
0
50
100
150
200


Temperature
325
320
315
310
305
Temperature (K)
0
0.5
1
1.5
Time (s)
×10^-13



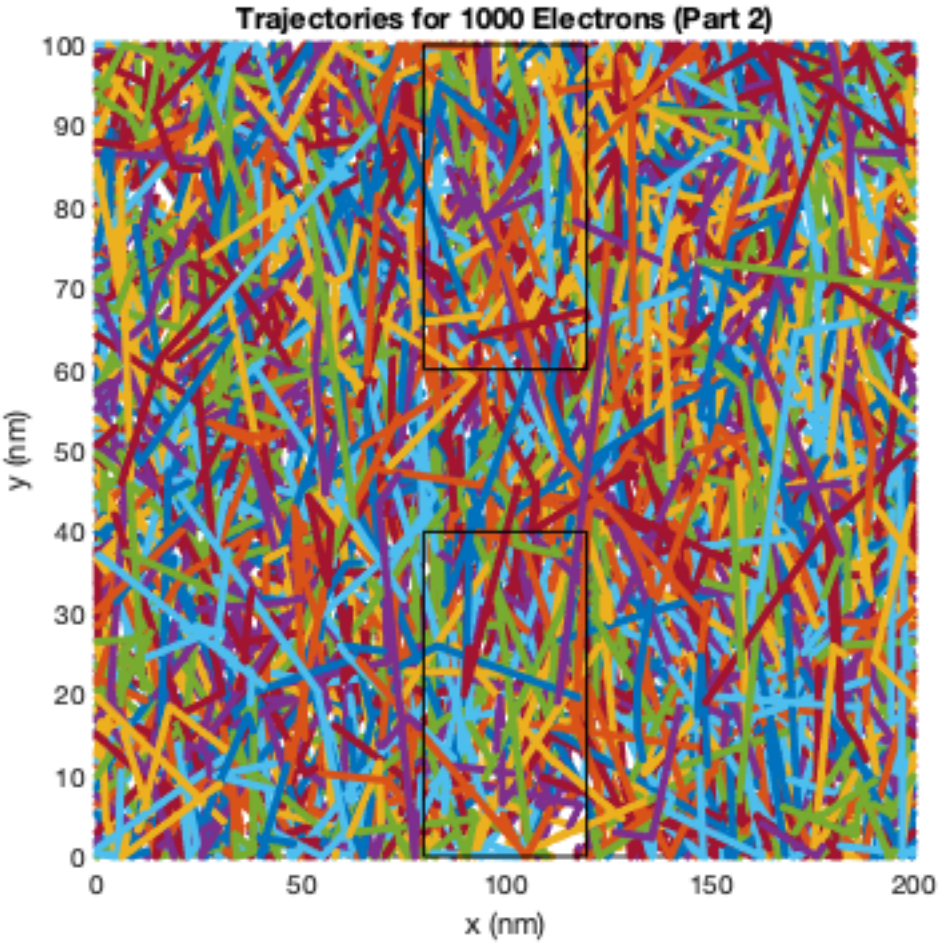
Part3: Enhancements

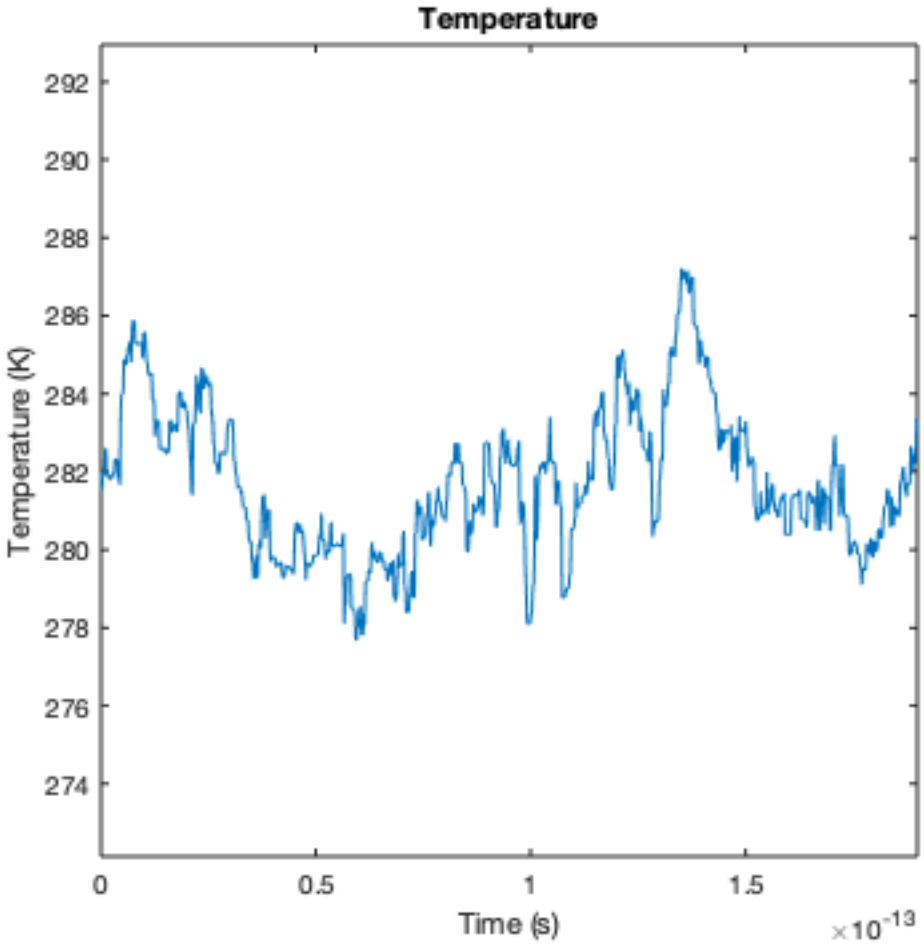
```
%Here, boundaries are either specular or diffusive

%Specular or diffusive boundaries
%diffusive = 0
%specular = 1
tb = 0;
bb = 0;
box = [0 1];
box1 = [80 120 0 40];
box2 = [80 120 60 100];
boxes = 1e-9 .* [box1; box2];

%initial condition
for i = 1:Nom
    ang = rand*2*pi;
    pos(i,:) = [width*rand height*rand random(v_o) random(v_o)];


end
```

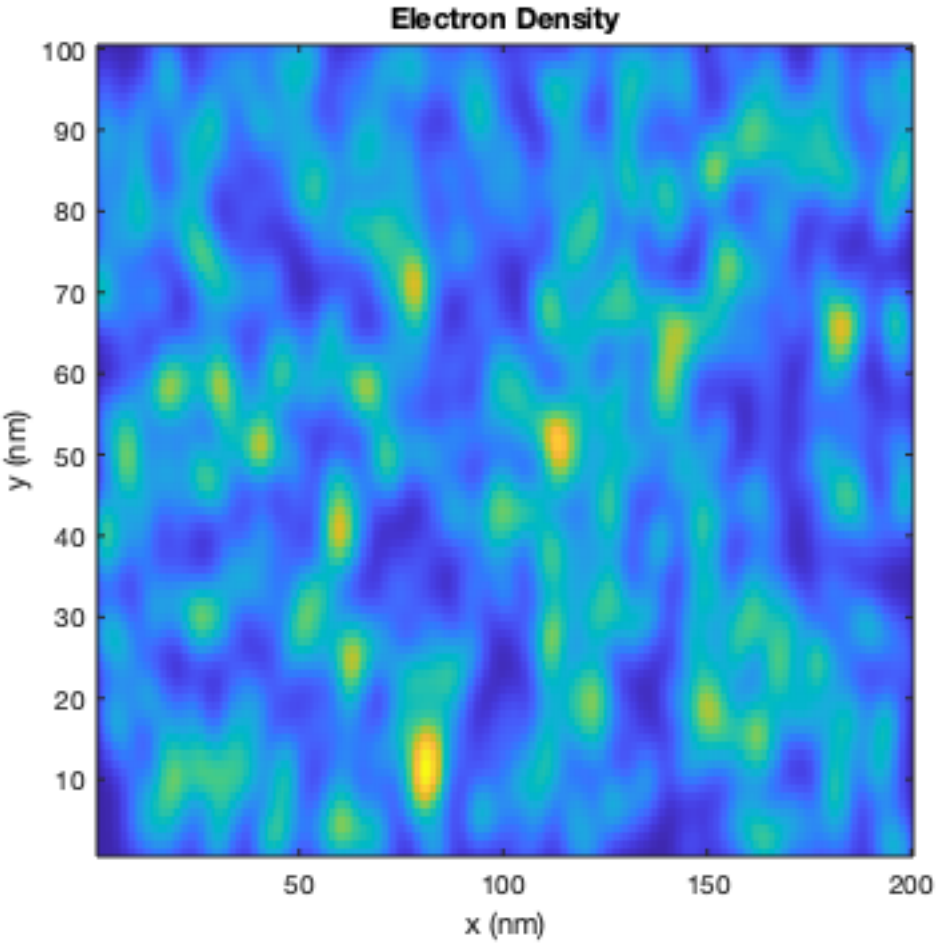
```
%third simulation
for i = 1:step
    pos(:,1:2) = pos(:,1:2) + t_step.*pos(:,3:4);

    j = pos(:,1) > width;
    pos(j,1) = pos(j,1) - width;

    j = pos(:,1) < 0;
    pos(j,1) = pos(j,1) + width;

    j = pos(:,2) > height;

    if(tb)
        pos(j,2) = 2*height - pos(j,2);
        pos(j,4) = -pos(j,4);
    else
        % Diffusive and electron bounce off  a random angle
        pos(j,2) = height;
        v = sqrt(pos(j,3).^2 + pos(j,4).^2);
        ang = rand([sum(j),1])*2*pi;
        pos(j,3) = v.*cos(ang);
        pos(j,4) = -abs(v.*sin(ang));
    end

    j = pos(:,2) < 0;

    if(bb)
        pos(j,2) = -pos(j,2);
        pos(j,4) = -pos(j,4);
    else
        % Diffusive and electron bounce off  a random angle
        pos(j,2) = 0;
        v = sqrt(pos(j,3).^2 + pos(j,4).^2);
        ang = rand([sum(j),1])*2*pi;
        pos(j,3) = v.*cos(ang);
        pos(j,4) = abs(v.*sin(ang));
    end

    %scatter
    j = rand(Nom, 1) < p_Scat;
    pos(j,3:4) = random(v_o, [sum(j),2]);

    temp(i) = (sum(pos(:,3).^2) + sum(pos(:,4).^2))*mn/k/2/Nom;

    %Trajectory
    for j=1:Nom
        traj(i, (2*j):(2*j+1)) = pos(j, 1:2);
    end

    %update movie after some iterations
    if movie && mod(i,10) == 0
        figure(6);
        hold off;
```

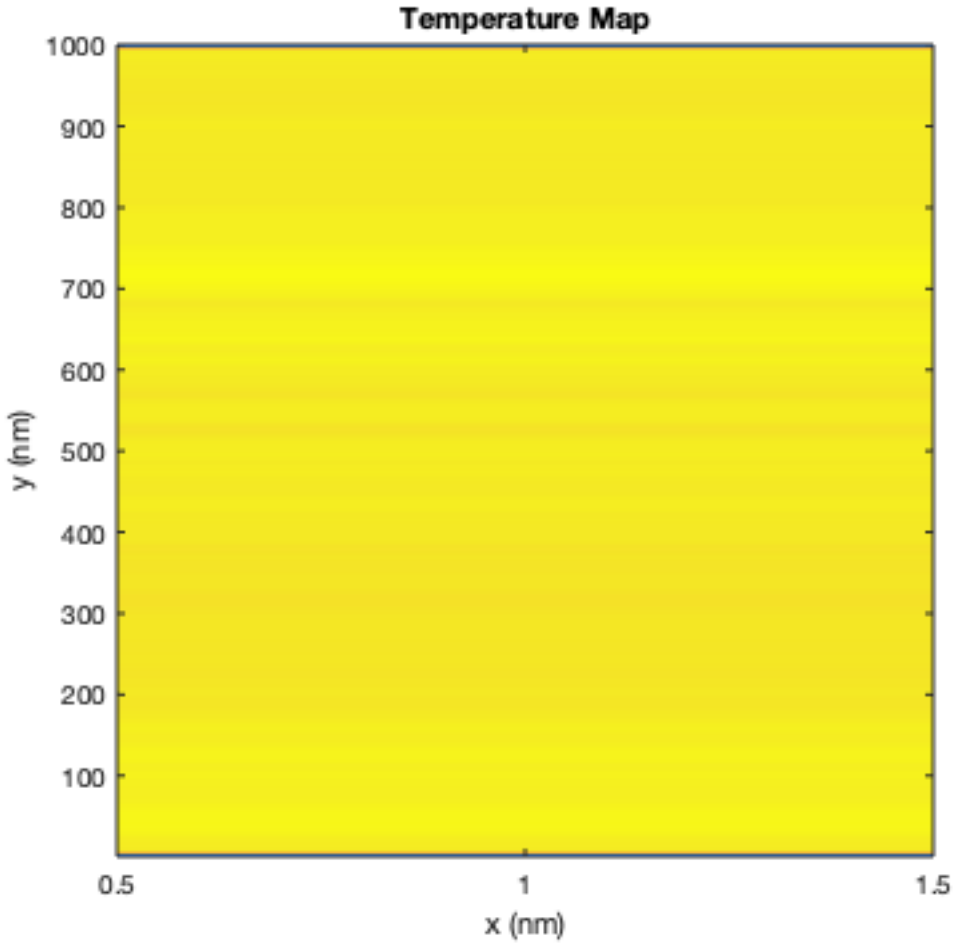
```
        plot(pos(1:Nom,1)./1e-9, pos(1:Nom,2)./1e-9, 'o');

        %plot boxes
        for j=1:size(boxes,1)
           plot([boxes(j, 1) boxes(j, 1) boxes(j, 2) boxes(j, 2)
 boxes(j, 1)]./1e-9,...
                [boxes(j, 3) boxes(j, 4) boxes(j, 4) boxes(j, 3)
 boxes(j, 3)]./1e-9, 'k-');
        end

        axis([0 width/1e-9 0 height/1e-9]);
        title(sprintf('Trajectories for %d Electrons (Part 3)',...
        Nom));
        xlabel('x (nm)');
        ylabel('y (nm)');
        if i > 1
            figure (7);
            hold off;
            plot(t_step*(0:i-1), temp(1:i));
            axis([0 t_step*step min(temp)*0.98 max(temp)*1.02]);
            title('Temperature');
            xlabel('Time (s)');
            ylabel('Temperature (K)');
        end
        pause(0.05);
    end
end

%trajectory after movie
figure(6);
title(sprintf('Trajectories for %d Electrons (Part 2)',...
    Nom));
xlabel('x (nm)');
ylabel('y (nm)');
axis([0 width/1e-9 0 height/1e-9]);
hold on;

for i=1:Nom
    plot(traj(:,i*2)./1e-9, traj(:,i*2+1)./1e-9, '.');
end

for j=1:size(boxes,1)
   plot([boxes(j, 1) boxes(j, 1) boxes(j, 2) boxes(j, 2) boxes(j,
 1)]./1e-9,...
       [boxes(j, 3) boxes(j, 4) boxes(j, 4) boxes(j, 3) boxes(j,
 3)]./1e-9, 'k-');
end

figure (7);
hold off;
plot(t_step*(0:step-1), temp);
axis([0 t_step*step min(temp)*0.98 max(temp)*1.02]);
title('Temperature');
xlabel('Time (s)');
```

```
ylabel('Temperature (K)');
%Electron density map using a histogram
density = hist3(pos(:,1:2),[200 100])';
N = 20;
sigma = 3;
[x, y]=meshgrid(round(-N/2):round(N/2), round(-N/2):round(N/2));
f=exp(-x.^2/(2*sigma^2)-y.^2/(2*sigma^2));
f=f./sum(f(:));
figure(8);
imagesc(conv2(density,f,'same'));
set(gca,'YDir','normal');
title('Electron Density');
xlabel('x (nm)');
ylabel('y (nm)');

%Temperature Map
N = 20;
sigma = 3;
[x y]=meshgrid(round(-N/2):round(N/2), round(-N/2):round(N/2));
f=exp(-x.^2/(2*sigma^2)-y.^2/(2*sigma^2));
f=f./sum(f(:));
figure(9);
imagesc(conv2(temp,f,'same'));
set(gca,'YDir','normal');
title('Temperature Map');
xlabel('x (nm)');
ylabel('y (nm)');
```


Trajectories for 1000 Electrons (Part 2)
y (nm)
x (nm)
0
10
20
30
40
50
60
70
80
90
100
0
50
100
150
200


Temperature
292
290
288
286
284
282
280
278
276
274
Temperature (K)
0
0.5
1
1.5
Time (s)
×10^-13


Electron Density
100
90
80
70
60
50
40
30
20
10
y (nm)
50
100
150
200
x (nm)



Published with MATLAB® R2019b