Assignment 1 plots:

Code can be found in the appendix section of this document:

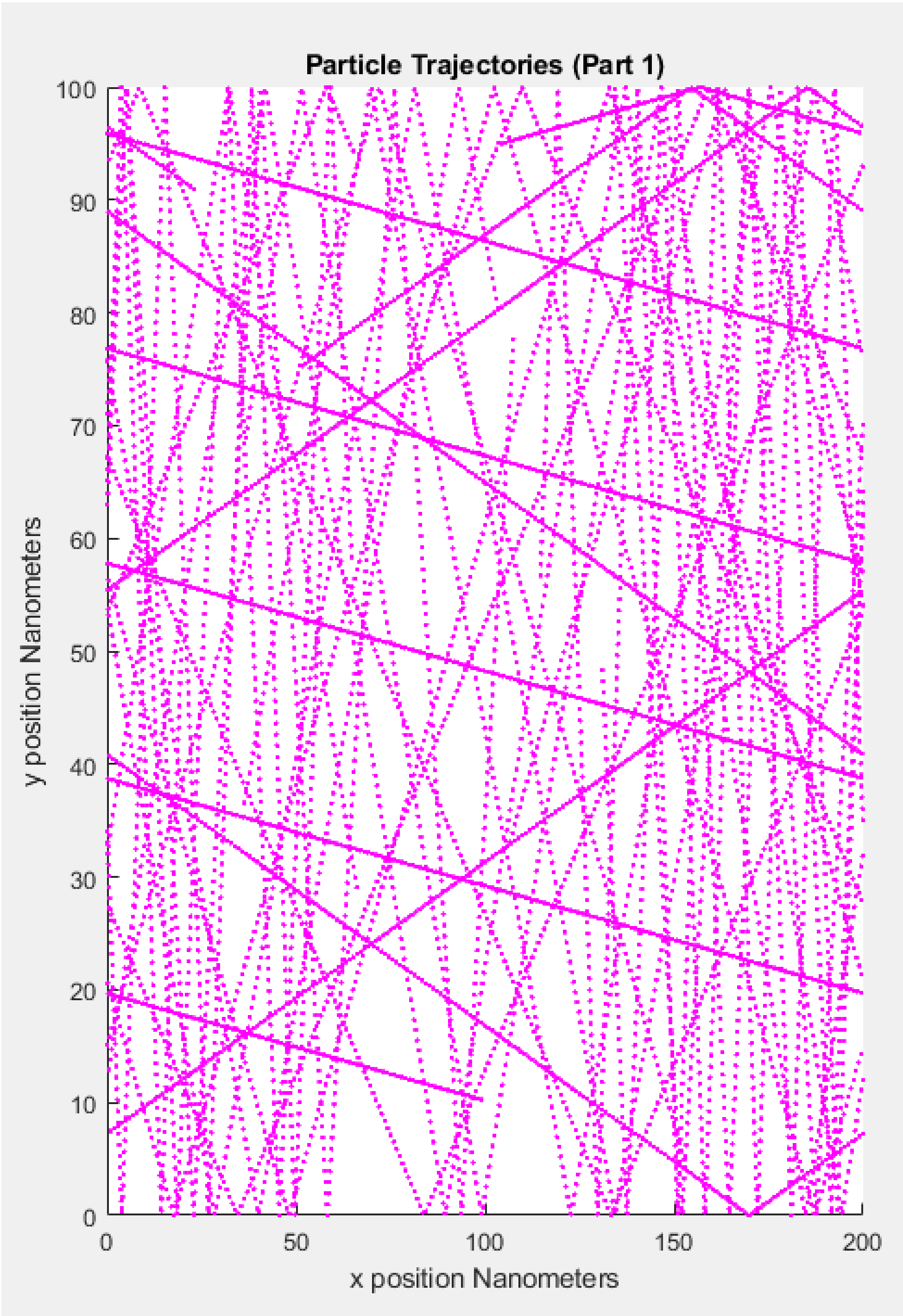

Figure 1: Particle trajectory without bottlenecks implemented

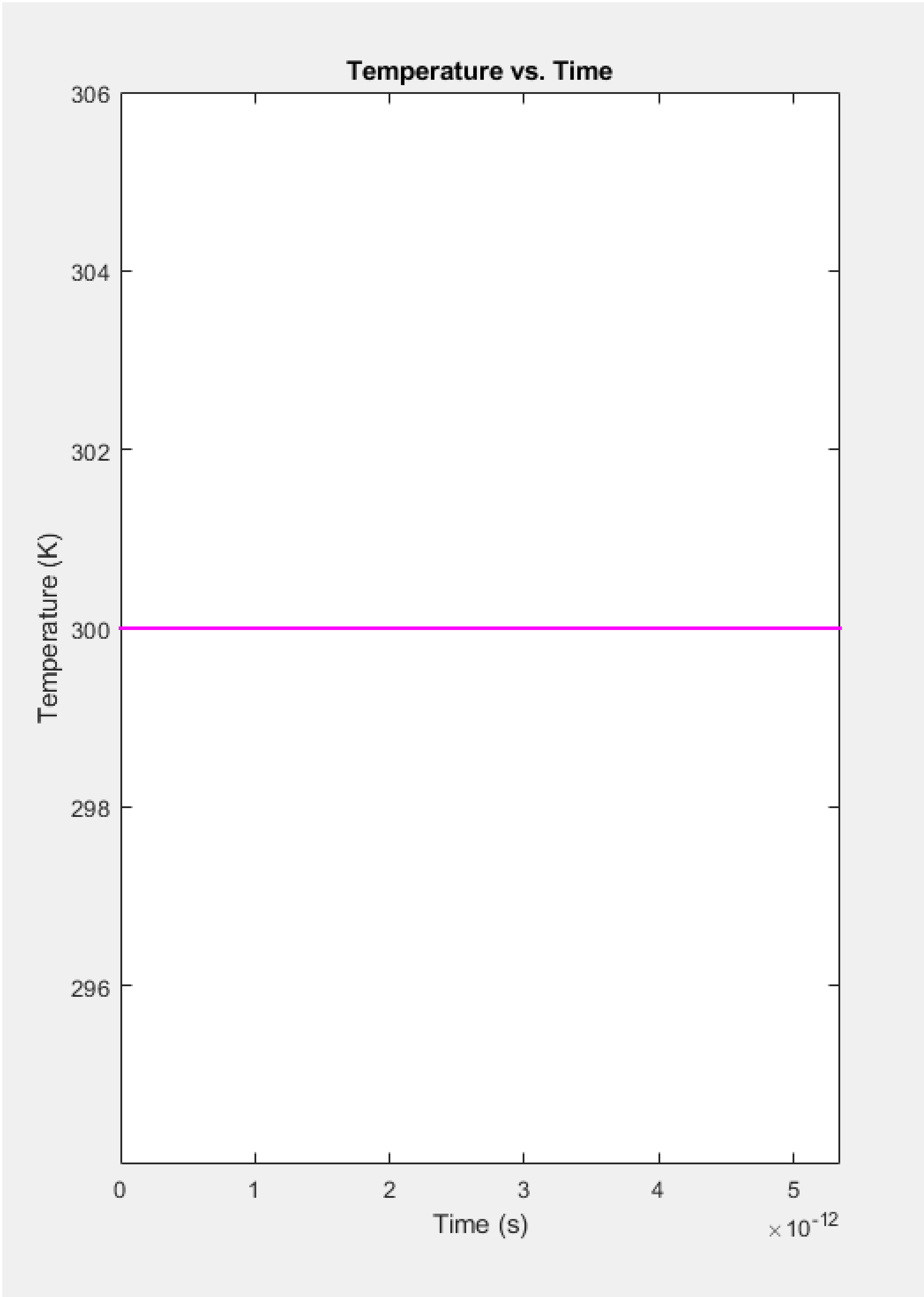

Figure 2: Temperature vs time

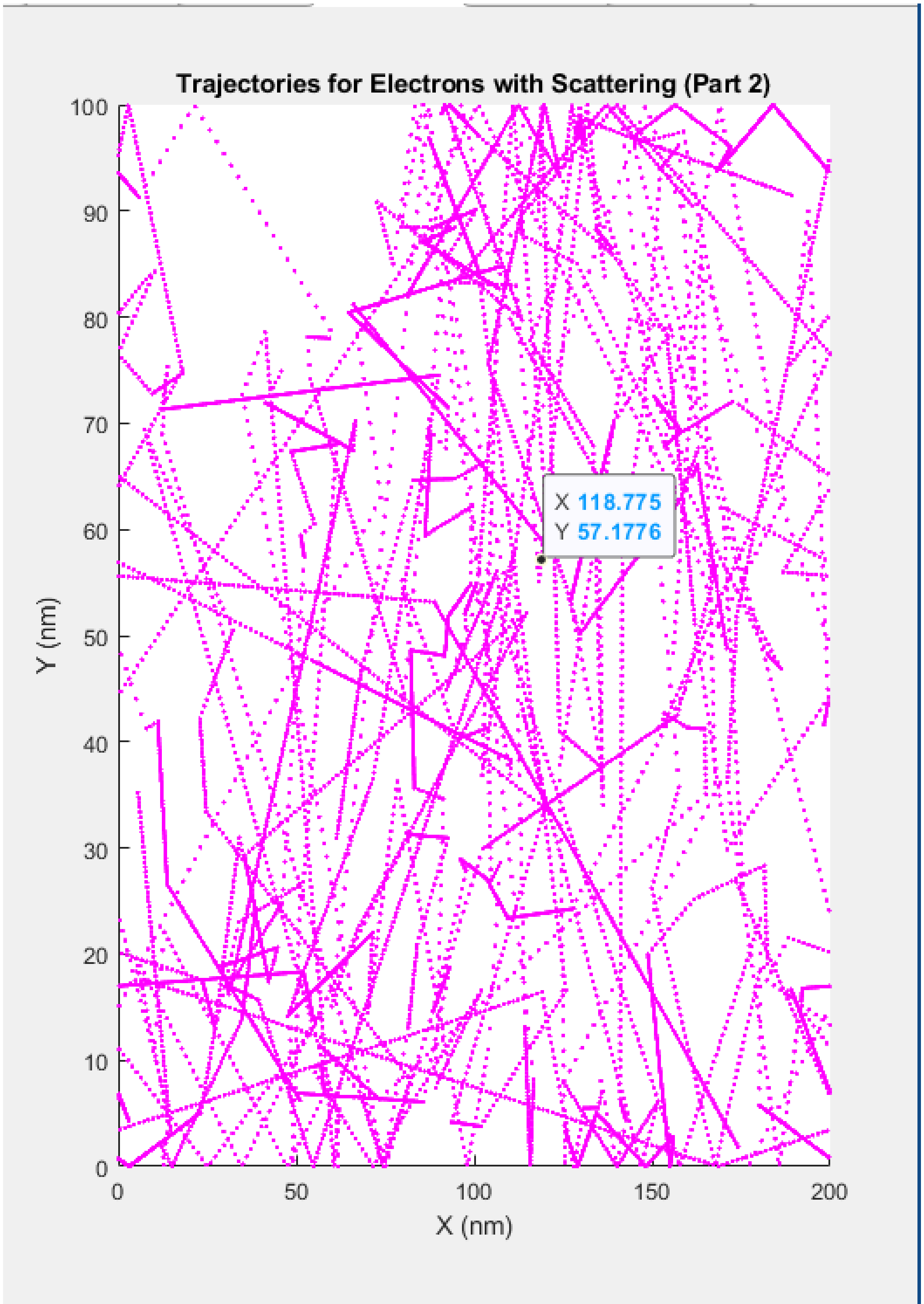

Figure 3: Particle trajectory with scattering

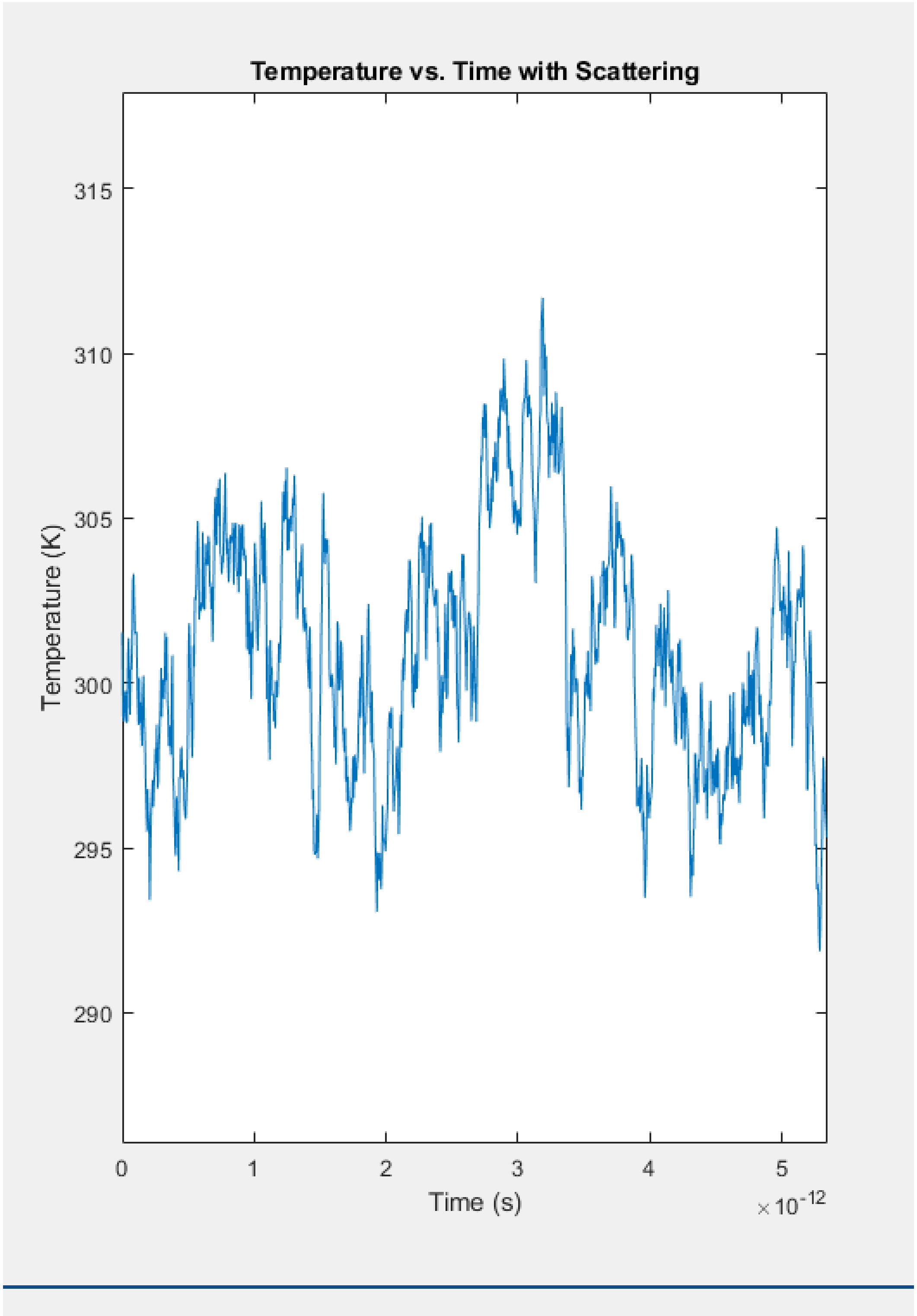

Figure 4: Average temperature vs time with scattering

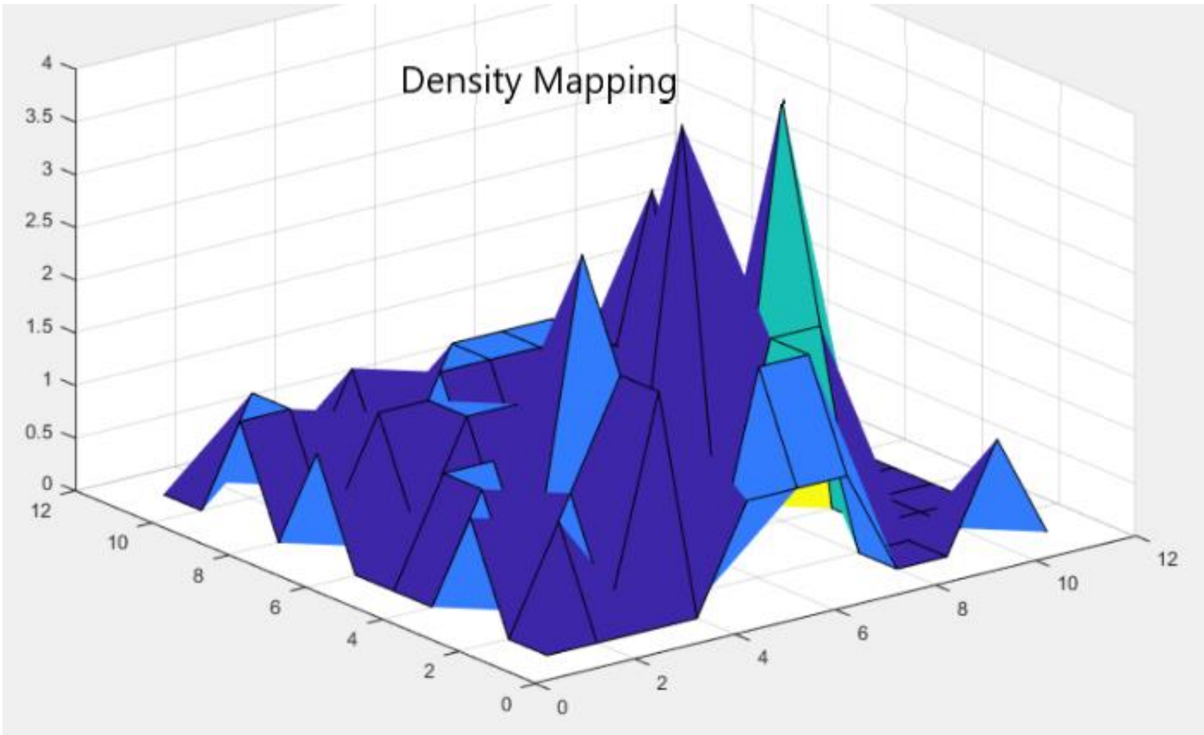

Figure 5: Density mapping

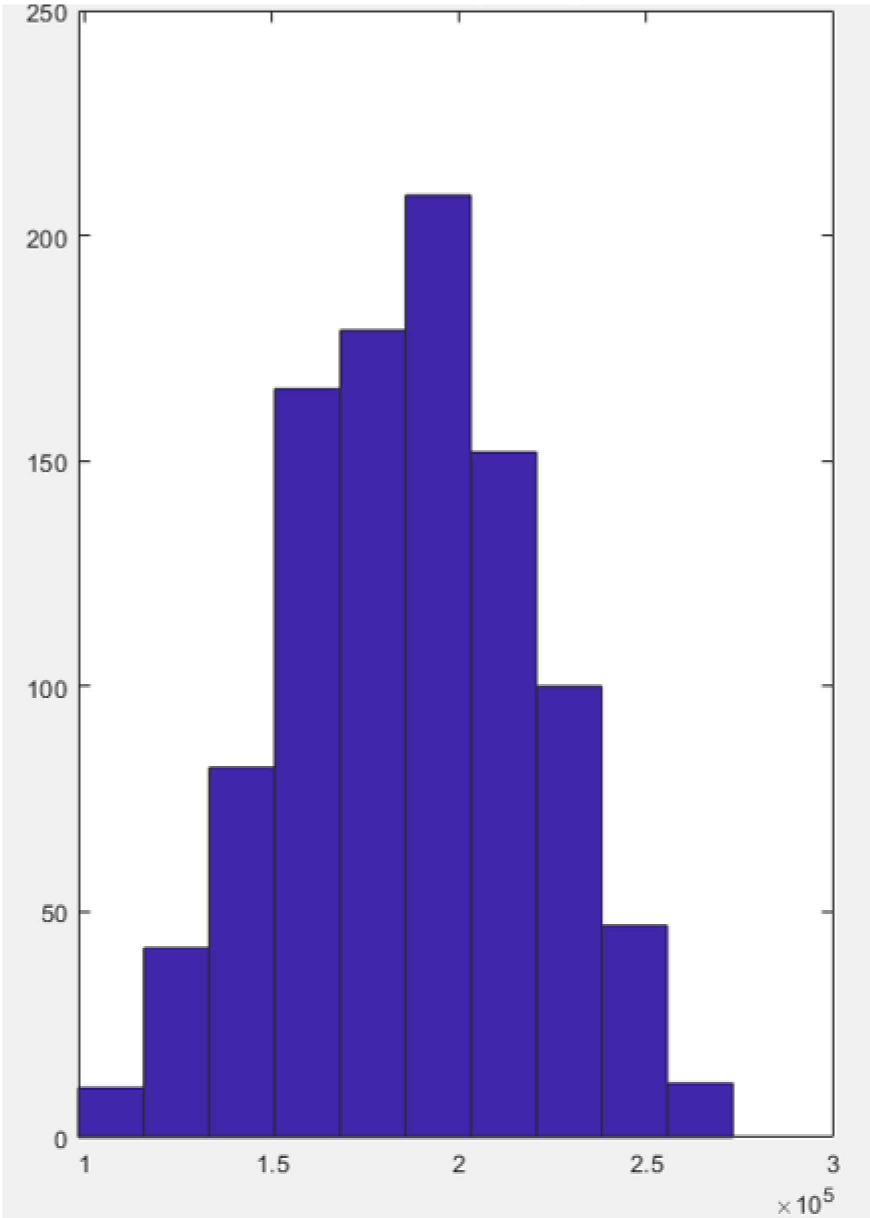

Figure 6: Histogram plot of particle velocity with a 1000 particles

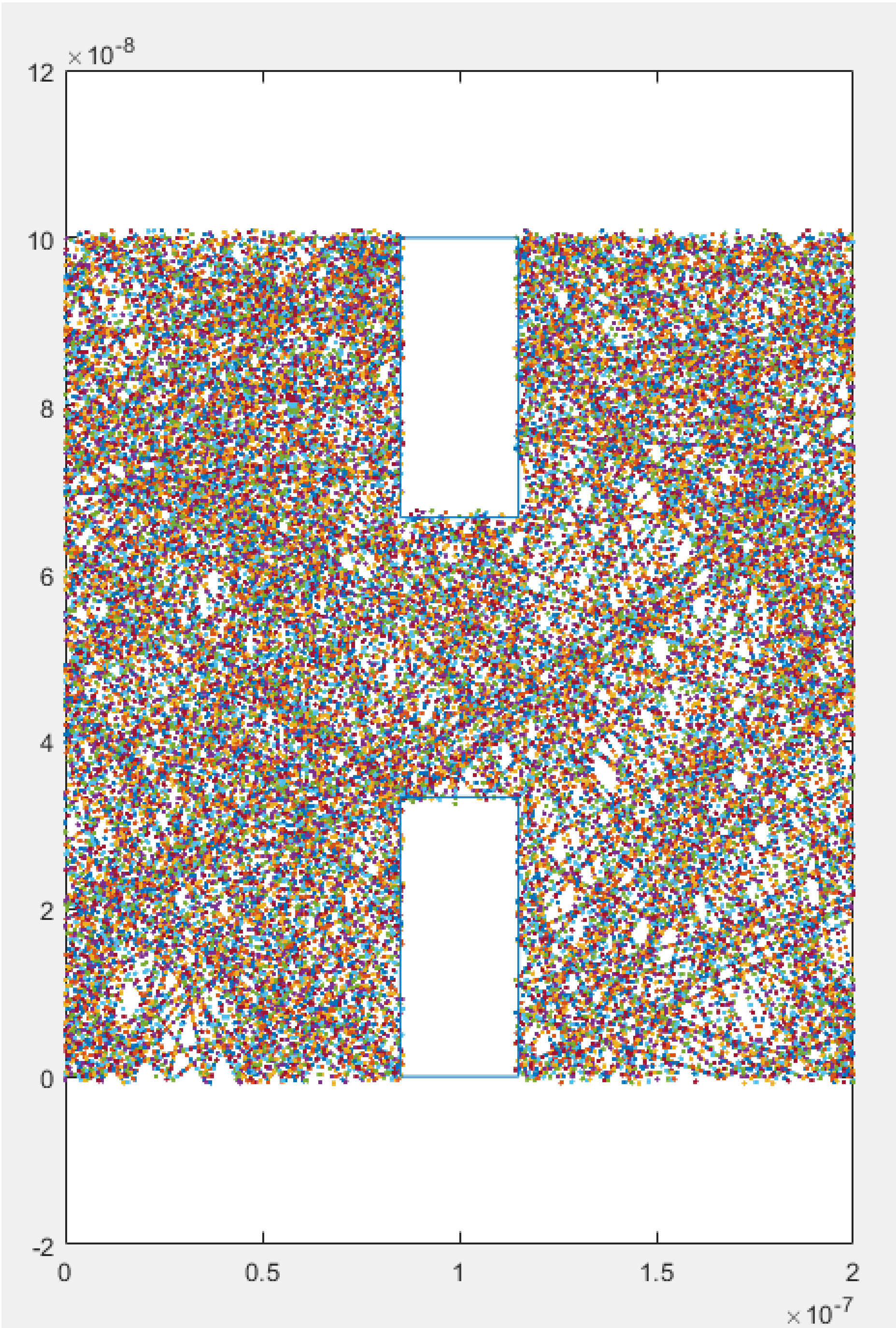

Figure 7: Particle trajectory with bottlenecks added

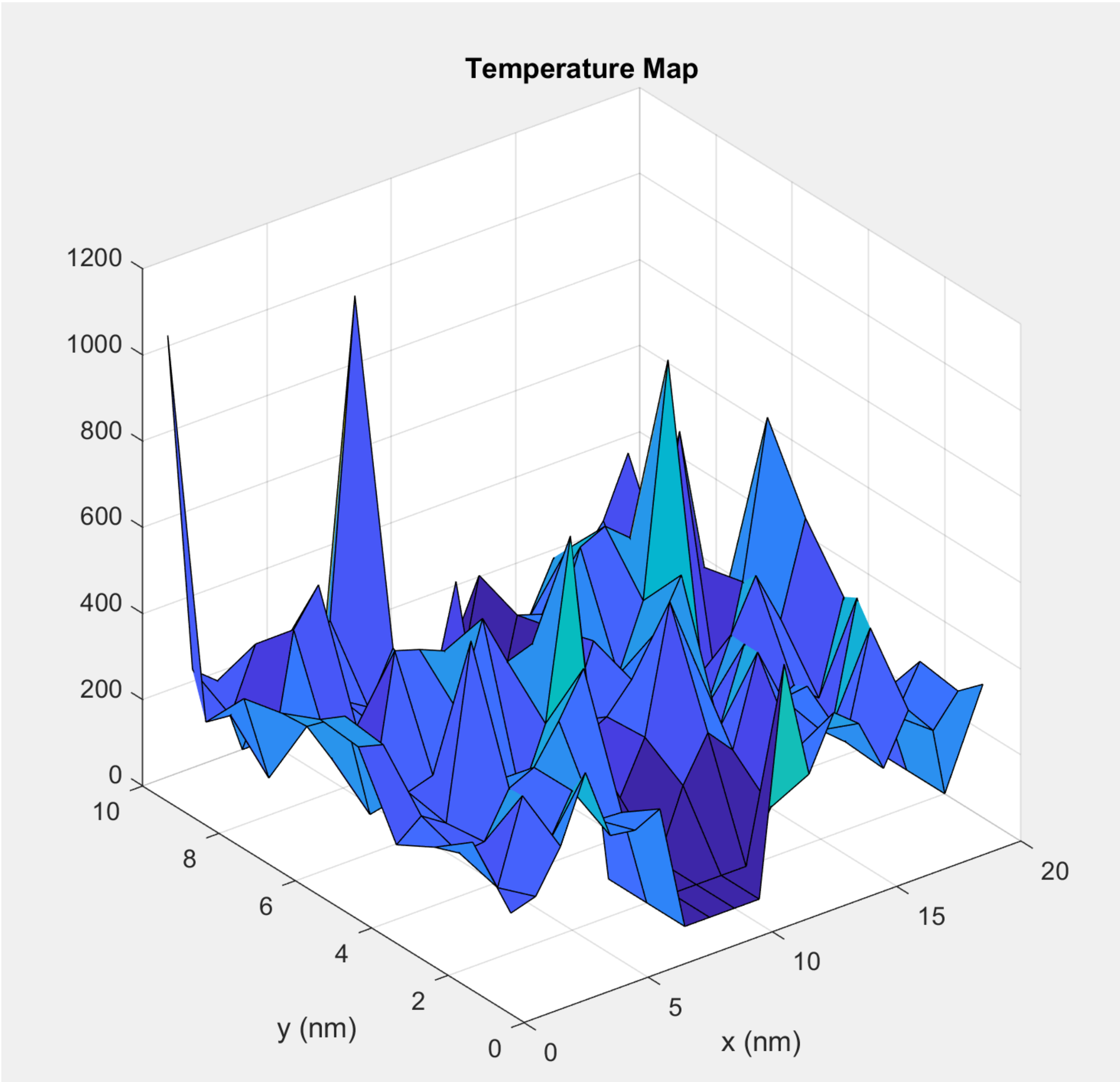

Figure 8: Temperature Mapping

Appendix:

```
clc
clear
clearvars
set(0,'DefaultFigureWindowStyle','docked')
% Name:Jarikre Efe Jeffery
% Student Number: 101008461
m0 = 9.10938215e-31;
effective_m = 0.26 * m0;
Temperature = 300;
q  = 1.60217653e-19;
h = 1.054571596e-34;
h = h * 2 * pi;
```

```
boltz_c = 1.3806504e-23;
permittivity = 8.854187817e-12;
u_permittivity = 1.2566370614e-6;
c = 299792458;
gravity = 9.80665;
am = 1.66053892e-27;
Thermal_v = sqrt((boltz_c * 300) / effective_m);

Standard_deviation = Thermal_v/(sqrt(2));
delta_t = 7.5 * 10 ^ -15;
array_1 = zeros(1, 1000);
tmp_arary_1 = (1:1:1000);
show_m = 300;
wid_x = 200 * 10 ^ -9;
len_x = 100 * 10 ^ -9;
size_particles = 50;
tmn = 0.2 * 10 ^ -12;
x_pos = rand(1, size_particles) .* wid_x;
y_pos = rand(1, size_particles) .* len_x;
Theta = rand(1,size_particles)*2*pi;
x_vel = randn(1,size_particles) .*Standard_deviation;
y_vel = randn(1,size_particles) .*Standard_deviation;

inbox1 = true;
while inbox1 == true
    inbox = ((x_pos <= (1.30 * wid_x/2) & (x_pos >= (0.90 * wid_x/2))) &
((y_pos < (len_x/3)) | y_pos >= (2*len_x/3)));
    if (sum(inbox) > 0)
        x_pos(inbox) = rand(1, sum(inbox)) .* wid_x;
        y_pos(inbox) = rand(1, sum(inbox)) .* len_x;
    else
        inbox1 = false;
    end

end

for i = 1:1000

    x_old = x_pos;
    x_pos = x_old + delta_t*x_vel;
    y_old = y_pos;
    y_pos = y_old + delta_t*y_vel;
    j1 = (y_pos >= len_x);
    j2 = (y_pos < 0);
    y_vel(j1) = -y_vel(j1);
    y_vel(j2) = -y_vel(j2);
    j3 = ((x_pos) >= wid_x);
    j4 = ((x_pos) < 0); %
    x_old(j3) = x_old(j3) - wid_x;
    x_pos(j3) = x_pos(j3) - wid_x;
    x_old(j4) = x_old(j4) + wid_x;
    x_pos(j4) = x_pos(j4) + wid_x;
    inbox = ((x_pos <= (1.30 * wid_x/2) & (x_pos >= (0.90 * wid_x/2))) &
((y_pos < (len_x/3)) | y_pos >= (2*len_x/3)));
    middle = ((x_old<= (1.30 * wid_x/2) & (x_old>= (0.90 * wid_x/2))) &
((y_old > (len_x/3)) & y_old <= (2*len_x/3)));
```

```
    plot (x_pos,y_pos,'r.');
    pause(0.05)
    hold on

    line([0.90*wid_x/2 0.90*wid_x/2], [len_x 2*len_x/3]);
    line([1.30*wid_x/2 1.30*wid_x/2], [len_x 2*len_x/3]);
    line([0.90*wid_x/2 1.30*wid_x/2], [len_x len_x]);
    line([0.90*wid_x/2 1.30*wid_x/2], [2*len_x/3 2*len_x/3]);

    line([0.90*wid_x/2 0.90*wid_x/2], [0 len_x/3]);
    line([1.30*wid_x/2 1.30*wid_x/2], [0 len_x/3]);
    line([0.90*wid_x/2 1.30*wid_x/2], [0 0]);
    line([0.90*wid_x/2 1.30*wid_x/2], [len_x/3 len_x/3]);

    vrms = sqrt ((x_vel.^2)+(y_vel.^2));
    x_vel(inbox&(~middle)) = -x_vel(inbox&(~middle));
    y_vel(inbox&middle) = -y_vel(inbox&middle);
    meanfreetime = (Thermal_v * q)/200;
    meanfreepath = mean(vrms) * meanfreetime;
    vrms = sqrt ((x_vel.^2)+(y_vel.^2));
    show_m = (sqrt (2) * (mean(vrms)^2) * effective_m )/boltz_c;
    array_1 (1,i)=  show_m;

    [x_mesh, y_mesh] = meshgrid(0:(wid_x/10):wid_x, 0:(len_x/10):len_x);
    electron_mat = zeros(11, 11);
    temperture_mat = zeros(11, 11);
    numelec_t = 0;
    t_vel = 0;

    for j = 1:10
        efx_min = x_mesh(1, j);
        efx_max = x_mesh(1,j+1);
        for k = 1:10
            efy_min = y_mesh(k, 1);
            efy_max = y_mesh(k+1, 1);
            for m = 1:size_particles
                if((x_pos(m) > efx_min) && (x_pos(m) < efx_max) && ((y_pos(m)
> efy_min) && y_pos(m) < efy_max))
                    numelec_t = numelec_t + 1;
                    electron_mat(j, k) = electron_mat(j, j) + 1;
                    t_vel = t_vel + sqrt((x_vel(m) .^ 2) + (x_vel(m) .^ 2));
                    temperture_mat(j, k) = ((sqrt(2)*(t_vel/numelec_t) ^ 2) *
effective_m) / boltz_c;
                end
            end
            t_vel = 0;
            numelec_t = 0;
        end
    end
end

fprintf("The Mean Free Time is = %f\n", meanfreetime);
fprintf("The Mean Free Path is = %f\n", meanfreepath);
figure (1)
title(["Average temperature value = " num2str(show_m)]);
xlabel("particle on x axis");
ylabel("particle on y axis");
```

```
figure (2)
plot(tmp_arary_1, array_1);
title('Temperature across Time');
ylabel('Temperature');
xlabel('Time');
hold on
figure(3); histogram(vrms, 15);
title('Histogram of Thermal Velocities');
xlabel("Xlable");
ylabel("Ylable");
figure(4); surf(electron_mat);
title('Density Mapping');
xlabel("Xlable");
ylabel("Ylable");
figure(5); surf(temperture_mat);
title('Temperature Mapping');
xlabel("Xlable");
ylabel("Ylable");

clc
clear
clearvars
set(0,'DefaultFigureWindowStyle','docked')
particles = 1000;

global C X Y
C.q_0 = 1.60217653e-19;
C.hb = 1.054571596e-34;
C.h = C.hb * 2 * pi;
C.m_0 = 9.10938215e-31;
C.kb = 1.3806504e-23;
C.eps_0 = 8.854187817e-12;
C.mu_0 = 1.2566370614e-6;
C.c = 299792458;
C.g = 9.80665;
C.m_n = 0.26*C.m_0;

region_x = 200e-9;
region_y = 100e-9;
step_size = 1e-9;
timestep = 1000;
T = 300;
v_th = sqrt(2*C.kb*T/C.m_n);
v_change = step_size/v_th;
MT_C = 0.2e-12;

X = rand(2,particles);
Y = rand(1,particles);
X_position(1,:) = X(1,:)*region_x;
Y_position(1,:) = Y(1,:)*region_y;

check_X_left = X_position > 0.8e-7;
check_X_right = X_position < 1.2e-7;
check_X = check_X_left & check_X_right;
check_top = Y_position > 0.6e-7;
check_bottom = Y_position < 0.4e-7;
box_top = check_top & check_X;
```

```
box_bottom = check_bottom & check_X;
IN_A_BOX = box_top | box_bottom;

while(sum(IN_A_BOX) > 0)

    temp_x = rand(1,sum(IN_A_BOX));
    temp_y = rand(1,sum(IN_A_BOX));
    X_position(IN_A_BOX) = temp_x*region_x;
    Y_position(IN_A_BOX) = temp_y*region_y;

    check_X_left = X_position > 0.8e-7;
    check_X_right = X_position < 1.2e-7;
    check_X = check_X_left & check_X_right;
    check_top = Y_position > 0.6e-7;
    check_bottom = Y_position < 0.4e-7;
    box_top = check_top & check_X;
    box_bottom = check_bottom & check_X;
    IN_A_BOX = box_top | box_bottom;
end

angle(1,:) = X(2,:)*2*pi;
sigma = sqrt(C.kb*T/C.m_n)/4;
max_boltz_dist = makedist('Normal',v_th,sigma);
velocity = random(max_boltz_dist,1,particles);
figure(6)
hist(velocity)
title('Particle Velocity Histogram')
X_velocity = v_change*velocity(1,:).*cos(angle(1,:));
Y_velocity = v_change*velocity(1,:).*sin(angle(1,:));
PSCAT = 1 - exp(-v_change/MT_C);
mfp_vec = zeros(1,particles);
clc
clear
clearvars
set(0,'DefaultFigureWindowStyle','docked')

K = 1.3806e-23;
m = 0.26*9.1093e-31;

x = 200e-9*rand(1000,1);
y = 100e-9*rand(1000,1);

yboundSpecular = true;
xboundSpecular = true;
boxSpecular = true;

inbox1 = x > 80e-9 & x < 120e-9 & y > 60e-9;
inbox2 = x > 80e-9 & x < 120e-9 & y < 40e-9;
x(inbox1) = x(inbox1) + ((rand() > 0.5)*2 - 1)*40e-9*rand(size(x(inbox1)));
x(inbox2) = x(inbox2) + ((rand() > 0.5)*2 - 1)*40e-9*rand(size(x(inbox2)));
y(inbox1) = y(inbox1) - 0.2*rand(size(y(inbox1)));
y(inbox2) = y(inbox2) + 0.2*rand(size(y(inbox2)));

T = 300;
vth = sqrt(2*K*T/m);
std = sqrt(K*T/m);
Vx = normrnd(0,std,[1000,1]);
```

```
Vy = normrnd(0,std,[1000,1]);
V = sqrt(Vx.^2 + Vy.^2);
Tplot = zeros(1000,1);
figure(1)
histogram(V);
dt = 0.5e-14;

xold = x;
yold = y;
for i =1:400
    %Defines the boundaries of the simulation as well as the boxes
    yboundTop = y > 100e-9;
    yboundBottom = y < 0;
    inbox1 = x >= 80e-9 & x <= 120e-9 & y >= 60e-9;
    inbox2 = x >= 80e-9 & x <= 120e-9 & y <= 40e-9;
    xboundRight = x > 200e-9;
    xboundLeft = x < 0;

    if xboundSpecular
        Vx(xboundRight | xboundLeft) = - Vx(xboundRight | xboundLeft);
    else
        theta = pi*rand();
        Vx(xboundRight | xboundLeft) = V(xboundRight |
xboundLeft)*cos(theta);
        Vy(xboundRight | xboundLeft) = V(xboundRight |
xboundLeft)*sin(theta);
    end

    %Reflection off of y boundary
    if yboundSpecular
        Vy(yboundTop | yboundBottom) = -Vy(yboundTop | yboundBottom);
    else
        theta = pi*rand();
        Vy(yboundTop | yboundBottom) = V(yboundTop |
yboundBottom)*cos(theta);
        Vx(yboundTop | yboundBottom) = V(yboundTop |
yboundBottom)*sin(theta);
    end

    if boxSpecular
        Vx(inbox1 & yold >= 60e-9) = -Vx(inbox1 & yold >= 60e-9);
        Vx(inbox2 & yold <= 40e-9) = -Vx(inbox2 & yold <= 40e-9);

        Vy(inbox1 & yold <= 60e-9) = -Vy(inbox1 & yold <= 60e-9);
        Vy(inbox2 & yold >= 40e-9) = -Vy(inbox2 & yold >= 40e-9);
    else
        theta = pi*rand();

        Vx(inbox1 & yold >= 60e-9) = V(inbox1 & yold >= 60e-9)*cos(theta);
        Vx(inbox2 & yold <= 40e-9) = V(inbox2 & yold <= 40e-9)*cos(theta);
        Vy(inbox1 & yold >= 60e-9) = V(inbox1 & yold >= 60e-9)*sin(theta);
        Vy(inbox2 & yold <= 40e-9) = V(inbox2 & yold <= 40e-9)*sin(theta);

        Vy(inbox1 & yold <= 60e-9) = V(inbox1 & yold <= 60e-9)*cos(theta);
        Vy(inbox2 & yold >= 40e-9) = V(inbox2 & yold >= 40e-9)*cos(theta);
        Vx(inbox1 & yold <= 60e-9) = V(inbox1 & yold <= 60e-9)*sin(theta);
        Vx(inbox2 & yold >= 40e-9) = V(inbox2 & yold >= 40e-9)*sin(theta);
```

```
    end
    y(yboundTop) = 100e-9;
    y(yboundBottom) = 0;
    x(xboundRight) = 200e-9;
    x(xboundLeft) = 0;
    x(inbox1 & yold >= 60e-9 & x <= 100e-9) = 80e-9;
    x(inbox1 & yold >= 60e-9 & x > 100e-9) = 120e-9;
    x(inbox2 & yold <= 40e-9 & x <= 100e-9) =80e-9;
    x(inbox2 & yold <= 40e-9 & x >= 100e-9) =120e-9;
    y(inbox1 & yold <= 60e-9) = 60e-9;
    y(inbox2 & yold >= 60e-9) = 40e-9;


    xold = x;
    yold = y;
    x = x + Vx*dt;
    y = y + Vy*dt;
    scatter = rand(1000,1) < (1 - exp(-dt/0.2e-12));
    Vx(scatter) = normrnd(0,std,size(Vx(scatter)));
    Vy(scatter) = normrnd(0,std,size(Vy(scatter)));
    xplot = transpose([xold(1:20) x(1:20)]);
    yplot = transpose([yold(1:20) y(1:20)]);
    Tplot(i) = (1/(2*K))*mean(Vx.^2 + Vy.^2)*m;
    xlim([0 200e-9])
    ylim([0 100e-9])
    drawnow
end
;
temp_sum_x = zeros(20,10);
temp_sum_y = zeros(20,10);
temp_num = zeros(20,10);

for i=1:1000
 x1 = floor(x(i)/1e-8);
 y1 = floor(y(i)/1e-8);
 if(x1<=0)
 x1 = 1;
 end
 if(y1<=0)
 y1= 1;
 end
 if(y1>100)
     y1 = 100;
 end
 if(x1>200)
     x1=200;
 end
 temp_sum_y(x1,y1) = temp_sum_y(x1,y1) + Vy(i).^2;
 temp_sum_x(x1,y1) = temp_sum_x(x1,y1) + Vx(i).^2;
 temp_num(x1,y1) = temp_num(x1,y1) + 1;

end

temp = (temp_sum_x + temp_sum_y).*m./K./2./temp_num;
temp(isnan(temp)) = 0;
temp = transpose(temp);
```

```
figure(7)
surf(temp)
title('Temperature Map');
xlabel('x (nm)');
ylabel('y (nm)');
```