**Assignment 1 plots:**

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

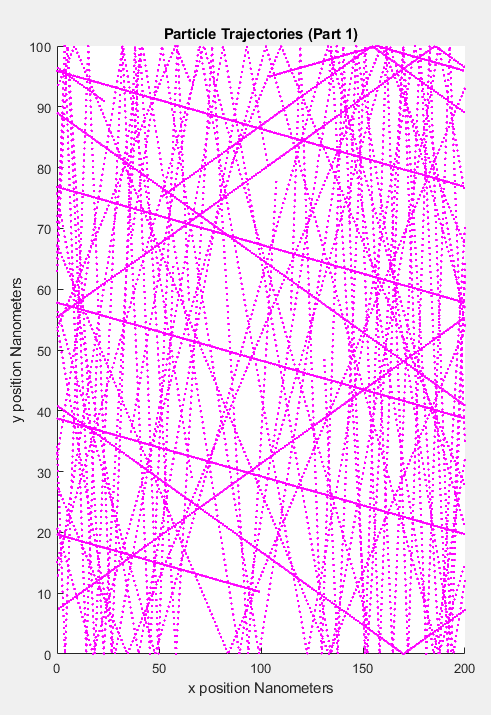


Figure 1: Particle trajectory without bottlenecks implemented

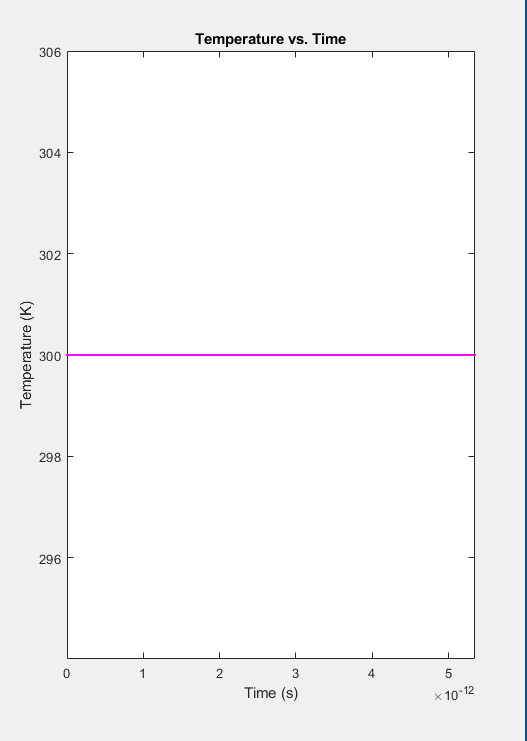


Figure 2: Temperature vs time

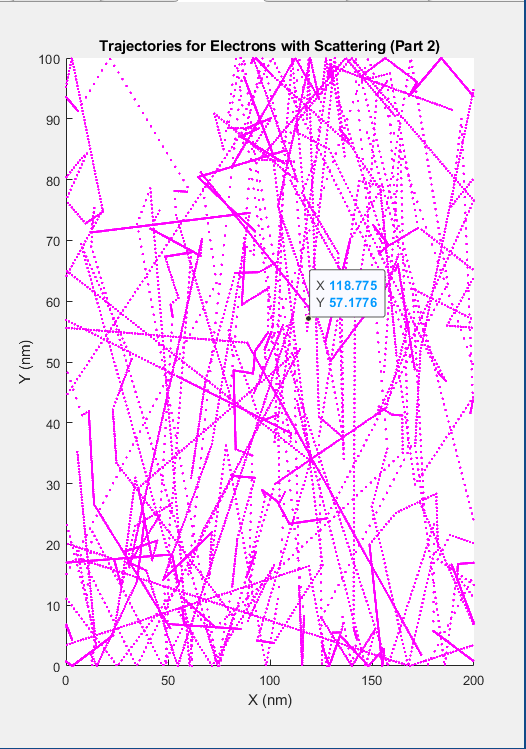


Figure 3: Particle trajectory with scattering

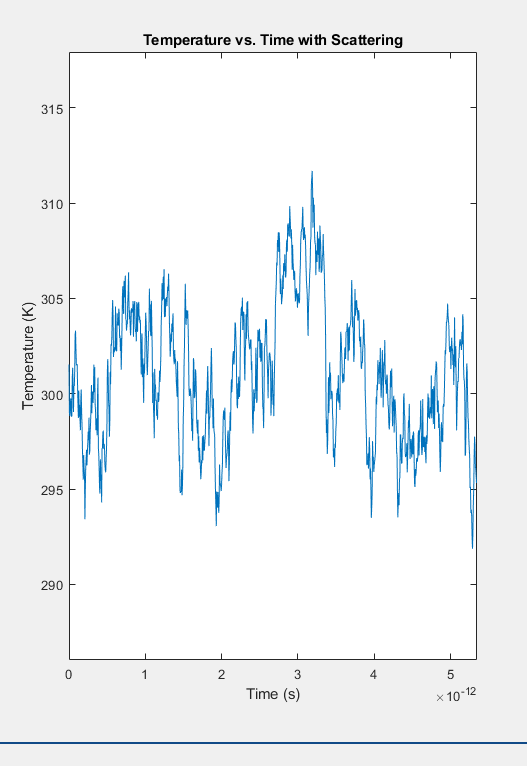


Figure 4: Average temperature vs time with scattering

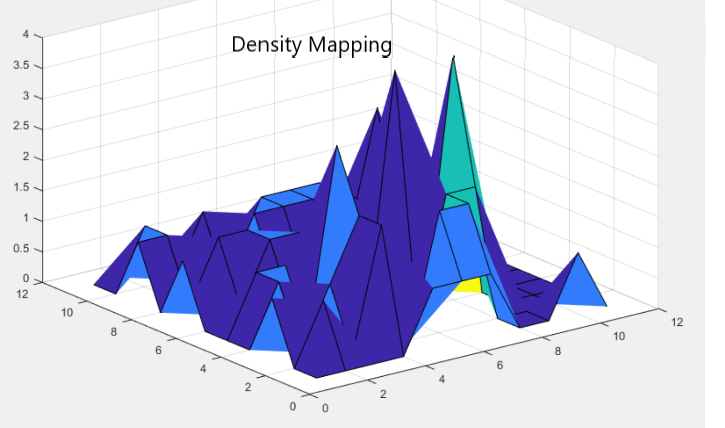


Figure 5: Density mapping

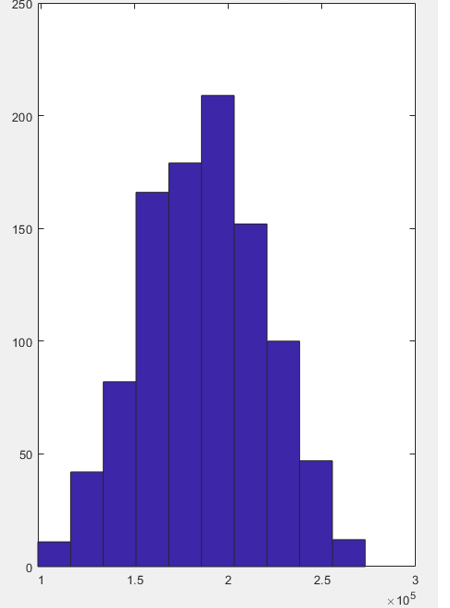


Figure 6: Histogram plot of particle velocity with a 1000 particles

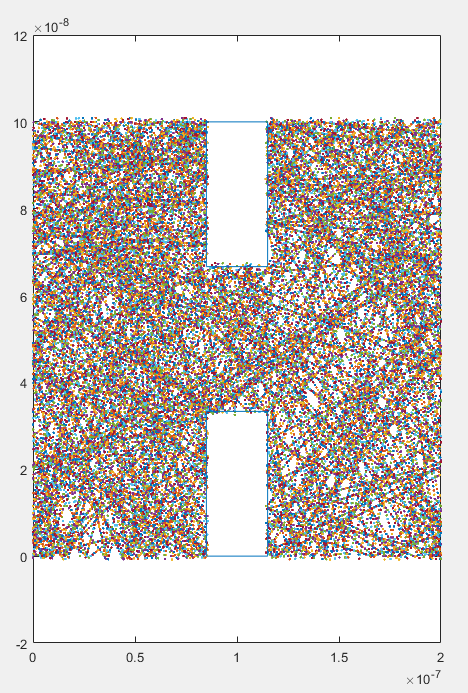


Figure 7: Particle trajectory with bottlenecks added

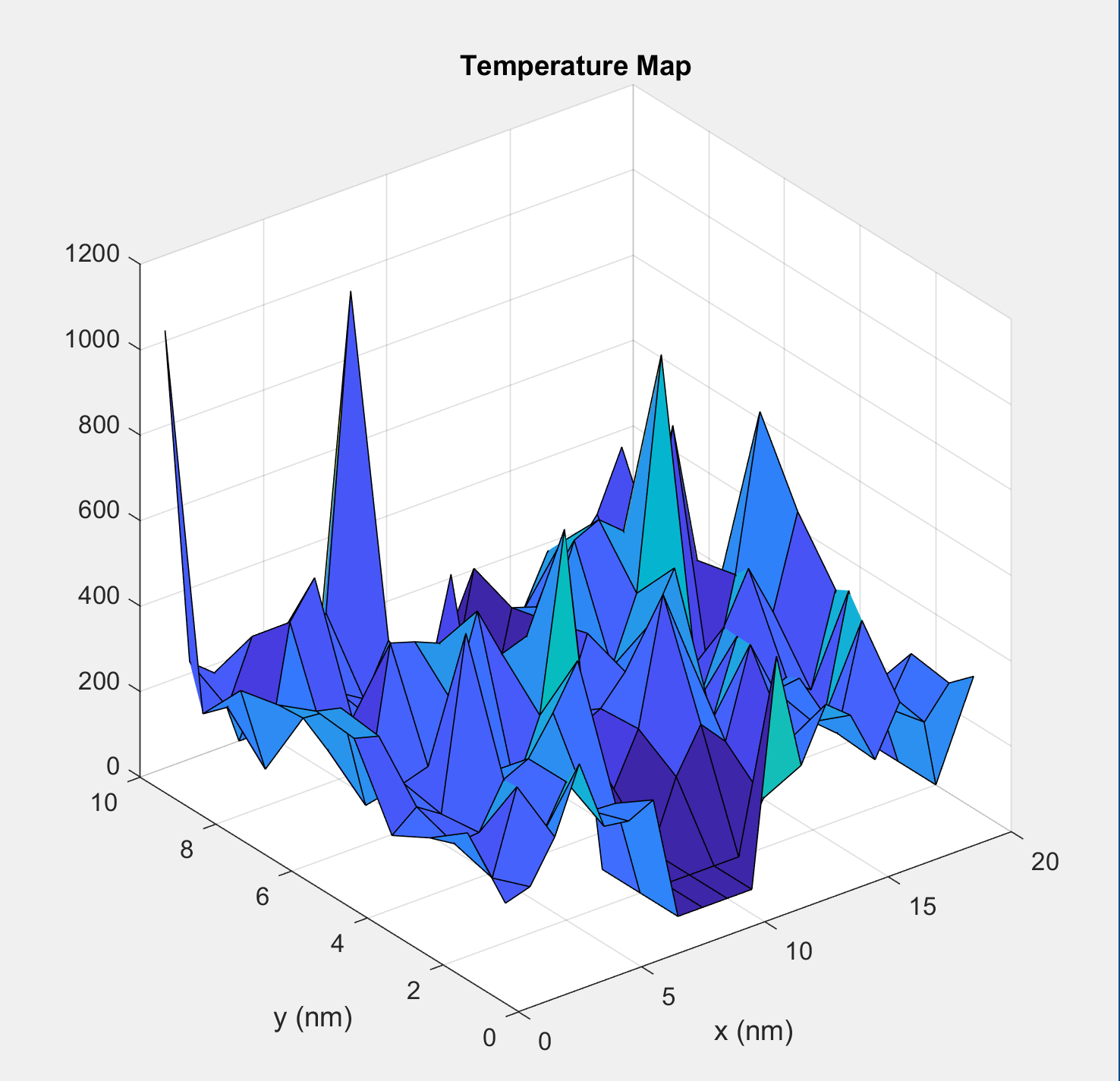


Figure 8: Temperature Mapping

**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)');