

Sample Lab Summary #1

FirstName LastName

PHYS 225 – Spring 2012

Contents

- [Purpose](#)
- [Matlab script](#)
- [Question 5](#)
- [Question 6](#)
- [Question 7](#)
- [Question 8](#)
- [Question 9](#)
- [Question 10](#)
- [Feedback](#)

Purpose

The purpose of this lab is to simulate Rutherford's experiment by 'firing' alpha particles (random points) at an atom (represented by two concentric circles). We will count the number of particles that strike the nucleus vs. the number of particles that strike the atom.

Matlab script

```
close all % close any open figures
format long

R=1.3*10^(-10); % size of atom R ~ 1e-10
r=1.42*10^(-14); % size of nucleus r ~ 1e-14

% counters for the number of points that fall inside the atom or
% nucleus (a circle of radius R or r).
j=0;
k=0;

n=1e3; %number of random points that will be generated

while k < n
    a=R*rand*(-1)^round(10*rand); % we want radom numbers between 0 and R
    b=R*rand*(-1)^round(10*rand);
    % (-1)^round(10*rand) becuae we want both positive and negative
    % values for x and y to cover the four quadrants in the circle

    if a^2 + b^2 <= R^2 %check if points lie inside the circle of radius R
        k = k+1;
        if a^2 + b^2 <= r^2 %check if points lie inside radius r
            j = j+1;
        end
    end
end
end
```

Question 5

```

disp(['Number of particles that strike the atom: ', num2str(k)])

disp(['Number of particles that strike the nucleus: ', num2str(j)])

disp(['Percent of particles that strike the nucleus: ', num2str(j/k*100), '%'])

```

```

Number of particles that strike the atom: 1000
Number of particles that strike the nucleus: 0
Percent of particles that strike the nucleus: 0%

```

Question 6

```

disp(['Ratio of areas: pi*r^2 / pi*R^2 = ', num2str(r^2/R^2)])

```

```

Ratio of areas: pi*r^2 / pi*R^2 = 1.1931e-08

```

Question 7

Code that generates table with data from several runs:

```

expected=9.745e-9; % from in-class calculations
nshot = [100; 1000; 5000; 10000; 62200]; % number of points 'shot'
pointsin = [0.5; 9.5; 45.3; 93; 597]; % points that fell inside b
fracin = pointsin./(nshot*1e6); % fraction of points that fell inside b
fracerror = 1-fracin/expected; % fractional error
data = horzcat(nshot, pointsin, fracin, fracerror); % all data collected for table
hFig1 = figure('Position',[0 0 420 150]);
hTable = uitable(hFig1, 'Data',data,... % make table with data
    'ColumnName',{'n (in million)' 'Avg. points inside' 'Avg. fraction inside' 'F
    'Position',[0 0 420 150]);

```

	n (in million)	Avg. points inside	Avg. fraction inside	Fractional error
1	100	0.5000	5.0000e-09	0.4869
2	1000	9.5000	9.5000e-09	0.0251
3	5000	45.3000	9.0600e-09	0.0703
4	10000	93	9.3000e-09	0.0457
5	62200	597	9.5981e-09	0.0151

Question 8

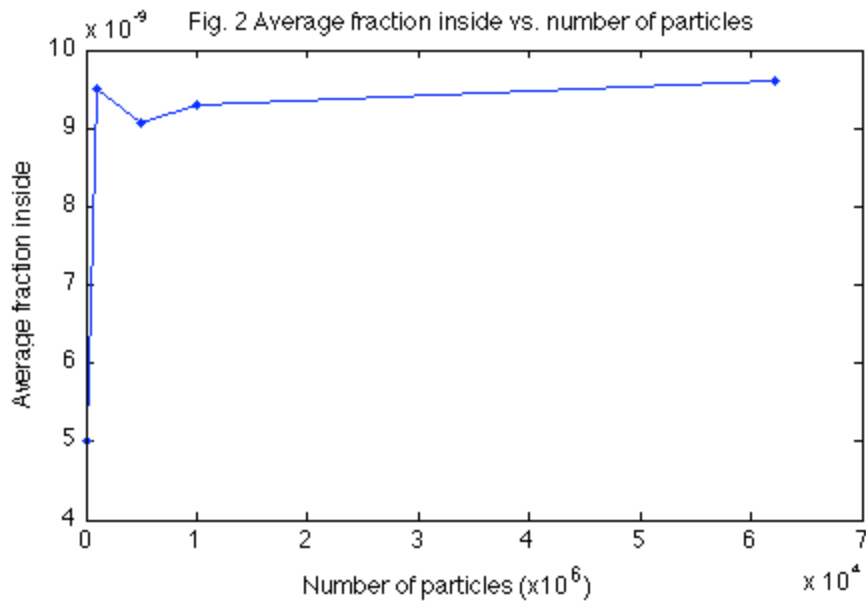
Code for plotting average fraction inside vs. number of particles:

```

hFig2 = figure('Position',[1 1 500 300]);
plot(nshot,fracin, '.')
axis([0 70000 4e-9 10e-9])
hold on
plot(nshot,fracin)
title('Fig. 2 Average fraction inside vs. number of particles');
xlabel('Number of particles (x10^6)')

```

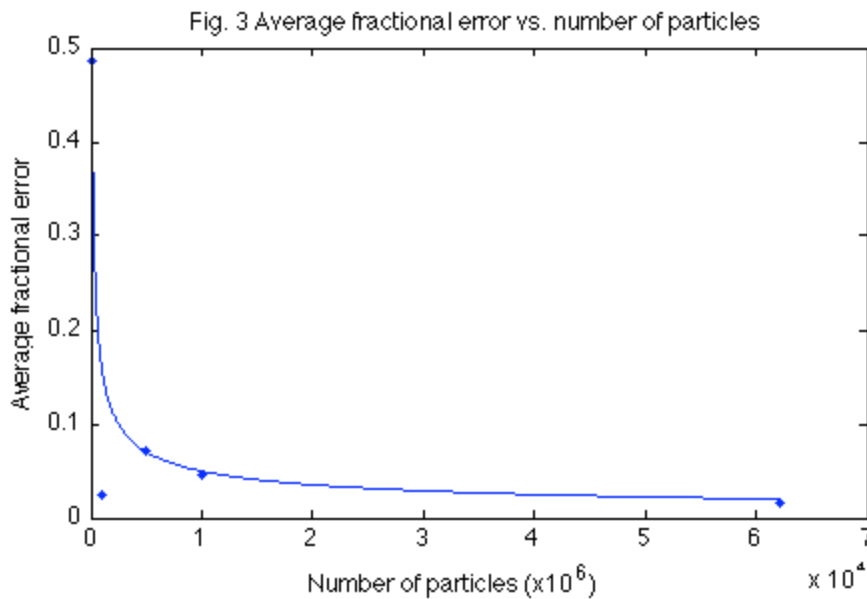
```
ylabel('Average fraction inside')
hold off
```



Question 9

Code for plotting average fractional error vs. number of particles:

```
hFig3 = figure('Position',[1 1 500 300]);
plot(nshot,fracerror, '.')
axis([0 70000 0 0.5])
hold on
nsqrt = [0:10:62200];
afe = 5./sqrt(nsqrt);
plot(nsqrt,afe)
title('Fig. 3 Average fractional error vs. number of particles');
xlabel('Number of particles ( $\times 10^6$ )')
ylabel('Average fractional error')
clear all
```



Question 10

As displayed in Figure 3, the average fractional error decreases rapidly with increasing number of particles. Notably, this is very close to an inverse square root relationship. Therefore, I will postulate that $AFE \sim 1/\sqrt{x}$. According to this model, increasing the number of random points by a factor of x causes the AFE to decrease by a factor of $1/\sqrt{x}$. With a significant amount of additional data, it would be possible to test and investigate this relationship.

Feedback

a) What did you learn from this lab?

In this lab, I successfully used the Monte Carlo method in the Matlab computing environment to develop a simulation of the Rutherford experiment. When a sufficient number of points were sampled, the results of the simulation matched expectations very closely.

b) What questions remain unclear?

Is the rate of convergence the same for all simulations that use Monte Carlo methods?

c) Which course objectives (if any) does this lab fulfill? How are these objectives fulfilled?

This lab fulfills the third course objective because it focuses on modeling physics problems on the computer.

d) How could this lab be improved?

This lab could be improved by adding a more detailed explanation of what is meant by the rate of convergence.

e) How much time did you spend in this lab outside of class?

In addition to the two class periods dedicated to this lab, I spent approximately eight hours collecting additional data, analyzing the data, and writing the lab report.