Rutherford Scattering Assessment

Lukasz R Tomaszewski

June 14, 2023

1 Answers to Questions

(a)

The computational problem that this script/ experiment has is in regards to the alpha particles starting location, the x term must be positive while the y terms has to always be negative, if not the plot depicted in fig. 1 would not display the incident lines bend away from the gold nucleus, this can be tested and proven by manipulating the [x,y] starting co-ordinates and observe how the output plot pushes the incident lines to the right hand side of the plot thus not representing the Rutherford Scattering experiment properly. In Physics terms, the amount of gold nuclei that is contained within the gold plate could alter the diffraction of the alpha particles path as it penetrates the material and diffracts off.

(b)

Even though the simulation in itself is an accurate representation of the Rutherford scattering experiment, thus meaning the values of the parameters can be changed and output the results as if the physical experiment was done separately. In terms of the limitations of the experiment, it cannot contribute the effect of natural forces that are present in the experiment, the simulation has few limitations in regards to this experiment as many parameters such as charge of the gold and alpha particle, speed of light and mass are constants within the simulations and the physical world.

(c)

Within the simulation, parameter have been set which are constants within the physical world, the output plot fig. 1 shows the physical reaction of the alpha particle diffracting when it comes close to the gold nucleus which when the experiment was completed by Rutherford, shows similarities. Only by changing the parameters could it be seen that the experiment was flawed or mathematically correct, as the simulation is based off the mathematics in bedded into the "Spyder" Python 3.6 script.

2 Plot

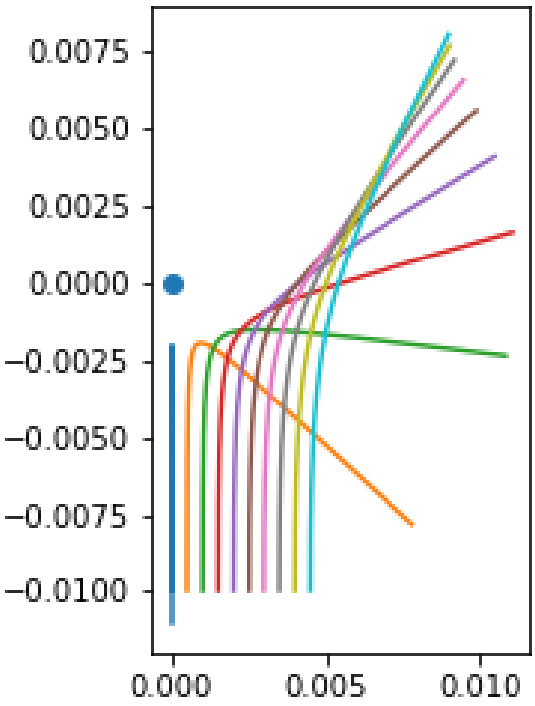

Figure 1: Output plot from results of the script

3 Code

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Mar 18 22:28:49 2018

@author: lrgtomaszewski
"""
import matplotlib.pylab as plt
import numpy as np
# Import packages: Matplotlib to plot the desired graphs.
#                  Numpy to calculate a series of equations.
qe = 2.0 #(AU)
qAu = 79.0 #(AU)
me = 7294.3 #(AU)
ke = 1.0 #(AU)
c = 137.035999 #(AU)
tA = 1e-5 #(AU)
# Presetting the constant parameters which are goin to be used within
# this script.
# NOTE: All units displayed are in astronomical units. NOT SI UNITS!

```

```
22 print("Enter the x-axis value for the particle:")
23 X_Par = float(input())
24 print("Enter the y-axis value for the particle:")
25 Y_Par = float(input())
26 # Displays message to "enter the x & y coordinates" and allows the
27 # user to input values.
28
29 for X_Par in np.arange(0.00000001,0.005,0.0005):
30     ri = np.array([X_Par,Y_Par])
31     vi = np.array([0.0,0.03*c])
32     rlim = np.sqrt(ri[0]**2 + ri[1]**2)*1.1
33     r_x = []
34     r_y = []
35 # If the x terms in the cartersian co-ordinates fit in the range of
36 # the values ditacted in line 29 then the script will run the
37 # following parameters, it will match both seperate x,y interger values
38 # into the format of [x,y] thus giving the script a starting location
39 # in whcih the alpha particle emits from.
40 # rlim (line 32) is a boundary that is 1.1 times the distance away
41 # from the nucleas to the starting position of the alpha particle.
42
43     while (np.sqrt(ri[0]**2 + ri[1]**2) < rlim):
44         ai = (ke/me) * ((qAu*qe) / (np.sqrt(ri[0]**2 + ri[1]**2)**3)) * ri
45         vi = (vi) + (ai * tA)
46         ri = ri + (vi * tA)
47         r_x.append(ri[0])
48         r_y.append(ri[1])
49 # The following lines of script are equations dictated in the assessment
50 # C4 brief. These lines of script are a loop in which the script will
51 # repeatedly run till the final values is suitable enought, the previous
52 # values will also be plotted with the final value to show the effect on
53 # the alpha particle will it moves closer to the gold nucleas.
54
55     plt.plot(r_x,r_y)
56 ax = plt.gca()
57 ax.set_aspect('equal')
58 plt.scatter([0],[0])
59 plt.savefig('RSA Graph')
60 # These lines of script format the data recieved into a plot.
```