Computer Model of Rutherford Scattering

1 Introduction

In this lab you will write a VPython program that models the motion of an alpha particle colliding with a gold nucleus. This simulates the experiment performed by Geiger and Marsden, under the supervision of Rutherford, which led to the discovery of the atomic nucleus. The main questions we want to examine are:

- How does the impact parameter affect the scattering angle of the alpha particle?
- How do the momenta of the alpha particle and the gold nucleus change during the interaction?
- How does the energy of the system change during the interaction?

The setup: An alpha particle (a helium nucleus, containing 2 protons and 2 neutrons) starts out with kinetic energy of 10 MeV (or 1$\times$10^7 eV, where 1 eV = 1.6$\times$10^{-19} J), and heads in the $+x$ direction toward a gold nucleus (containing 79 protons and 118 neutrons). The particles are initially far apart. You can use the approximation that a proton and a neutron both have the about the same mass, 1.66$\times$10^{-27} kg.

2 Preliminary calculations

Consider first the case of the alpha particle moving head-on towards the gold nucleus – that is, the impact parameter is 0. The motion is then confined to one dimension. Based on this, answer the following questions.

Work out your calculations on a piece of paper.

- What is the initial momentum of the alpha particle, $\vec{p}_{\alpha,i}$? (You may assume its speed is small compared to the speed of light).
- What is the initial momentum of the gold nucleus, $\vec{p}_{\mathrm{gold},i}$?
- What is the approximate final momentum of the alpha particle, long after it interacts with the gold nucleus? (**HINT**: This is an elastic, head on collision between a low mass projectile and a large mass, stationary target.)
- What is the approximate final momentum of the gold nucleus, long after it interacts with the alpha particle, $\vec{p}_{\mathrm{gold},f}$?
- What is the approximate final kinetic energy of the alpha particle, $K_{\alpha,f}$?
- What is the approximate final kinetic energy of the gold nucleus, $K_{\mathrm{gold},f}$?
- Assuming that the movement of the gold nucleus is negligible, calculate how close (r_{closest}) the alpha particle will get to the gold nucleus in this head-on collision. (Think about what principle you need to use to answer this question.)
- The radius of the alpha particle is 2$\times$10^{-15} m, and the radius of the gold nucleus is 8$\times$10^{-15} m. At the point of closest approach, do the two particles touch each other?

We need need to choose an initial location for the alpha particle. In your program, you will start the alpha particle at a distance of 2$\times$10^{-13} m from the gold nucleus. Note that this is about 10 times the distance of closest approach, so we will approximate that the particles are very far away from each other initially.

Next we need to estimate a time step `deltat` for the program. To do this, we can first estimate the time it takes the alpha particle to reach the nucleus.

- What is the initial speed of the alpha particle, v_i?
- If the alpha particle did not slow down, how long would it take to travel all the way to the location of the nucleus (a distance of 2$\times$10^{-13} m)?

For an accurate calculation you need to take many steps during the time the alpha particle is near the gold nucleus. Considering your estimate above, we suggest that you use `deltat = 1e-23` s in your program (you can always reduce it if your model is not accurate enough).

3 Program shell

Locate the program shell ("rutherford_shell.py"). Start "IDLE for VPython" and open the shell file. As usual, the program contains several main parts:

- Constants
- Objects
- Initial values
- Calculation loop

The instructions below will help you determine what statements go into each part of the program.

3.1 Constants

Enter the correct masses of the two particles, and enter the suggested time step from above for delta. Note the impact paramter b is defined here. You alter this value later. Also, you may need to define additional constants in this section of the program.

3.2 Objects

Change the `pos` attribute of the alpha particle so that it starts 2$\times$10^{-13} m to the left of the gold nucleus and has an impact parameter of b.

This section also contains the code that creates trails for the particles and graphing windows. We will come back to this section later when adding graphs for momentum and energy.

3.3 Initial values

Change the alpha particles initial momentum to the value you calculated earlier.

3.4 Calculation loop

Here, you will write the physics statements needed to model the motion of the two particles.

3.4.1 The `while` statement

You want the program to stop running when the alpha particle gets about as far from the gold nucleus in the final state as it was in the initial state. What should be the final value of `t`? Change the `while` statement so that the program runs while `t` is less than this final value.

3.4.2 Force

You first need to calculate the vector forces acting on the alpha particle and the gold nucleus. To help you plan out the calculation, answer the following questions on a piece of paper.

- What type of interaction is this?
- What constants will you need in order to calculate this force? Define any new constants in the "Constants" section of the program, then use these defined symbols in your force calculation instead of raw numbers.
- What is the direction of the force on the alpha particle? How can you make the program determine the direction of the force at each time step? (**HINT**: You did a similar calculation in your program that modeled a spacecraft orbiting the Earth and Moon. You may look back at this old program for help.)
- What is the magnitude of the force? How can you use the magnitude and direction to create a vector force?
- Once you have a vector force on the alpha particle, how is it related to the vector force on the gold nucleus?

3.4.3 Applying the momentum principle

Once you have the forces, use the momentum principle to update the momenta of the alpha particle and the gold nucleus.

3.4.4 Updating position

Once you have the new momenta of each particle, use the new momenta to update the position of each particle.

3.4.5 Updating trails

After updating the positions, add the following statements to create trails for the particles:

```
alpha.trail.append(pos=alpha.pos)
gold.trail.append(pos=gold.pos)
```

- At this point, run the program. Does your model behave in the way you expect?
- Experiment with different values of the impact parameter b, and observe the results. Approximately what range of values give a large amount of deflection? What values give a small amount?

3.4.6 Graphing momentum and energy

We will now add graphs of momentum and energy to better understand the motion of the particles during the collision.

In the "objects" section of the program, there are three "gdisplay" lines. These lines create graphing windows. In the first window, we will plot the x-components of momentum of each particle and the total momentum. After the first "gdisplay" statement, write the following statements:

```
alpha_px_graph = gcurve(color=color.cyan)
gold_px_graph = gcurve(color=color.yellow)
px_total_graph = gcurve(color=color.red)
```

The second graphing window will display plots the y-components of momentum. Add the following statements after the second "gdisplay" statement:

```
alpha_py_graph = gcurve(color=color.cyan)
gold_py_graph = gcurve(color=color.yellow)
py_total_graph = gcurve(color=color.red)
```

The final graphing window will display plots of kinetic energy for each particle, potential energy, and K+U. Add the following statements after the third "gdisplay" statement:

```
Kalpha_graph = gcurve(color=color.cyan)
Kgold_graph = gcurve(color=color.yellow)
Ugraph = gcurve(color=color.green)
KplusUgraph = gcurve(color=color.red
```

Now, at the end of the loop (after the time update statement), add statements that will plot these quantities vs. time. For example, to plot the x-component of the alpha particle's momentum, add the statement:

```
alpha_px_graph.plot(pos=(t, alpha.p.x))
```

To plot the energies, you will first have to add calculations for the kinetic and potential energy. Be sure to calculate the **TOTAL** kinetic energy of the system, not just the kinetic energy of one of the particles. You may use the nonrelativistic approximation for kinetic energy.

NOTE: Having all three graphs plotting at the same time may slow down your program. You can always comment out plot statements for one of the graphs if slow-down is a problem.

Consider the following questions:

- Experiment with different values of the impact parameter and observe the behavior of the graphs. What do you notice about the total x-momentum of the system? The total y-momentum? K+U?
- During the interaction, what happens to the x-momentum of the alpha particle? Of the gold nucleus? Does this make sense?
- What happens to the x-momentum of the alpha particle? Of the gold nucleus? Does this make sense?
- What is the value of the total y-momentum? What happens to the y-momentum of the individual particles? Does this make sense?

- Are kinetic energies of the particles always the same during the interaction? What about the total kinetic energy? What about the potential energy of the system?
- Is each particle's initial initial kinetic energy the same as its final kinetic energy? Is the initial total kinetic energy the same as the final total kinetic energy? (To examine this, you may want to temporarily change one of the graphs of kinetic energy to a graph of the sum of the kinetic energies of the particles.) Does this make sense?

3.4.7 Outside the loop

At the end of the program, add statements that print out the final values of the impact paramter b and the momentum components. Make sure these lines are **UNINDENTED**, so that they are outside the loop, and will only execute once the loop is complete.

For example, to print out the impact paramter, add the statement:

```
print('impact parameter=', b)
```

- What do you notice about the y-components of momentum for the two particles?
- Change the impact parameter to 0. Do the final x-components of momentum for the two particles agree with your results in your preliminary calculations?

Finally, calculate and print the angle of the final alpha particle momentum from the +x direction. To calculate the angle, add the following statement:

```
theta = atan2(alpha.p.y, alpha.p.x) * (180/pi)
```

This finds the arctangent of the ratio of the alpha particle's y-momentum to its x-momentum in radians and converts it to degrees.

- What is the scattering angle when b = 0? Does this make sense?
- What value of b gives a scattering angle of 90 degrees?
- What value of b gives a scattering angle of 10 degrees?