Assignment 1 - Option B

PHYS30053

Advanced Computational Physics & Machine Learning

In this assignment, you will develop Python code to simulate a particle physics experiment. You will assess the validity of your solutions and quantitatively assess them against expectations. Finally, you will write a short report that describes your solution, the choices you made, and key conclusions of the exercise.

1 Problem

1.1 Overview

In this exercise, you will simulate an experiment designed to measure the lifetime of an unstable particle X, as illustrated in Figure 1.1. A beam of X particles is produced with a known average momentum and injected into a vacuum chamber. Each X particle will travel some distance into the chamber before decaying at the "decay vertex", r_{vtx}. The charged daughter particle is then detected by four tracking stations, located at various distances (z_i) along the beam line. Each tracking station measures the position where the track hits the station, r_{hit}.

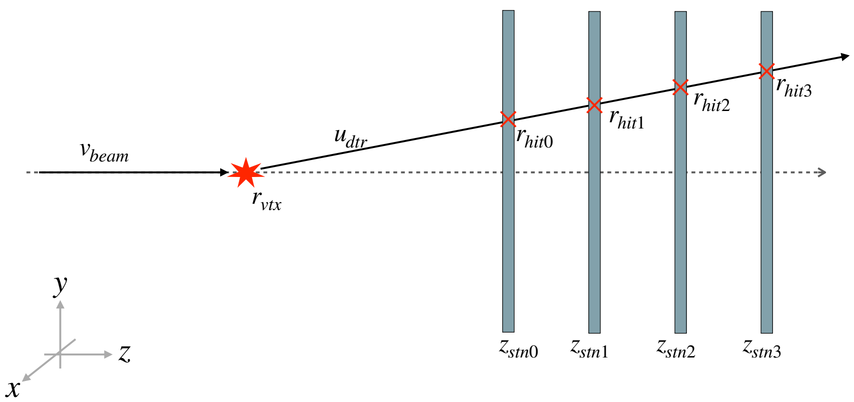

Figure 1.1: Cartoon of the experiment.

The coordinate system used is shown in Figure 1.1, where the beam is produced at the origin and the average beam momentum is in the z-direction.

1.2 Simulation

Your simulation should proceed in several steps, each of which may involve generating random numbers with a particular distribution. The steps required include :

- Generation of a beam particle, with a velocity in the z-direction that is drawn from a normal distribution.

- Decay of the beam particle after a time that is drawn from an exponential distribution. Given the velocity and the decay time, the decay position in the lab frame can then be calculated, assuming the beam is non-relativistic.

- Production of a charged daughter particle in a direction that is isotropic in the rest frame of the beam particle[1]. You can assume the direction in the lab frame is equivalent to the direction in the beam particle rest frame[2]. You can also assume the daughter moves in a straight line, which means we do not need to compute its momentum and the direction vector can be a unit vector.

- Propagation of the daughter particle to the tracking stations. Since the tracking stations are at fixed values of z, the propagation is most easily performed by writing the path as two functions, giving the x and y coordinates as a function of z, ie :
$$x(z) = m_x z + c_x$$
$$y(z) = m_y z + c_y$$
The gradients $m_x = \frac{dx}{dz}$ and $m_y = \frac{dy}{dz}$ can be computed directly from the components of the the daughter particle direction vector, and the intercepts

[1]The isotropic direction distribution is equivalent to generating points on the surface of a unit sphere, with uniform probability over the solid angle, $d\Omega$. In spherical coordinates, this can be achieved by generating two angles, θ and ϕ. However, the distribution is not uniform in both these variables, since the area element is given by $\sin(\theta) d\phi d\theta$

[2]This is a valid approximation, provided the daughter particle has much higher momentum than the beam particle

c_x, c_y can then be found by solving the straight line equations given the (x, y, z) coordinates of the decay vertex.

- Simulating *measured* hits in the tracking detectors. You can assume the tracking stations measure position with finite resolution and normally distributed errors. This can be simulated by adding suitably distributed offsets to the *true* (x, y) hit position of the daughter particle intercept with the detector.

You should develop Python code that will simulate each of these steps. It is strongly recommended that you use 3D vectors throughout the exercise, using numpy arrays, and numpy functions for vector algebra.

Once your code includes all steps, you should simulate a large sample of beam particle events and the corresponding measured hits in each station. From this sample of events, it is possible to compute many different quantities, the distributions of which will inform the design of the experiment. You should present at least :

- The 2D distribution in $(x, y$ of measured hits at each of the four stations
- The distribution of n_{hits}, where n_{hits} for a single beam particle is equal to the number of stations that contain a hit.

Your code should be capable of running with an arbitrary set of parameters, to allow the impact of different experimental choices to be studied. However, you can assume the initial set of parameters below :

- X mean lifetime, $\tau = 2.5\text{m}s$
- Beam particle velocity, $\mu_v = 2000\text{m}s^{-1}$
- Tracking station longitudinal positions, $z_{stn} = 30, 35, 40, 45\text{m}$
- Tracking station transverse size, $\Delta x = 5\text{m}$, $\Delta y = 5\text{m}$
- Tracking hit resolution, $\sigma_{rx,ry} = 0.01\text{m}$

2 Assignment Instructions

2.1 Code

You may develop your code however you wish, but you should submit your final results as a single Jupyter notebook (ie. one .ipynb file). Make sure you check the notebook runs correctly from a fresh kernel before submitting.

Your code must produce all results presented in your report. It should also include all code required to validate your solution, including any quantitative assessment of the output or comparison with experimental data or analytic solutions.

You MAY use AI to generate code for this assignment. However, you must properly attribute any generated code using comments, to make it completely clear what lines of code have been generated, including prompts and information about the AI/generator used.

Some things you will need to think about while developing your solution :

- How to validate your approach, including intermediate steps
- How to present your results appropriately
- How to quantitatively assess your results
- How to factorise your code (eg. what functions to use)
- Whether and how you can make your code re-usable

2.2 Report

Your report should describe how your code solves the problem set, why you made the choices you did, and what you have learnt from the exercise.

Do not re-state the problem. Assume the reader has read this assignment brief before they read your report - you can refer to sections and equations in this brief.

The final section of your report must be titled “Use of Generative AI”. In this section you should describe how you used AI for the assignment. Make sure to include a list of the AI tools used, and a description of your approach. You do not need to repeat prompts that are included in the code comments. If you did not use AI at all, include the section but simply state “AI was not used in this assignment”.

The report should be no longer than 1000 words.

Below are some ideas for what you might want to include in your report :

- How did you validate each of the steps in the simulation ?
- What qualitative conclusions can you draw from your results ?
- Can you draw any conclusions for the experiment design ?