Computing Exercise 3: Using Monte Carlo methods to investigate particle decay

Saskia Jackson

July 20, 2020

INTRODUCTION

The Monte Carlo method is a technique in which randomly generated numbers are used to obtain a numerical analysis. [1] This method is often used in order to simulate random processes or processes with many coupled degrees of freedom, for example, they can be used to determine how galaxies evolve, how cellular structures behave and to model particle physics detectors.

Often pseudo-random numbers are used in Monte Carlo simulations due to the speed at which they can produce random number simulations and their reproducible nature. [1] Pseudo-random numbers approximate random numbers but are not truly random as they are determined by an initial value.

Random number generators often generate numbers with a uniform distribution which sometimes need to be converted into a non-uniform distribution, for example when approximating a function.

This analytical method can be used to simulate many particle physics problems, especially when dealing with decay due to the randomness of this process.

PART ONE: INVESTIGATING ANALYTICAL AND MONTE CARLO METHODS TO APPROXIMATE A SINE CURVE

The first part of the exercise compared two different methods of approximating a sine function.

The first method was an analytical method that involved converting a uniform distribution ($P(x)$) to a non-uniform distribution ($P'(x')$) to approximate a sinusoidal curve. This non-uniform distribution can then be equated to a definite integral ($Q(x)$) from which the result can be determined.

In this case, of approximating a sine curve, $P'(x')$ must be proportional to $sin(x)$, therefore the result can be determined as half the integral of $sin(x)$ where the half appears from the normalization of the distribution.

This result can then be inverted in order to find the required value of x.

$$x'_{req} = Q^{-1}(x_{gen}) = \arccos(1 - 2x_{gen}) \tag{1}$$

Equation 1 then generates a set of random numbers distributed sinusoidally that can be formatted as a histogram to show the sinusoidal curve as shown by figure 1a.

An alternate method to approximate a sine curve is to use an accept-reject method that is very similar to Monte Carlo integration methods. In this method a set of randomly generated coordinates are generated between 0 and π for the x axis and between 0 and 1 for the y axis. Each coordinate is compared to the in built sine function to determine whether the points are greater than (x,sin(x)) in which case they are rejected, otherwise they are accepted and appended to an array. This can then be plotted in a histogram format and compared to a sinusiodal curve as shown in figure 1b.

The graph in figure 1c compares the time taken for each method for a varying number of iterations. The plot shows the analytical method being slightly faster than the accept-reject method. This is due to the fact that the accept-reject method involves more steps than the analytical method.

If the inverse function used in the analytical method is more complex, the accept-reject method may actually be faster.

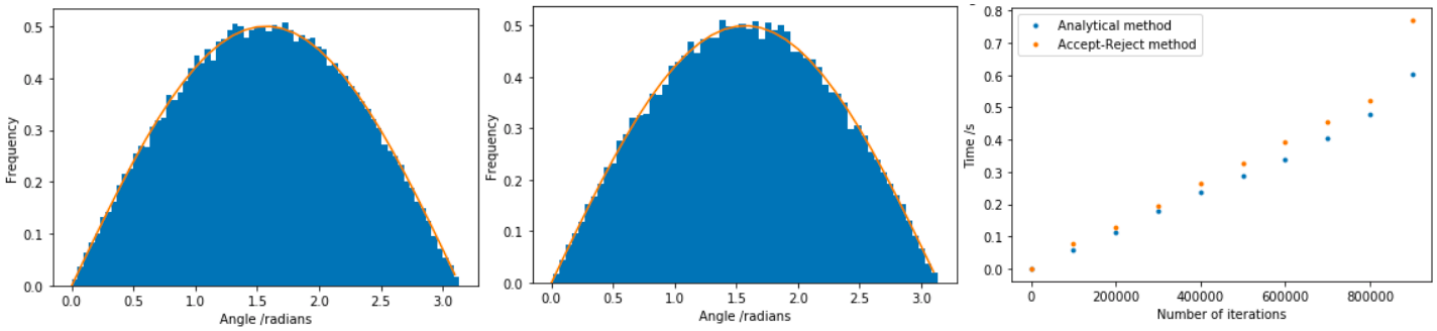

Figure 1: Left: (1a) A histogram showing the analytical method (with 1000000 iterations) of approximating a sine curve compared to the numpy sine function (shown as the orange curve). Middle: (1b) A similar histogram comparing the accept-reject method (with 1000000 iterations) compared to the numpy sine function. Right: (1c) A plot showing how the speed of the code varies with the number of iterations for both methods.

The approximation gets more accurate if more iterations of the method are used, which can be seen in both the graphical representation and in the chi squared value for each method for different amounts of iterations.

Number of iterations	Chi squared value	
	Analytical method	Accept-reject method
1,000	3.3173439438261494	4.75694737983071
1,000,000	0.030430547500084944	0.029931814299838537

TABLE 1: A table showing the chi squared values for the two methods for different number of iterations.

When more iterations are used, the chi squared value is closer to 0, thus indicating a more appropriate approximation. It can also be seen that the accept-reject has a smaller chi squared value, indicating it may be a better approximation than the analytical method, despite the analytical method being slightly faster for all numbers of iterations.

When you increase the number of iterations, the histogram begins to represent the sine curve much more accurately. However even at a large number of iterations (1,000,000 iterations) it can be seen by eye that the analytical method is a more accurate method. This is likely because the analytical method plots 100% of points onto the histogram whereas the accept-reject method only plots 63% of points due to the rest being rejected. This, therefore, introduces some random error to the approximation.

PART TWO: USING MONTE CARLO METHODS TO SIMULATE GAMMA RAYS EMITTED FROM NUCLEAR DECAY

The second part of this exercise involved using the methods investigated previously to simulate the distribution of gamma rays emitted from a beam of unstable nuclei.

The nuclei can decay at any point between the injection point and the decay and the point at which this occurs is random for each nucleus. It is also assumed that the nuclei do not deviate from the beam, thus always have coordinates (0,0,z). The random nature of the decay means it can be simulated using a random distribution and the Monte Carlo method.

The number of particles decayed at any point is determined using equation 2 [2] and then plotted on a histogram to investigate the distribution (as shown in figure 2a).

$$N = N_0 e^{\frac{d}{v}\tau} \tag{2}$$

where N_0 is the number of particles at the beginning of the simulation (set to 1 for simplicity), d is the distance travelled by the particle (for this simulation this is set to be a random distribution), v is the velocity of the nuclei and τ is the lifetime of the nuclei.

The gamma particles are assumed to be emitted isotropically. This can be shown graphically by plotting the position of each gamma ray after decay at a set radius from the decay point (a radius of 1m was used in the 3D plot in figure 2b)

Spherical coordinates (r, θ, ϕ) can be used to describe the position of the gamma rays after decay. θ can be set as a randomly generated distributed but ϕ is a variable in the differential area element equation (left in equation 3) and therefore has to be calculated (to give the equation on the right in equation 3) using the same method as in the analytical method in part 1.

$$da = r^2 \sin(\phi)\, d\phi d\theta \qquad \phi = \arccos(1 - 2(\text{random distribution})) \tag{3}$$

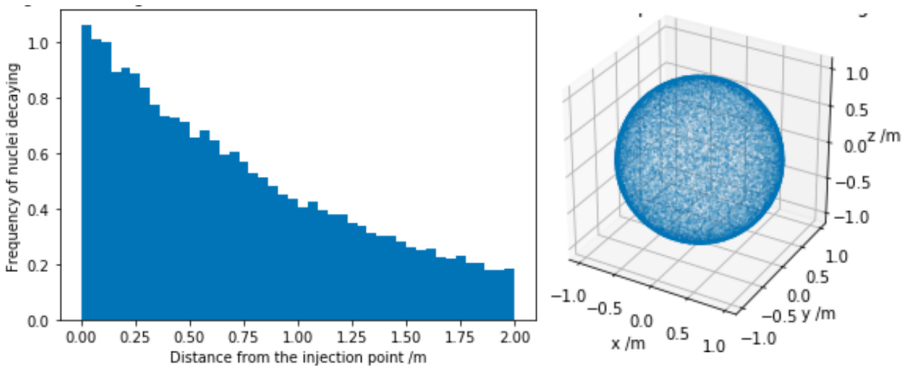

Figure 2: Left: (2a) A histogram to show the distribution of the position of 100,000 unstable nuclei decays between the injection point and the detector. Right: (2b) A 3D plot showing the isotropic nature of the emitted gamma rays.

The distribution of the number of gamma rays that are incident on a given point on the detector can be determined using the randomly generated θ and ϕ angles. These angles can be converted into x and y Cartesian coordinates using equation 4.

$$x = \frac{r\cos(\theta)}{\tan(\phi)} \qquad y = \frac{r\sin(\theta)}{\tan(\phi)} \tag{4}$$

Once it has been checked that the gamma ray hits the detector, the position can be recorded and plotted on a 2D histogram (as shown in figure 3a).

The detector has a set resolution of 10cm and 30cm in the x and y directions respectively. To introduce this resolution into the simulation, smearing of the (x,y) coordinates must be introduced. This is done by applying a randomly distributed Gaussian to the results with the standard deviation of the detector resolution. The corrected simulation of the distribution on the detector is shown in figure 3b.

It can be seen from a comparison of the simulation with and without smearing, that the smearing has a visable effect on the shape of the particle pattern on the detector.

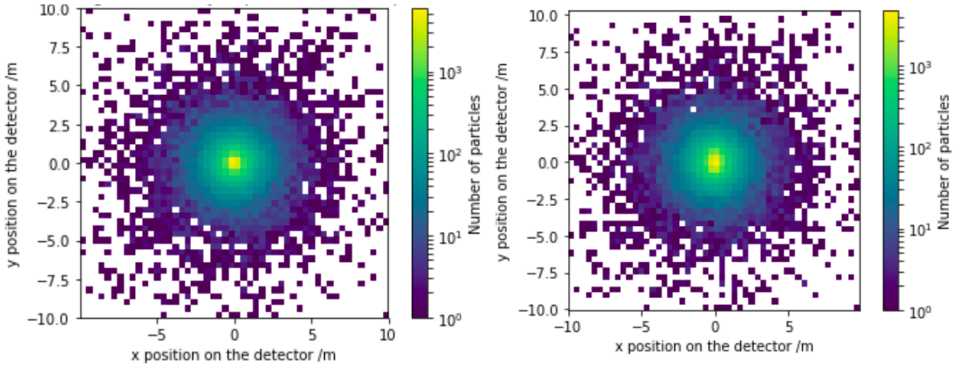

Figure 3: Left: (3a) A 2D histogram to show the distribution of the number of particles at each point on a detector 10x10m in size. Right: (3b) The same graphical representation but with a correction for smearing included for a standard deviation of 1.

PART THREE: INVESTIGATING THE CONFIDENCE LEVEL OBTAINING IN A PARTICLE COLLIDER EXPERIMENT

The rate of particle emitted in particle collider experiments can be predicted from the luminosity (L) multiplied by the cross section (σ).

The number of background events is already known to be 5.7 ± 0.4 therefore a Gaussian distribution centered on this number can be found. A random Poisson distribution can then be found around this distribution for the background and events determined from the predicted number of events. These two Poisson distributions can be summed to find the number of events. If this number of events is greater than 5 then the process is significant and should be recorded.

The percentage of significant events can then be recorded. A 95% confidence level is the required confidence level from which the corresponding cross section can be found.

Using the graph in figure 4, the cross section corresponding to 95% confidence was found to be 0.41nb.

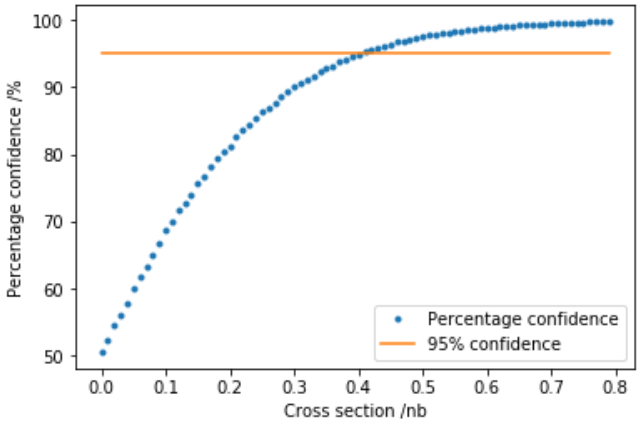

Figure 4: A plot to show how the percentage confidence varies for different cross sections for a luminosity of $12nb^{-1}$.

DISCUSSION

To improve the speed of the program, numpy arrays could be used, as they are homogeneous and can be iterated through much faster than lists. The dynamic nature of python lists means they take longer to interpret as the program has to check what object types are stored at each step.

An additional improvement to the code would be to determine the error associated with the cross section corresponding to 95% confidence by reversing the technique to find the confidence level for a cross section of 0.41nb and investigate the deviation from a percentage confidence of 95%. This would give an indicator of the accuracy of the measurement.

1 References

[1] D.P. Kroese, T. Taimre Z.I. Botev *Handbook of Monte Carlo methods* Wiley Publication (2011)

[2] W.N. Cottingham D.A. Greenwood *An introduction to Nuclear Physics* (2001)