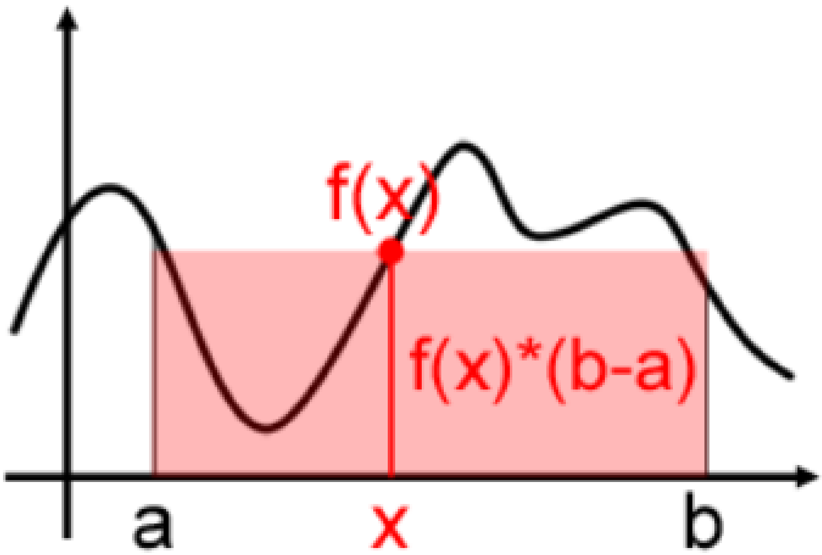
SOP TEMPLATE

Author: John Hu

Date Created: 05/25/22
Date Edited: 05/25/22
Version 1.0

Abstract

In this SOP, the principle of operation of integration through Monte Carlo will be discussed. Certain issues and design considerations will also be explored.



Contents

1 Description of Monte Carlo

Monte Carlo Integration is a technique which can be applied to quantum field theory, dynamical systems of particles, and Finite Element Analysis. [1]

The major types of Monte Carlo integration are uniform sampling, stratified sampling, sequential Monte Carlo, and mean field particle methods. Consider the fundamental theorem of Calculus:

Theorem 1.

$$F(x) = \int_a^b f(x)dx$$

By choosing a random value x constrained by a and b and multiplying by the difference (b-a), this approximates a single rectangle with area of x(b-a). Adding up these rectangles approximates the area under f(x) like a Riemann Sum with a random distribution. Hence, the general formula is

$$I = \frac{b-a}{N}\sum_{i=0}^{N} f(x_i).$$

[2] Uniform sampling samples a uniform probability distribution is one where there is an equal probability for all numbers within . Stratified sampling divides the total space into smaller groups rather than uniform distribution.

2 Error of Monte Carlo

The error of such an integration is evidently important for such a method. This can be quantified using the variance (σ) or alternatively the standard error of the mean which is calculated by $\frac{\sigma}{\sqrt{N}}$ due to the central limit theorem. [3]

3 Numerically Solving with Monte Carlo Integration

Example: Let the equation $f(x) = 5x^2 + 2x$. Find the area under the curve from -2 to 7 using Monte Carlo integration.

In this case, we need to import ***random*** to assign uniform probability distribution. There are better ways to generate random numbers as the algorithm by scipy is deterministic however they are beyond the scope of this tutorial.

```
from scipy import random
import numpy as np
import matplotlib.pyplot as plt
```

Next, we need to define the bounds of integration as well as the sample size

```
a=-2
b=7
N=10000
xrandom=np.zeros(N)
```

Finally, define the target function listed above and iterate through the sample size while multiplying by the formula (b-a)/n.

```
def f(x):
    return 5*x**2+2*x

integral=0.0
for i in range(N):
```

```
    integral+=f(xrandom[i])

answer=(b-a)/float(N)*integral
print(answer)
```

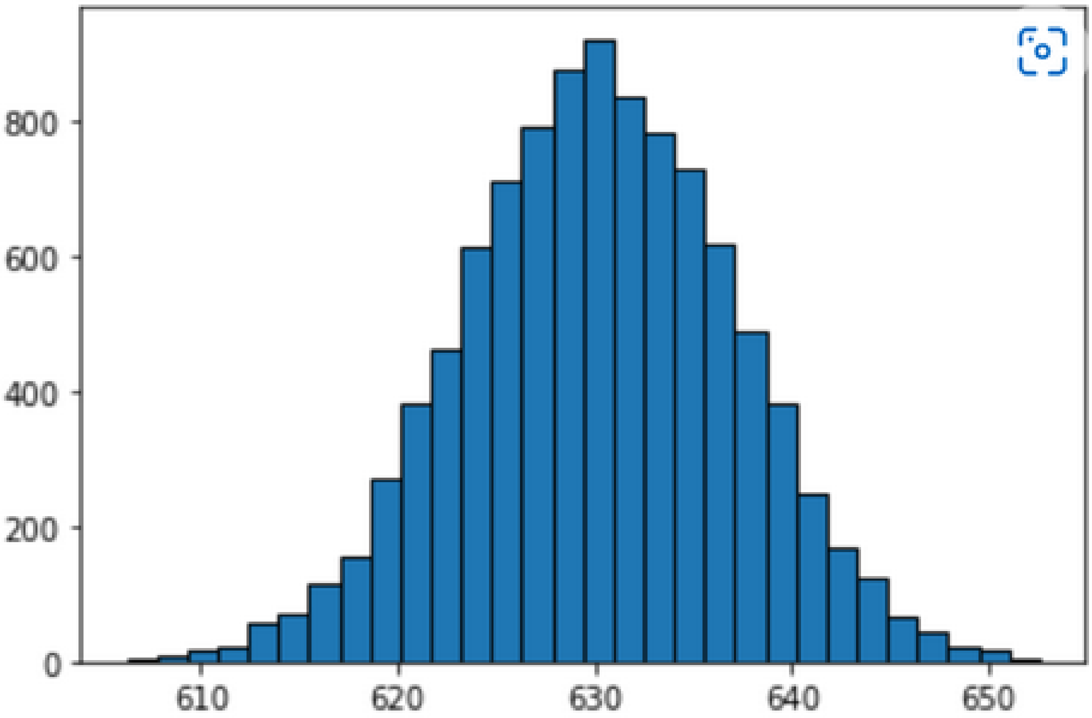


Figure 1: Histogram of simulated Monte Carlo values

REFERENCES

[1] Monte carlo methods in practice.
https://www.scratchapixel.com/lessons/mathematics-physics-for-computer-graphics/monte-carlo-methods-in-practice/monte-carlo-integration.

[2] Monte carlo simuations in physics.
https://www.oulu.fi/tf/montecarlo/lectures/mc_notes1.pdf.

[3] The monte carlo method.
https://compphys.quantumtinkerer.tudelft.nl/proj2-monte-carlo/.