

CS 357 - 07 RNG and Monte Carlo Methods

Boyang Li (boyangl3)

Truly-random and pseudorandom:

- **Truly-random numbers:** Generate random numbers according to some random physical phenomenons. (Example:rolling a fair die and get the number)
- **Pseudorandom:** Generate random numbers based on algorithms, the patterns are predictable.

Properties of random number generators:

- **Random pattern:** passes statistical tests of randomness.
- **Efficiency:** executes rapidly and requires little storage.
- **Long period:** goes as long as possible before repeating.
- **Repeatability:** produces same sequence if started with same initial conditions.
- **Portability:** runs on different kinds of computers and is capable of producing same sequence on each.

Linear congruential generator:

Here is the standard form of linear congruential generator (LCG):

$$x_0 = \text{seed}$$

$$x_{n+1} = (ax_n + c) \mod M$$

The period of an LCG cannot exceed M (the modulus). The period may be less than M depending on the values of a and c . The quality depends on both a and c .

Random variable:

A random variable X can be thought of as a function that maps the outcome of unpredictable (random) processes to numerical quantities. The outcomes of discrete random variables are countable, but the outcomes of continuous random variables are uncountable.

The expected value of a random variable is $\sum x_i p_i$.

Monte Carlo method: Monte Carlo methods are algorithms that rely on repeated random sampling to approximate a desired quantity. Monte Carlo methods are typically used in modeling the following types of problems:

- Nondeterministic processes,
- Complicated deterministic systems and deterministic problems with high dimensionality (e.g. integration of non-trivial functions).

Monte Carlo works in these scenarios because:

1. Uniform random sampling means that we ensure points are scattered throughout region and cover it well
2. Based on the law of large numbers, as we increase the number of samples, the average area value will converge to the true area value. So, using a large number of samples improves our estimate.

Use Monte Carlo to estimate integration: Assume there is an integration $I = \int_a^b f(x)dx$. Then let X to be a random variable between a and b . Assume we take n random samples. Then the estimation is

$$I_n = (b - a) \frac{1}{n} \sum_i^n f(X_i) = (b - a) \cdot \mathbb{E}[f(X_i)]$$

Error of Monte Carlo method: Let Z to be a random variable with normal distribution $N(0, \sigma^2)$, then

$$Error \rightarrow \frac{1}{\sqrt{n}} Z$$

Therefore, the asymptotic behavior of the Monte Carlo method is $\mathcal{O}(\frac{1}{\sqrt{n}})$.

Random numbers:

```
import numpy as np

a = np.random.rand(3, 2)
# 3 x 2 array of random numbers 0 to 1

b = np.random.randint(100)
# random int from 0 to 100
x = np.random.randint(100, size=(5))
# array of size 5 with random number 0 to 100
x = np.random.uniform(a, b)
# random number uniformly distributed between a and b

c = np.random.choice([1, 2, 3, 4])
# will randomly return one of the values within the array
```