

# Introduction to Programming and Computational Physics

## Lecture 13

- Monte Carlo integration
- Simulation of non-uniform distributions
- Random walks

# Monte Carlo integration

Monte Carlo integration is typically used to estimate multidimensional definite integral when the shape of the domain is not regular and classical numerical methods can't be applied. Convergence speed is quite low compared to the other methods seen in lecture 10.

A simple Monte Carlo method for one-dimensional integrals is the **sample mean** method. The method is based on **the mean-value theorem of calculus**, which states that the definite integral

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

is determined by the average value of the integrand  $f(x)$  in the range  $a \leq x \leq b$

The sample mean is calculated by sampling the function with a sequence of uniform random numbers.

$$\int_a^b f(x)dx \approx (b-a) \underbrace{\frac{1}{N} \sum_{i=1}^N f(x_i)}_{\text{sample mean}} \quad x_i \in U[a, b]$$

The error in the Monte Carlo method approaches zero as  $\sim \frac{1}{\sqrt{n}}$  where  $n$  is the number of samples.

The  $n$  dependence of the error is independent of the nature of the integrand.

If we define the variance as:

$$\sigma^2 = \left| \langle f^2 \rangle - \langle f \rangle^2 \right| = \left| \frac{1}{n} \sum_{i=1}^n f(x_i)^2 - \left( \frac{1}{n} \sum_{i=1}^n f(x_i) \right)^2 \right|$$

$\sigma/\sqrt{(n-1)}$  is the *standard deviation of the mean* and can be assumed as an estimation of the error for the sample mean, while  $(b-a)\sigma/\sqrt{(n-1)}$  is an estimation of the error for the integral

To sample from a  $U(a,b)$  distribution, use

```
x = (b-a) * (double) rand() / (double) RAND_MAX + a;
```

# Transformation of Uniform Deviates

Sampling of random variates from nonuniform distribution is usually done by applying a transformation to uniform variates.

Each realization of the nonuniform random variable might be obtained from a single uniform variate or from a sequence of uniforms.

For some distributions there may be many choices of algorithms. They differ in speed, accuracy, storage requirement and complexity of coding.

# Inverse CDF method

Given a random variable  $X$  from a given probability density function  $f(X)$ , the *cumulative distribution function* (CDF) is the function  $F_X$  defined by:

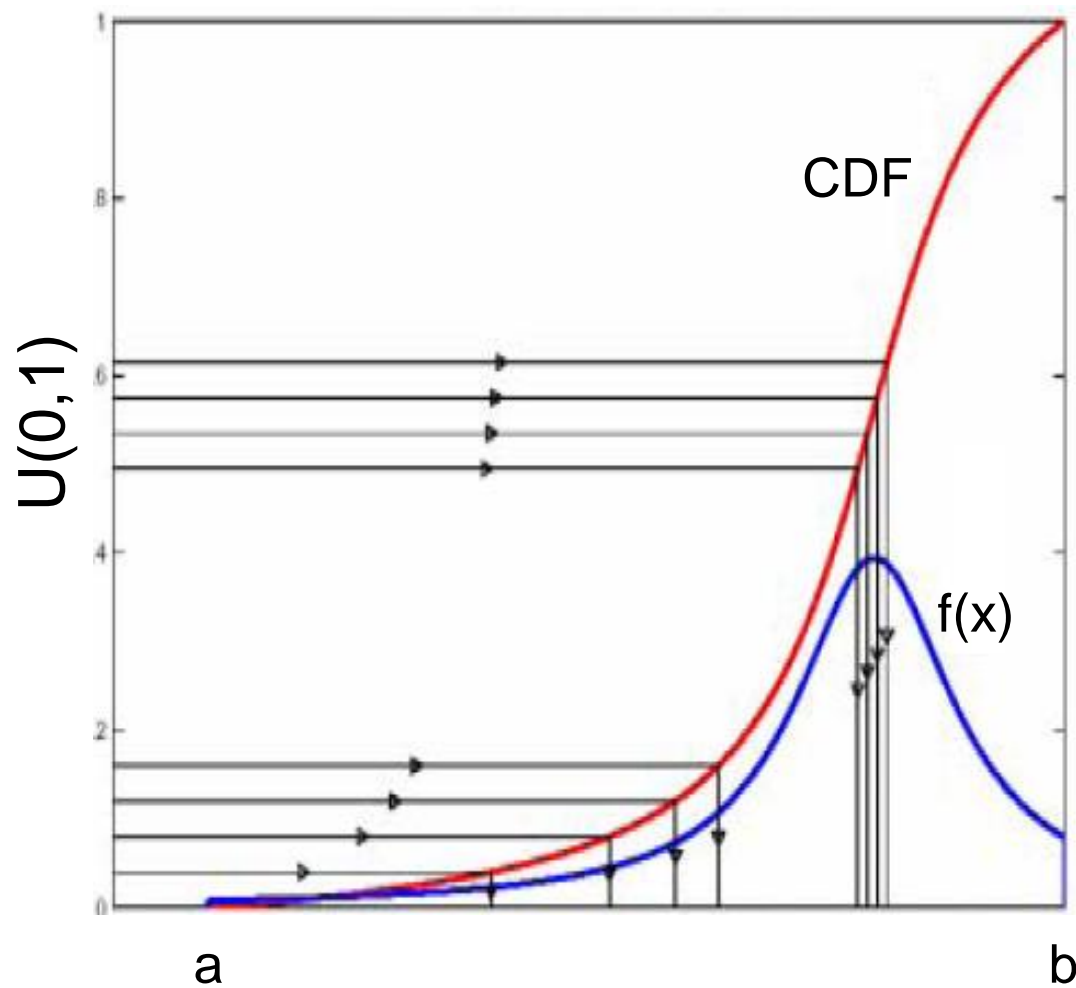
$$F_X(t) = P(t \leq X) = \int_a^X f(t)dt$$

If  $X$  is a random variable with a continuous CDF  $F_X$  then the random variable

$$U = F_X(X)$$

has a  $U(0,1)$  distribution. This provides a very simple relationship between a uniform random variable  $U$  and a random variable  $X$  with cumulative distribution function  $F$

$$X = F_X^{-1}(U)$$



Limitations:

- $f(x)$  must be integrable
- CDF must be invertible

Example

To generate from the distribution:

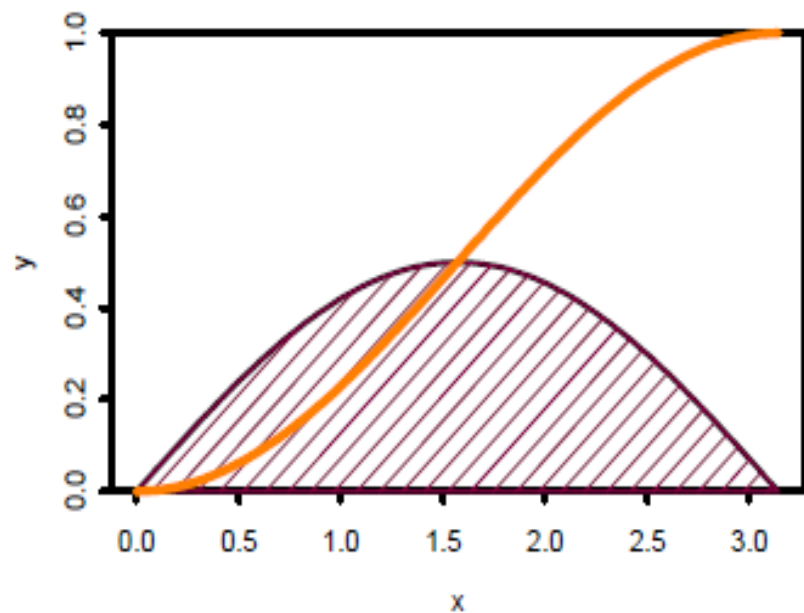
$$f(x) = \begin{cases} \sin(x)/2 & 0 \leq x \leq \pi \\ 0 & \text{elsewhere} \end{cases}$$

The cumulative distribution in this case is easily shown to be:

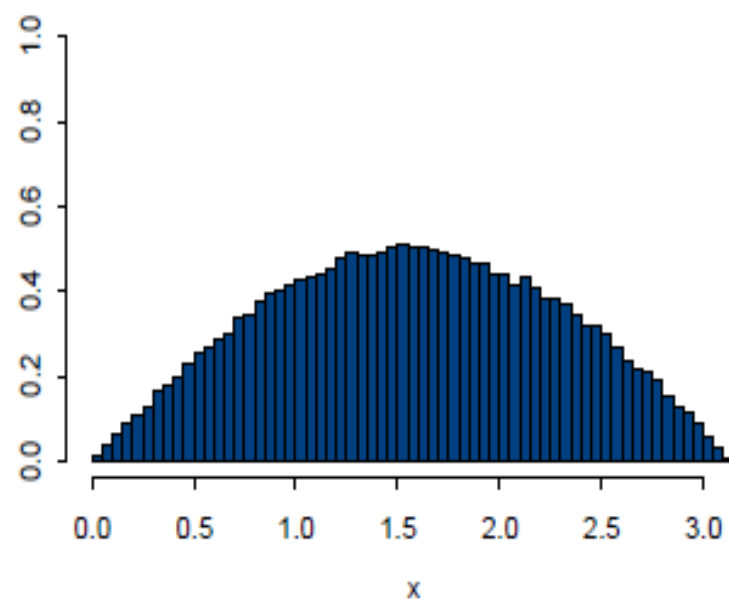
$$P(x) = \begin{cases} 0 & x < 0 \\ (\cos(0) - \cos(x))/2 & 0 \leq x \leq \pi \\ 1 & x > \pi \end{cases}$$

The inverse function to P is therefore:

$$P^{-1}(z) = \arccos(\cos(0) - 2z)$$



The true distribution and  
its cumulative function



Sample of 100,000  
generated using the  
transformation method



# Simulating random numbers from specific distributions

For the most used distribution, dedicated algorithms have been developed.

The starting point for any algorithm is a generator of uniform random numbers. The aim of these algorithms is to use a small number of uniforms to yield a variate of the target distribution.

# Normal distribution

The normal (Gaussian) distribution which we denote by  $N(\mu, \sigma^2)$  has the probability density

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

If  $Z \sim N(0, 1)$  and  $X = \sigma Z + \mu$  then  $X \sim N(\mu, \sigma^2)$

Because of this simple relationship, it is sufficient to generate deviates from the *standard* normal distribution  $N(0, 1)$

## Method 1: Using the Central Limit Theorem

Let  $X_1, X_2, \dots, X_n$  be independent and identically distributed random variables each having finite values of expectation  $\mu$  and variance  $\sigma^2$ . As the sample size  $n$  increases, the distribution of the sample average of these random variables approaches the normal distribution with a mean  $\mu$  and a variance  $\sigma^2 / n$  irrespective of the shape of the original distribution.

Let  $S_n = X_1 + X_2 + \dots + X_n$  and  $Z_n = \frac{S_n - n\mu}{\sigma\sqrt{n}}$

The distribution of  $Z_n$  converges toward  $N(0,1)$  as  $n$  approaches infinity.

If we consider the uniform distribution  $U(0,1)$  it has  $\mu = 0.5$  and variance  $\sigma^2 = 1/12$

If we fix then our  $n=12$  we will have that

$$S_n = X_1 + X_2 + \dots + X_{12} \quad X_i \in U(0,1)$$

And  $Z_n = \frac{S_n - 12 \cdot 0.5}{(1/\sqrt{12})\sqrt{12}} = S_n - 6$  Will approximate a Gaussian  $N(0,1)$

This method is very slow as for 12 generated random numbers from  $U(0,1)$  we have only one deviate from  $N(0,1)$

## **Method 2:** The Box-Muller method

If  $U_1$  and  $U_2$  are independently distributed as  $U(0,1)$  and

$$X_1 = \sqrt{-2\log(U_1)} \cos(2\pi U_2)$$

$$X_2 = \sqrt{-2\log(U_1)} \sin(2\pi U_2)$$

then  $X_1$  and  $X_2$  are independently distributed as  $N(0,1)$

The method require the generation of two deviates from  $U(0,1)$  for two deviates from  $N(0,1)$  but the evaluation of one square root and two trigonometric functions make it rather slow.

**Method 3:** An acceptance/rejection polar method

$v_1$  and  $v_2$  are independently generated from  $U(-1,1)$  and we set:

$$r^2 = v_1^2 + v_2^2$$

If  $r^2 < 1$  then

$$x_1 = v_1 \sqrt{\frac{-2\log(r^2)}{r^2}}$$

$$x_2 = v_2 \sqrt{\frac{-2\log(r^2)}{r^2}}$$

are deviates from  $N(0,1)$ , otherwise  $v_1$  and  $v_2$  are rejected

#accepted / #generated = 78.6%

# Exponential distribution

The exponential distribution with parameter  $\lambda > 0$  has the probability density

$$p(x) = \lambda e^{-\lambda x} \quad 0 \leq x \leq \infty$$

If  $Z$  has the *standard* distribution ( $\lambda=1$ ), then  $X=Z/\lambda$  has the exponential distribution with parameter  $\lambda$ . Because of this simple relationship, it is sufficient to develop methods to generate deviates from the standard exponential distribution.

The inverse CDF method is very easy to implement and is generally satisfactory. The method is to generate  $u$  from  $U(0,1)$  and then take

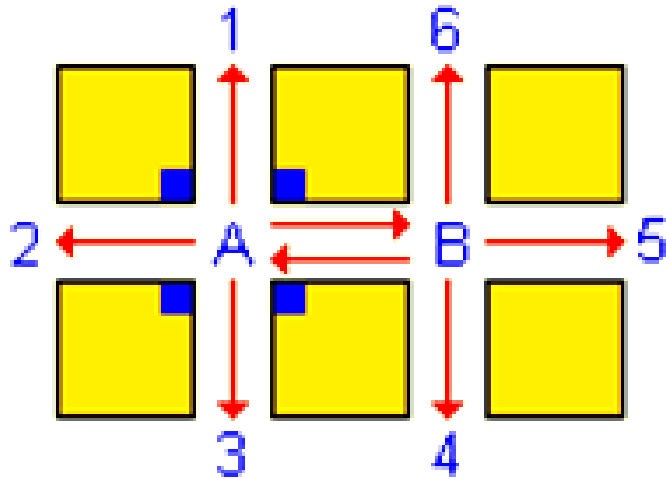
$$x = -\frac{\log(u)}{\lambda}$$

# Random walks

A **random walk** is a mathematical formalization of a trajectory that consists of taking successive random steps. The results of random walk analysis have been applied to a large number of fields, including physics, computer science, economics and so on.

Brownian motion is a typical example where Monte Carlo method can perform a realistic simulation of a physical process

# The “Drunken Sailor’s” random walk



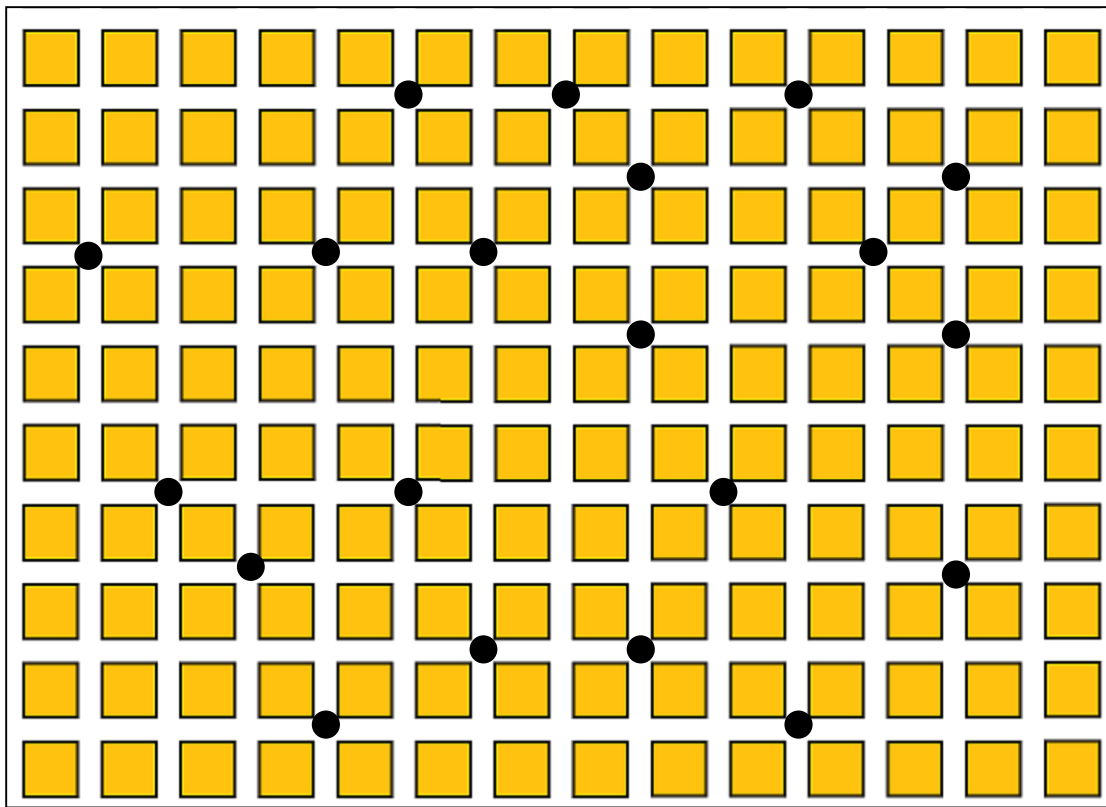
Consider a town consisting of 3x2 blocks. A "drunken sailor" stands in one of the two crossroads and he wants to leave the town. Since the sailor is very drunk, the probabilities of travelling up, down, left or right are equal. What is the probability for the sailor to reach each one of the six town exits?

Starting from A the probability to move to each one of the exits 1, 2, 3, or to B is obviously  $1/4$ . Supposing that the sailor has moved to B, then again the probability to move to the exits 4, 5, 6, or back to A is again  $1/4$ , therefore  $1/4$  of the initial probability to be found there, i.e.  $1/4(1/4) = 1/4^2$ . Again, supposing that he has returned to A the probability to move to the exits 1, 2, 3, or back to B are  $1/4(1/4^2) = 1/4^3$  and so on. Therefore, by summing up the probabilities for each exit, we have:

$$\text{Probability for each one of the exits 1, 2, 3: } \frac{1}{4^1} + \frac{1}{4^3} + \frac{1}{4^5} + \dots = \sum_{i=1}^{\infty} \left( \frac{1}{4^{2i-1}} \right) = 0.2666\dots$$

$$\text{Probability for each one of the exits 4, 5, 6: } \frac{1}{4^2} + \frac{1}{4^4} + \frac{1}{4^6} + \dots = \sum_{i=1}^{\infty} \left( \frac{1}{4^{2i}} \right) = 0.0666\dots$$





What if...

- the town is much bigger and it has all the exit *closed*
- drunken sailors are many more and they keep the same direction unless they hit another sailor or an exit(if this is the case they revert their direction)

And we want to evaluate the average number of sailors hitting one exit out of a certain number of steps...

...on the way of modeling a Brownian motion...