Aalto University
School of Science
and Technology

Monte Carlo method in particle transport simulations
Lecture 1 – Introduction to Monte Carlo method

Jaakko Leppänen

Department of Applied Physics
Aalto University, School of Science
Jaakko.Leppanen@aalto.fi

Sept. 15, 2020

Topics of this lecture

Lecture topics:

- Brief introduction to radiation and particle transport problems
- General description of the Monte Carlo method
- Probability and statistics: basic concepts
- Sampling from distributions

1st programming exercise

Background: particle transport calculations

Radiation and particle transport problems are encountered in nuclear engineering, but also in various industrial applications, radiography, radiotherapy, radiation shielding, particle physics and space research.

The applications and methods depend on particle/radiation type. A clear division can be made between neutral particles:

neutrons – fission, fusion and other energetic nuclear reactions

photons – radioactive decay and nuclear reactions

and charged particles:

electrons and positrons – beta decay, accelerator applications

heavy ions – alpha decay, accelerator applications and space physics

The topics of this course are limited to neutral particles. Programming exercises cover methods used for neutron transport and fission reactor physics, but many of the same algorithms are directly applicable to photon transport as well.

Background: particle transport calculations

Fission reactor physics essentially deals with a neutron transport problem on a self-sustaining system (chain reaction). The challenges include:

- Heterogeneity of fuel and reactor core – neutron mean-free-path varies from millimeters to tens of centimeters, neutrons experience the surrounding world in high detail
- Complicated energy dependence of neutron reactions – cross sections may vary several orders of magnitude over a narrow energy interval
- Complicated angular dependence of collisions – scattering is the dominant reaction mode (> 90% of all interactions) and highly anisotropic in hydrogeneous media (water)
- Different time scales for prompt and delayed neutrons – prompt fission chains are typically completed within tens of microseconds, emission of delayed neutrons may take several seconds

In this course it is assumed that the transport problem is linear, i.e. that:

- Neutrons do not interact with each other (good approximation for neutral particles)
- Cross sections are independent of neutron flux and constant in time (poor approximation for operating nuclear reactors)

In reality, reactor modeling deals with a coupled problem, which is typically handled by operator splitting and iterating between different solvers.

Transport theory: deterministic approach

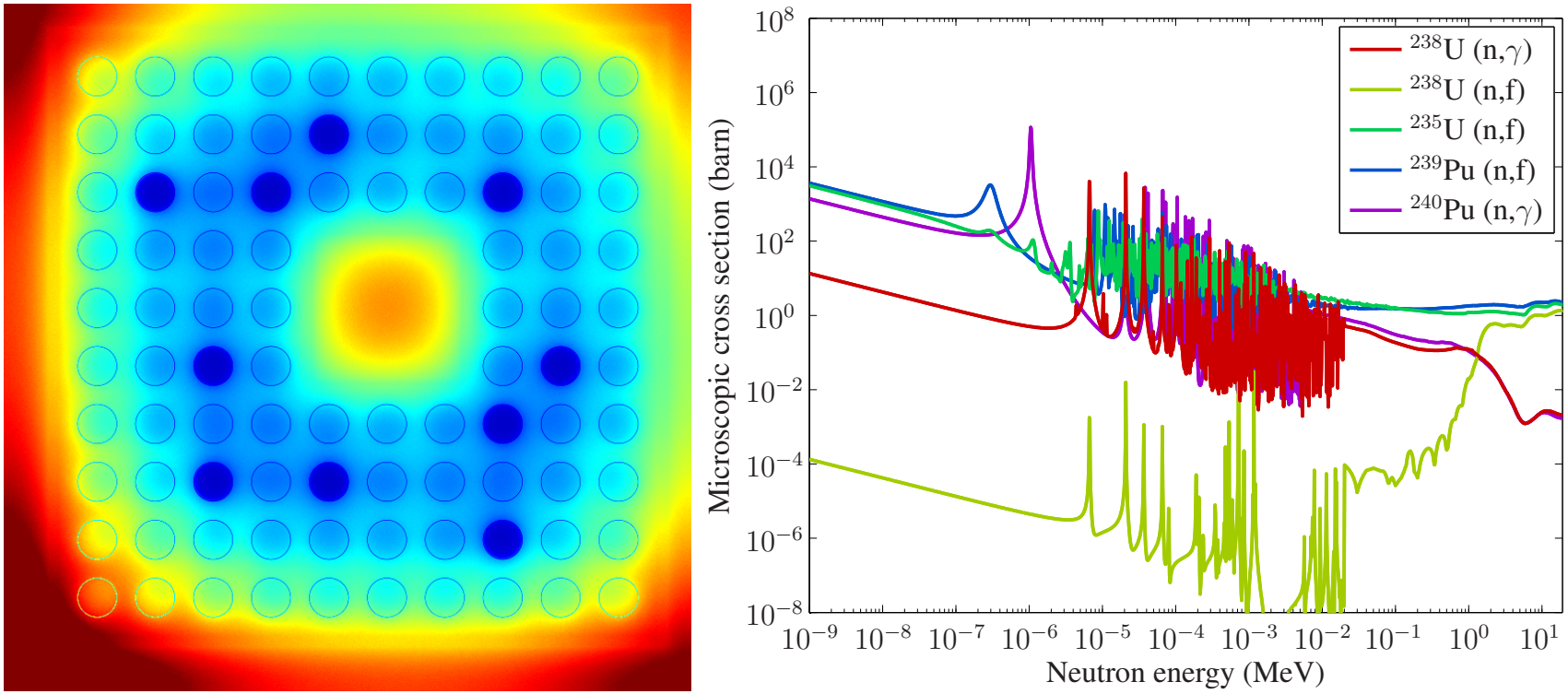


Figure 1: Left: Neutron density distribution in a BWR fuel assembly. The distribution peaks in the internal and external moderator channels where thermal neutrons are collected. Left: fission and radiative capture cross sections of major actinides.

Transport theory: deterministic approach

All deterministic transport methods rely on the concept of neutron flux, $\psi(\boldsymbol{r}, \hat{\boldsymbol{\Omega}}, E)$, which is essentially a six-dimensional density-like function describing the collective behavior of the neutron population.

The neutron flux is needed for forming the transport equation, which is essentially a balance equation in the six-dimensional phase space:

$$\frac{1}{v}\frac{\partial}{\partial t}\psi(\boldsymbol{r}, \hat{\boldsymbol{\Omega}}, E, t) + \hat{\boldsymbol{\Omega}} \cdot \nabla\psi(\boldsymbol{r}, \hat{\boldsymbol{\Omega}}, E, t) + \Sigma(\boldsymbol{r}, E)\psi(\boldsymbol{r}, \hat{\boldsymbol{\Omega}}, E, t) = Q + S + F \tag{1}$$

where Q is the external source, S is the scattering source:

$$S(\boldsymbol{r}, \hat{\boldsymbol{\Omega}}, E, t) = \int_{4\pi}\int_{0}^{\infty} \Sigma_{\mathrm{s}}(\mathbf{r}, \hat{\boldsymbol{\Omega}}' \rightarrow \hat{\boldsymbol{\Omega}}, E' \rightarrow E)\psi(\mathbf{r}, \hat{\boldsymbol{\Omega}}', E', t)d\hat{\boldsymbol{\Omega}}'dE' \tag{2}$$

and F is the fission source:

$$F(\boldsymbol{r}, \hat{\boldsymbol{\Omega}}, E, t) = \frac{\chi(E)}{4\pi}\int_{0}^{\infty} \nu\Sigma_{\mathrm{f}}(\mathbf{r}, E')\psi(\mathbf{r}, \hat{\boldsymbol{\Omega}}', E', t)dE' \tag{3}$$

Similar transport equation can be written for photons. The equation cannot be solved without approximations because of the complicated energy dependence of cross sections and angular dependence of streaming term and scattering source.

Transport theory: deterministic approach

Deterministic transport methods generally rely on at least three fundamental approximations:

1) The geometry is discretized into a number of homogeneous material regions
2) The continuous energy dependence of cross sections is condensed into a number of discrete energy groups
3) The angular dependence of double-differential scattering cross sections is represented by functional expansions, the directional dependence of flux is represented by functional expansions or discrete directions

The first two approximations are common to all deterministic transport methods, the treatment of angular dependence is what differentiates the methods from each other (S_n, P_n, method of characteristics, diffusion theory, etc.).

But it should be noted that, in the end, the flux itself is not a physical measurable quantity,[1] but rather the mathematical means to calculate physical reaction rates:

$$R_x = \int_V \int_{\hat{\boldsymbol{\Omega}}} \int_E \Sigma_x(\boldsymbol{r}, E)\psi(\boldsymbol{r}, \hat{\boldsymbol{\Omega}}, E)dV\, d\hat{\boldsymbol{\Omega}} dE \tag{4}$$

[1] This interpretation is subject to argument, but what cannot be argued is that flux can only be measured via physical reaction rates.

Transport theory: Monte Carlo approach

The Monte Carlo approach to particle transport problems is fundamentally different – instead of dealing with the entire population (flux), the transport process is viewed from the perspective of an *individual particle*.

The procedure is based on a simulated random walk, in which the probability of each random event depends only on the position (material located at the collision point) and the energy of the transported particle. Interactions are described by probability distributions, representing the "laws of physics", from which the outcome is randomly sampled.

When the simulation is repeated for a very large number of particle histories, various results describing the physical behavior of the system can be obtained by counting interactions, track lengths, boundary crossings, etc. and applying statistical methods on the collected results.

It is important to note that the Monte Carlo method does not solve the transport equation, but rather provides statistical estimators for integrals of the form:

$$\int_V \int_{\hat{\boldsymbol{\Omega}}} \int_E f(\boldsymbol{r}, \hat{\boldsymbol{\Omega}}, E)\psi(\boldsymbol{r}, \hat{\boldsymbol{\Omega}}, E)dV\, d\hat{\boldsymbol{\Omega}} dE \tag{5}$$

where f can be an arbitrary response function depending on particle coordinates in the six-dimensional phase space, most typically a cross section.

Transport theory: Monte Carlo approach

Monte Carlo simulation has several fundamental advantages over the deterministic approach:

- Events occur at discrete locations, no need to integrate flux over space to obtain solution for the transport equation – complex geometries can be modeled at an arbitrary resolution
- Collisions occur at discrete energies, no need to integrate flux over energy to obtain solution for the transport equation – interaction data can be handled in continuous-energy form without group condensation
- Time dependence of reaction chains can be modeled explicitly without additional effort

But there are also a few fundamental disadvantages:

- Result estimates are based summation over simulated events, and they represent integrals over space, direction and energy (and time) – differential quantities are difficult to calculate
- All results are random variables, associated with a statistical uncertainty – accuracy of the simulation depends on the number of simulated particle histories

The fact that interactions can be handled as independent events is based on the same linearity assumption that makes cross sections independent of flux in the formulation of the transport equation. Coupled simulations with the Monte Carlo method require similar iteration between different solvers.

Monte Carlo method: general

Computations based on random sampling date back to the 18th century,[2] but the theoretical basis (Markov's chains) was not formulated until the early 20th century. The method became practical along with the development of first electronic computers in the late 1940's, capable of producing random numbers and processing data in an automated manner.

The Monte Carlo method is typically used for solving complicated linear problems that can be divided into multiple simple and separate sub-tasks. Transport calculation is actually a very good example, since solving the particle flux (transport equation) is a complicated task, but simulating individual particle histories is relatively straightforward.

In practice, Monte Carlo particle transport simulation requires:

1) Geometry routine (simple linear algebra and vector calculus)
2) Physics model, handling the individual interactions with target nuclides (relatively simple nuclear physics and a lot of data)
3) Collecting the results (simple statistics)

The first Monte Carlo applications involved radiation transport problems in the aftermath of the Manhattan Project in the late 1940's. The method has been developed along with computing power, especially during the early years, and computer capacity is still one of the major factors limiting the scale of Monte Carlo simulations.

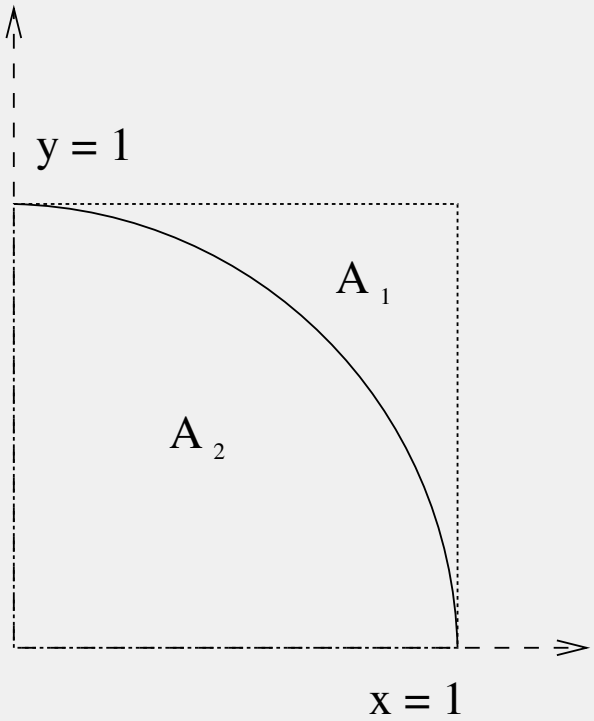
[2] See, for example, Buffon's needle problem for calculating the value of π

Monte Carlo method: simple example

Example 1: Monte Carlo estimate for π

Consider the quarter of a unit circle:

- Randomly sample M points between $(0 < x < 1, 0 < y < 1)$
- Count the number of points falling inside the circle: m
- Based on the geometry it is known that the probability of a point falling inside the circle is the ratio of areas: $P = A_2/A_1 = \pi/4$
- From the simulation: $m/M \longrightarrow P$ when $M \longrightarrow \infty$
- Or: $4m/M \longrightarrow \pi$



Matlab example: `piex1.m`

Statistical mean and standard deviation

In the previous example the ratio of successful samples to total approached value $\pi/4$, but each repetition produced a slightly different, random result. The larger the number of samples, the closer the values were to the actual result.

From basic courses of probability and statistics it is recalled that the sample mean is given by:

$$\overline{x} = \frac{1}{N} \sum_{n=1}^{N} x_n \tag{6}$$

and the associated standard deviation:[3]

$$\sigma(\overline{x}) = \sqrt{\frac{1}{N(N-1)} \sum_{n=1}^{N} (x_n - \overline{x})^2} \tag{7}$$

or in a more practical form:

$$\sigma(\overline{x}) = \sqrt{\frac{1}{N(N-1)} \left[\sum_{n=1}^{N} x_n^2 - \frac{1}{N} \left(\sum_{n=1}^{N} x_n \right)^2 \right]} \tag{8}$$

The standard deviation is also associated to the variance, σ^2.

[3] Assuming that values x_n are not correlated.

Central limit theorem

The standard deviation σ is a measure of how much a value, randomly selected from distribution $f(x)$, differs (on the average) from the mean $\overline{x}$. This information has very little practical significance if the type of distribution $f(x)$ is unknown.

Luckily, the central limit theorem states that:

> *The sum of arbitrarily distributed random variables is itself a random variable. Under certain conditions, the distribution of this variable approaches the normal distribution, as the sum of terms approaches infinity.*

The main conditions are:

1) The number of terms is sufficiently large
2) All terms are similarly distributed
3) The values are independent of each other (no correlations)

These conditions are easily met in Monte Carlo simulations, although the correlation between batches (violation of condition 3) may become a problem in criticality source simulations.

Confidence intervals

The significance of central limit theorem is that, since the type of the distribution is known for the simulated results, standard deviation can be associated with the more quantitative concept of confidence interval, which gives the range of values between a minimum and maximum positioned symmetrically about the mean.

For the normal distribution:

- 68% confidence interval: $\overline{x} \pm \sigma$
- 95% confidence interval: $\overline{x} \pm 1.96\sigma$
- 99% confidence interval: $\overline{x} \pm 2.58\sigma$

It is important to note that these confidence intervals apply only to the normal distribution.

The confidence intervals of Monte Carlo simulation are usually formed by assuming that the results follow the normal distribution, i.e. that the conditions of the central limit theorem are met. When this is not the case, statistical precision is either under- or over-estimated.[4]

[4] Correlations between values that should be independent also leads to the under-estimation of the standard deviation itself.

Confidence intervals

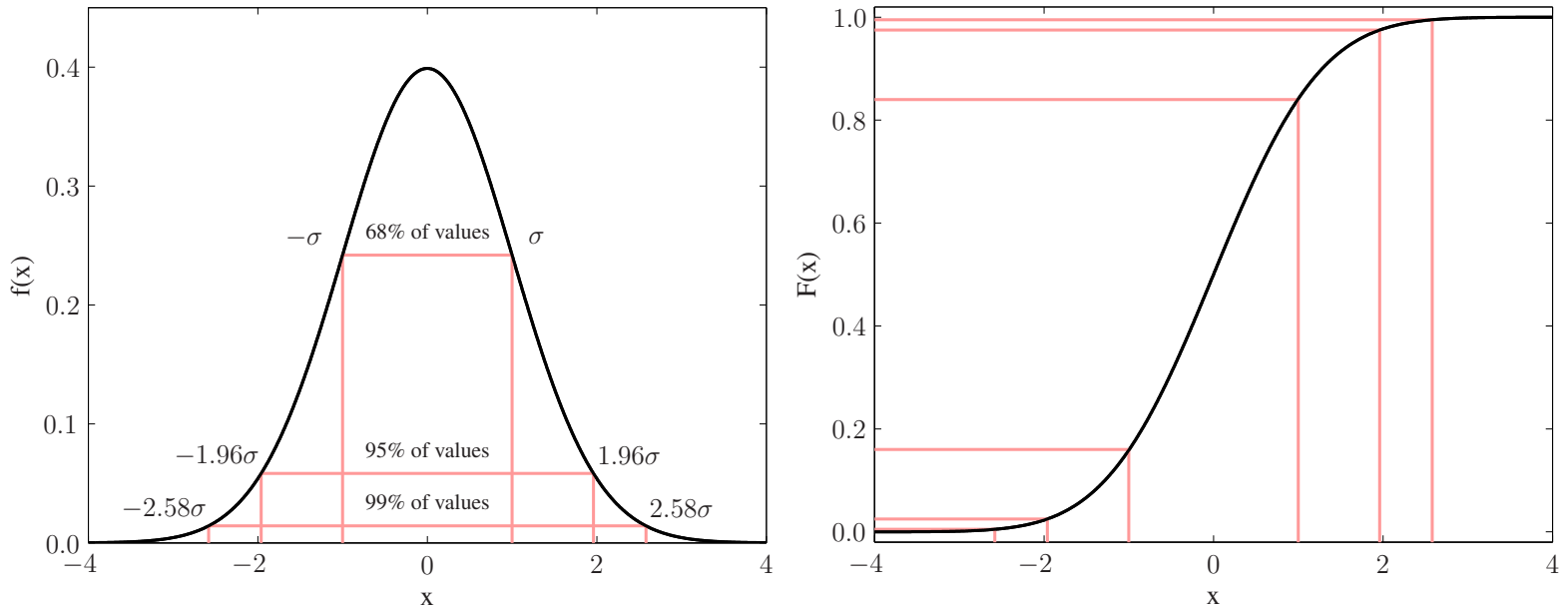


Figure 2: The (0,1) normal distribution with 68%, 95% and 99% confidence intervals. Left: probability density function (PDF), Right: cumulative distribution function (CDF). The PDF gives the probability of a value on interval $[x, x + dx]$. The CDF is obtained by integrating the distribution from $-\infty$ to x.

Monte Carlo method: simple example

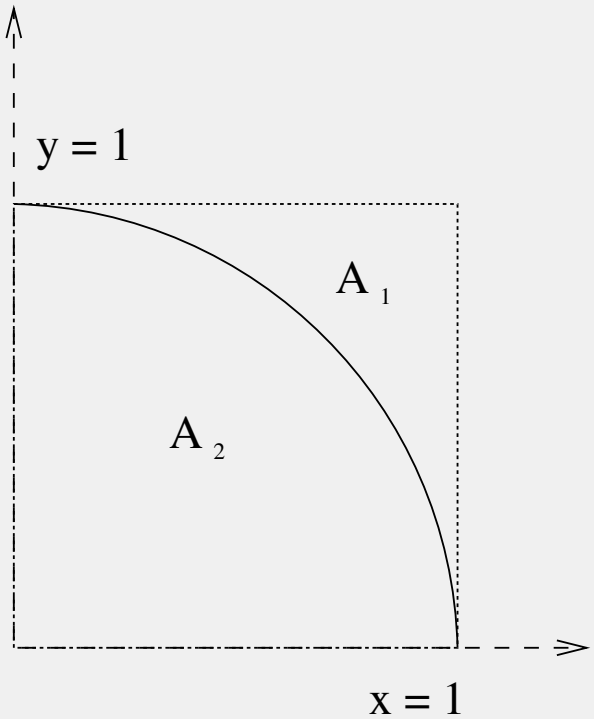
Example 2: Monte Carlo estimate for π with statistical errors

Consider the same quarter of a unit circle from the previous example, but instead of calculating the estimate directly, the problem is divided in two parts:

1) Randomly sample M points between $(0 < x < 1, 0 < y < 1)$, and calculate estimate $x_i = 4m/M$, where m is the number of points falling inside the circle
2) Repeat the procedure N times to obtain estimates $x_1, x_2, \ldots x_N$

Based on the central limit theorem, the values follow the normal distribution, which allows applying the known confidence intervals.

Matlab example: `piex2.m`



Monte Carlo method: simple example

Example 2: Monte Carlo estimate for π with statistical errors

Example results (M = 100):[a]

$N = 10$	$\pi \approx 3.20800 \pm 0.05053$	$[3.10897, 3.30703]$
$N = 100$	$\pi \approx 3.15320 \pm 0.01581$	$[3.12222, 3.18418]$
$N = 1000$	$\pi \approx 3.14476 \pm 0.00544$	$[3.13409, 3.15543]$
$N = 10000$	$\pi \approx 3.13973 \pm 0.00165$	$[3.13649, 3.14297]$
$N = 100000$	$\pi \approx 3.14141 \pm 0.00052$	$[3.14040, 3.14243]$
$N = 1000000$	$\pi \approx 3.14130 \pm 0.00016$	$[3.14098, 3.14162]$

[a] The value of π with five decimals is 3.14159

Figure-of-merit

The standard deviation in the previous example depends on:

1) The variation of values $x_1, x_2, \ldots x_N$, which depends on the number of points M sampled in the geometry
2) The number of repetitions N

If the number of sampled points M is kept constant, it is observed that:

1) The standard deviation falls as $\sigma \sim 1/\sqrt{N}$ (or variance as $\sigma^2 \sim 1/N$)
2) The computing time increases as $T \sim N$

This implies that product $\sigma^2 T$ is approximately constant regardless of N, which allows the definition of figure-of-merit:

$$FOM = \frac{1}{\sigma^2 T} \tag{9}$$

Since two algorithms performing the same task can have different running times and convergence rates,[5] the FOM provides a useful quantitative measure of computational efficiency.

Since standard deviation depends the magnitude of the result, FOM is often calculated from the relative statistical error instead.

[5] In Monte Carlo transport simulations it is very common that variance reduction techniques lead to both faster convergence and longer simulation time, so looking at the standard deviation alone does not give a fair estimate for the performance of the algorithm.

Notes on statistical precision

It is important to note that standard deviation or confidence intervals do not measure the accuracy of the simulation. The correct interpretation is rather related to statistical precision, for example:

- The 95% confidence interval means that a random value selected from the distribution falls on interval $[\overline{x} - 1.96\sigma, \overline{x} + 1.96\sigma]$ with 95% probability
- The probability that the value is outside the range is 5%

An alternative interpretation is that if N repetitions gives a result with mean value $\overline{x}$ and standard deviation σ, the result obtained with a very large (infinite) number of repetitions falls on interval $[\overline{x} - 1.96\sigma, \overline{x} + 1.96\sigma]$ with 95% probability.

The results are also subject to systematic error resulting from approximations, errors and uncertainties in data, bugs in computer code, etc. The outcome of the simulation is only as good as the model: garbage in – garbage out.

Probability distributions

The laws of physics in Monte Carlo simulation are essentially represented by probability distributions, describing the outcome of random events. The *probability density function* (PDF), $f(x)$, is defined in such way that the probability of the event occurring between x and $x + dx$ is given by:

$$dP = f(x)dx \tag{10}$$

Consequently, the probability of the event occurring on interval $[x_0, x_1]$ of the variable is:

$$P(x_0 < x < x_1) = \int_{x_0}^{x_1} dP = \int_{x_0}^{x_1} f(x)dx \tag{11}$$

The probability that the event occurs before the variable reaches a certain value is given by the *cumulative distribution function* (CDF), $F(x)$, which is calculated from (10) by direct integration:

$$F(x) = P(x' < x) = \int_{-\infty}^{x} f(x')dx' \tag{12}$$

The PDF's are assumed to be properly normalized, i.e. the integration of $f(x)$ over all value space must yield $P = 1$:

$$\lim_{x \to \infty} F(x) = 1 \tag{13}$$

Sampling from probability distributions: inversion method

Probability distributions in Monte Carlo simulations are used for sampling the outcome of various events: distance to collision site, interaction after collision, direction and energy of scattered particle, etc.

The simplest algorithm for sampling random values from distribution $f(x)$ is called the inversion method, which is based on the inverse of the CDF, $F^{-1}(x)$, and a uniformly distributed random variable ξ on the unit interval.[6]

The algorithm can be described as follows:

1) Calculate $F(x)$ and $F^{-1}(x)$ from $f(x)$
2) Sample a uniformly distributed random variable ξ on interval $[0, 1)$
3) Set $F(x) = \xi$
4) Value for variable x is obtained from the inverse of the CDF: $x = F^{-1}(\xi)$

Inversion sampling is an efficient way to produce samples from $f(x)$ if the distribution is provided in functional form and $F(x)$ and $F^{-1}(x)$ can be resolved.

This method is revisited when sampling particle path lengths in homogeneous medium.

[6] The production of uniformly distributed (pseudo-)random numbers can be done with computational algorithms, but there is a surprising amount of science behind the methodology.

Sampling from probability distributions: inversion method

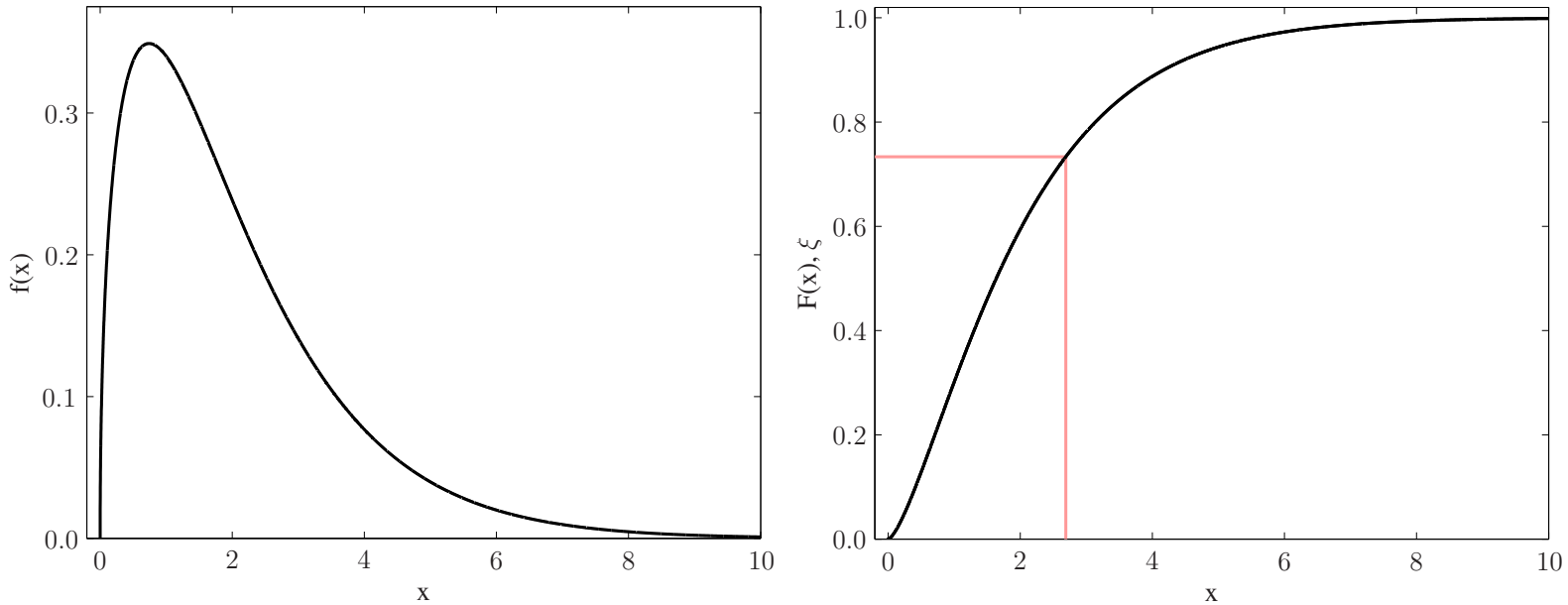


Figure 3: Sampling a random number from probability distribution using the inversion method. Left: Probability density function (PDF), $f(x)$, from which the value is sampled. Right: The corresponding cumulative distribution function (CDF), $F(x)$. A random number ξ is sampled on the unit interval, and the value is set $\xi = F(x)$. The distributed variable is obtained using the inverse function: $x = F^{-1}(\xi)$. In the example, ξ = 0.733, x = 2.69.

Sampling from probability distributions: rejection method

The inversion sampling method cannot be used if F or F^{-1} cannot be resolved in closed form. This is the case, for example, for the normal distribution and the Maxwell-Boltzmann distribution. In such case, one option is to obtain the value for F^{-1} numerically.

Another option is to use rejection sampling, which is based on two distribution functions, the original distribution $f(x)$ and a majorant function $g(x)$, which can be sampled using the inversion method, and for which:

$$g(x) \geq f(x) \tag{14}$$

for all values of x. The algorithm has two steps:

1) Sample random variable x from distribution $g(x)$ using the inversion method
2) Perform rejection sampling and accept the sample with probability $P(x) = f(x)/g(x)$

If the sample is rejected, the procedure is repeat from the beginning.

The method is best understood by considering the sampling of a uniform distribution of points inside a circle (see the previous example). Points are sampled uniformly on a square enveloping the circle, and accepted only if they fall inside the circle.

Another example is the Woodcock delta-tracking method used for sampling particle path lengths, discussed later on.

Sampling from probability distributions: tabular distributions

Energy and angular distributions of secondary particles are often given in tabular form, along with some interpolation rules for obtaining values between the tabulated points. In the case of linear-linear interpolation, the values of the PDF and the CDF are given by:

$$f(x) = \frac{x - x_{i-1}}{x_i - x_{i-1}}(f_i - f_{i-1}) + f_{i-1} \tag{15}$$

and

$$F(x) = \frac{x - x_{i-1}}{x_i - x_{i-1}}(F_i - F_{i-1}) + F_{i-1} \tag{16}$$

where i is the table index for which $x_{i-1} \leq x < x_i$, and f_{i-1}, f_i, F_{i-1} and F_i are the corresponding tabulated values of the distributions.

The sampling can be accomplished by selecting a uniformly distributed random number ξ on the unit interval, performing index search to obtain i such that $F_{i-1} \leq \xi < F_i$, and interpolating the value from:[7]

$$x = x_{i-1} + \frac{1}{a}\left(\sqrt{f_{i-1}^2 + 2a(\xi - F_{i-1})} - f_{i-1}\right) \tag{17}$$

where:

$$a = \frac{f_i - f_{i-1}}{x_i - x_{i-1}}\,. \tag{18}$$

[7] In practice, the procedure is not this simple, because the shape of the distribution may depend on particle energy which requires additional interpolation between two distributions.

1st programming exercise

Mandatory tasks:

- Implement the Monte Carlo algorithm described in the previous example (estimation of the value of π by sampling random points inside a circle)
- Look up the description of Buffon's needle problem and implement the corresponding algorithm
- Calculate mean values, standard deviations, running times and figure-of-merits using both algorithms, compare the results and performance
- All input values must be read from a text file

Bonus tasks:

- Implement a statistical test that checks that the results follow normal distribution (+2 points)

Milestones:

- Selection of programming language and implementation of basic structure of the source code
- Implementation of subroutines to handle the sampling of uniformly distributed random variables, collection of statistical results, timing and file I/O
- Data structures for storing statistical variables

1st programming exercise

Tips:

- In the Buffon's needle problem, the length of the needle makes a difference for computational performance.
- In theory, the Buffon's needle algorithm should be able to produce a better FOM, but in practice this depends on the implementation (computational bottlenecks).