GSSI LE-6 course
Monte Carlo Techniques

Luciano Pandola
`pandola@lns.infn.it`
Academic Year 2020-2021

1 Introduction to Monte Carlo techniques: applications and use cases

Introduction to Monte Carlo, what it is: numerical solution to a (complex) macroscopic problem by simulating the microscopic interactions among the components. Calculates average behaviour of a system (which is usually what one wants to know about). Uses random sampling, until convergence is achieved.
Class of numerical methods which employ random numbers.

Not only physics but also finance, social sciences, traffic flow. Not only applications that are intrinsically probabilistic, but also deterministic problems, e.g. numerical integration.

Can be used to:

- to predict the results of an experiment (if the theory is well-known, e.g. QED);
- to prove/disprove a theory (e.g. if it is not validated or well-known) by comparison against experiments;
- to provide corrections to the theory (by comparison with experiments).

Particle tracking, for the simulation of a detector response (design and optimization of an experiment, support for data analysis).

More useful for complex phenomena: for "model" problems, the quickest solution is typically analytical/deterministic, while for "real life" problem, having an increasing complexity, the Monte Carlo approach becomes more and more efficient.

Monte Carlo usually implemented on computers (but the concept of Monte Carlo is older than computers!): mainly CPU load, and minimal I/O.

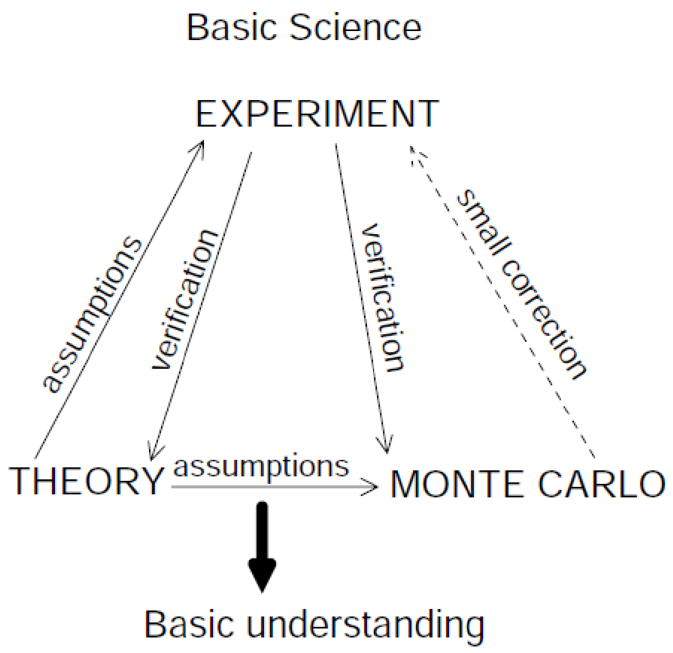

Figure 1: The role of Monte Carlo methods in basic science

Laplace and Buffon, XVIII Century. Boost by Ulam and von Neumann for the development of thermonuclear weapons, in the '50 (the naming "Monte Carlo" comes after von Neumann).

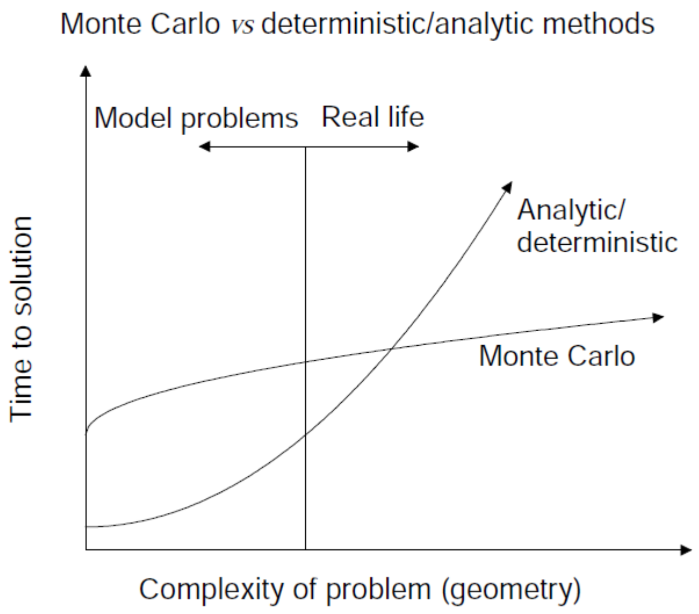

Figure 2: Time to solution of Monte Carlo vs. deterministic/analytic approaches.

2 Random variables

2.1 Probability distributions

Probability density function $p(x)$ for continuous random variables. Measure of the likelihood to observe a value x. Properties $p(x) \geq 0$ and $\int p(x) = 1$.

Moments:

$$\langle x^n \rangle = \int x^n p(x) dx. \tag{1}$$

The existence of all moments other than $\langle x^0 \rangle$ is not necessary (and not guaranteed). For instance, the Cauchy distribution

$$p(x) = \frac{1}{\pi} \frac{\Gamma}{\gamma^2 + x^2} \tag{2}$$

lacks of even-order moments.
If $\langle x \rangle$ (called "mean", or "expected value") and $\langle x^2 \rangle$ exist, one can define the variance (second-order central moment) as

$$\mathrm{var}(x) = \langle (x - \langle x \rangle)^2 \rangle = \langle x^2 \rangle - \langle x \rangle^2 \tag{3}$$

Variance is zero for the δ-function and greater than zero for any other $p(x)$.

In n-dimensional cases, correlations arise $p(x, y)$. Covariance

$$\mathrm{cov}(x, y) = \langle xy \rangle - \langle x \rangle \langle y \rangle \tag{4}$$

and

$$\mathrm{cov}(x, x) = \mathrm{var}(x) \tag{5}$$

If x and y are independent random variables, $p(x, y) = p_1(x)p_2(y)$, i.e. the pdf can be factorized into two terms. In this case, $\mathrm{cov}(x, y) = 0$ and $\mathrm{var}(x + y) = \mathrm{var}(x) + \mathrm{var}(y)$.

In general the moments are defined as $\langle x^n y^m \rangle$; only the existence of $\langle x^0 y^0 \rangle$ is required.

Marginal pdf when all variables but one are integrated out:

$$m(x) = \int p(x, y) dy \tag{6}$$

One can define the conditional probability as:

$$p(x, y) = m(x)p(y|x) = m(y)p(x|y) \tag{7}$$

Notice that the conditional probability $p(y|x)$ is a normalized pdf in y for any given value of x: as expected, fixing the value of x changes the probability distribution of y.

Cumulative distribution for 1D:

$$c(x) = \int_{x_{min}}^{x} p(x')dx' \tag{8}$$

Monotonically increasing function of x between 0 and 1.

A discrete distribution can be always seen as a composition of continuous distributions via the Dirac δ function:

$$p(x) = \sum_i p_i \delta(x - x_i) \tag{9}$$

In this case the cumulative probability is a sum of step functions.

2.2 Random number generation

Generation of *pseudo-random* numbers is the core of any Monte Carlo. Active field of theoretical developments. Need reproducibility of sequences. Different seeds are expected to produce (long) and independent series of random numbers. There are good reasons to wish a *pseudo-random* number with respect to a true random number.

Not all commonly-available random generators are suitable for “real” Monte Carlo applications, even if they come in standard packages (e.g. common `C++` libraries).

Some generator based on mod() functions (family of “linear congruential generators”, LCG, or Lehmer generators), after Lehmer (1949)

$$R_n = (aR_{n-1} + c) \bmod(m) \tag{10}$$

Depends on four parameters: R_0 (seed), a (multiplier), c (increment) and m (modulus). For $c = 0$, “multiplicative congruential generators”. The parameters a and m must be appropriately chosen to get a suitable pseudo-random sequence (there are “magic” choices). Here, m fixes the period of the generator, so it must be large.
One example, Park and Miller (1988) is

$$\begin{aligned} R_n &= 7^5 R_{n-1} \bmod(2^{31} - 1) \\ \xi_n &= \frac{R_n}{2^{31} - 1} \end{aligned} \tag{11}$$

here: $m = 2^{31} - 1$ and $a = 7^5 = 16807$ (MINSTD). However, the period of this generator is too low for good sampling. More sophisticated algorithms are required (implemented in scientific libraries, like CERNLIB, CLHEP, etc.).
Notice that the mod() function is very much suitable for computers, that

use binary logic: in this case, $m = 2^n$, with n depending on the variable type which is being used. For instance: $m = 2^{32}$ for `unsigned int` or $m = 2^{31}$ for `int` (one bit is taken by the sign). Upon multiplication or addition in the LCG, the high-order bits (those in the position $> n$) are simply lost, while the low-order bits are scrambled in a pseudo-random fashion.

2.3 Random sampling

Two main categories of sampling methods:

- direct sampling, for $p(x)$ which are (analytically) integrable and invertible;
- rejection method.

Direct sampling Cumulative pdf

$$c(x) = \int_a^x p(x')dx' \tag{12}$$

with properties $c(a) = 0$ and $c(b) = 1$. One can map the x to

$$t = c(x) \tag{13}$$

(and $0 \leq t \leq 1$). Since x is a random variable, t is also a random variable, with its own pdf $p_t(t)$. Due to the conservation of probability, one can apply a change of variable such that

$$p_t(t)dt = p(x)dx. \tag{14}$$

Therefore

$$p_t(t) = p(x)\left(\frac{dt}{dx}\right)^{-1} = p(x)\left(\frac{dc(x)}{dx}\right)^{-1} = p(x)\frac{1}{p(x)} = 1. \tag{15}$$

This means that the distribution $p_t(t)$ of *any* cumulative function $t = c(x)$ is *always* uniform between 0 and 1. As a practical recipe, one can invert

$$x = c^{-1}(t) \tag{16}$$

If t is generated from a uniform distribution in [0,1], i.e. $p_t(t)$, x will be distributed according to $p(x)$.
The method can be applied to continuous functions that are mapped on a discrete grid (requires interpolation on the cumulative function). In some cases the approximation of the cumulative function with steps is too crude (use "rational interpolation scheme"), which provides linear (or quadratic) interpolation within the grid.

Examples Exponential distribution:

$$p(x) = \mu e^{-\mu x} \tag{17}$$

$t = c(x) = 1 - e^{-\mu x}$ and the inversion gives $x = -\frac{1}{\mu}\ln(1-t)$. Can be also written as $x = -\frac{1}{\mu}\ln t$ (if t is uniformly random between [0,1], also $1-t$ is such).

Wentzel distribution (scattering of charged particles)

$$p(x) = \frac{A(A+1)}{(A+x)^2}, \quad A > 0, \quad 0 \le x \le 1 \tag{18}$$

Sampling equation is

$$t = A(A+1)\left[\frac{1}{A} - \frac{1}{A+x}\right] \tag{19}$$

which yields $x = \frac{At}{A+1-t}$.

Discrete functions Can be sampled by using the inverse transform method, by mapping the cumulative distribution (which is a stepwise function), $c_0(=0), c_1, c_2 \ldots c_N(=1)$. Deliver i if $c_i < t \le c_{i+1}$. To avoid a long sequence of comparisons, better to use the *bipartition method.* This is a binary search using two indices, starting from $i = 1$ and $j = N+1$, $k = (i+j)/2$.

Rejection method Due to von Neumann. Find a pdf $\pi(x)$ (which you can "easily" sample from, by inversion method), such that $C\pi(x) \ge p(x)$ for any x (with C being a *positive constant*).
Notice: due to the normalization, $\pi(x) \ge p(x)$ cannot hold for all x, if $p(x)$ and $\pi(x)$ are true PDFs. So sample x from $C\pi(x)$ and calculate $y = C\pi(x)$. Then sample a *new* random number ξ between 0 and y. If $\xi < p(x)$, then accept x, otherwise reject x. After this procedure the x passing the procedure is distributed according to $p(x)$.

Math: The first sampling produces x-values in $(x, x+dx)$ with probability $\pi(x)dx$, and these values are accepted with probability $r(x) = \frac{p(x)}{C\pi(x)}$. So, the probability of delivering x-values in $(x, x+dx)$ is distributed as $p(x)$ apart from normalization constants.
The efficiency of the algorithm is the ratio of the areas below the curves. In fact, $1/C$ is the average number of random numbers to be sampled for each delivery. Maximal efficiency for $p(x) = \pi(x)$ (= we can sample easily by $p(x)$). Main advantage: can sample by any arbitrary pdf.
How to choose π: fast sampling (easy to invert). High efficiency is desirable but not decisive: easiness of sampling from $\pi(x)$ can compensate

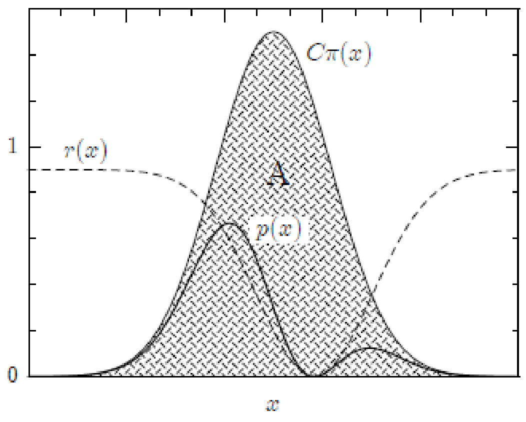

Figure 3: Random sampling from a distribution $p(x)$ using a rejection method.

for the lower efficiency. Generally speaking, a constant function $\pi(x) = \max[p(x)] = k$ is (almost) always a good test function, but it could be very inefficient.

Mixed method. If $p(x)$ can be factorized as

$$p(x) = f(x)g(x) \tag{20}$$

in which $f(x)$ is easily invertible, one can: sample x according to $f(x)$ and use the rejection technique on $g(x)$, i.e. keep the number if $\xi g_{max} < g(x)$, where g_{max} is the maximum of $g(x)$ over the range. This is basically equivalent to a change of variable $u = c(x) = \int_{-\infty}^{x} f(x')dx'$, such that

$$p(u) = f(u)g(u)\frac{dx}{du} = f(u)g(u)\frac{1}{f(u)} = g(u) \tag{21}$$

Two-dimensional distribution $p(x, y)$ can always be factorized as

$$p(x, y) = m(x)p(y|x) \tag{22}$$

One can sample x from $m(x)$ and y from $p(y|x)$. True also for any number of dimensions. Since $p(x, y) = m(y)p(x|y)$, the sampled values of y are distributed according to $m(y)$. A N-dimensional problem can always be reduced to individual sampling of N variables, by using marginalized and conditional pdfs. Composition method applicable when $p(x)$ is the product of several pdfs, i.e.

$$p(x) = \int w(y)p_y(x)dy \tag{23}$$

or

$$p(x) = \sum_y w_y p_y(x) \tag{24}$$

First, sample y from $w(y)$ and then sample x according to $p_y(x)$ chosen on the sampled y. This will be distributed according to $p(x)$.

Examples Sample a isotropic direction in 3D (i.e. sample a point on the surface of a unit sphere): all solid angles are equiprobable, so

$$p(\Omega) = \frac{1}{4\pi} \tag{25}$$

(the factor $\frac{1}{4\pi}$ guarantees the normalization). The probability dp for the solid angle $d\Omega$ is hence

$$p(\Omega)d\Omega = \frac{1}{4\pi}\sin\theta d\theta d\phi = \frac{1}{4\pi}d(\cos\theta)d\phi \tag{26}$$

that can be factorized in θ and ϕ separately as

$$p(\Omega)d\Omega = \frac{d\phi}{2\pi}\cdot\frac{1}{2}sin\theta d\theta = \frac{d\phi}{2\pi}\cdot\frac{1}{2}d(\cos\theta). \tag{27}$$

Therefore, $\cos\theta$ is sampled from a uniform distribution in $[-1,1]$ and ϕ is sampled from a uniform distribution in $[0,2\pi]$. Notice that this does *not* correspond to sample θ uniformly in $[0,\pi]$. The direction cosines can be expressed as

$$(u,v,w) = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta) \tag{28}$$

Sample from a circle: all surface elements have the same probability, to the distribution is $p(S) = \frac{1}{\pi R^2}$. In polar coordinates $dS = rdrd\theta$ and distribution can be factorized

$$p(S)dS \to p(r,\theta)drd\theta = p(r)dr\cdot p(\theta)d\theta = \frac{1}{\pi R^2}rdrd\theta \tag{29}$$

(the coefficient ensures the normalization to 1). Sample θ from $p(\theta) = \frac{d\theta}{2\pi}$, i.e. from a uniform distribution in $[0,2\pi]$. Sample r from $p(r) = \frac{2}{R^2}r$. Cumulative: $c(r) = \frac{r^2}{R^2}$. Inversion: $r = R\sqrt{\xi}$. Once sampled r and θ, $x = r\cos\theta$ and $y = r\sin\theta$.
The sampling can also be done by rejection method: sample a and b uniformly between -1 and 1. If $(a^2+b^2)\leq 1$, deliver $x = Ra$ and $y = Rb$, otherwise try again. The efficiency of this algorithm is $\frac{\pi}{4} = 0.785$ (while the direct method is 1), but this might be faster because it does not involve the sqrt and the calculation of trigonometric functions.

Sample from a Gaussian distribution: Box-Müller method. Gaussian is not integrable analytically, difficult to invert, $p_G(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}$.

Suppose to sample two independent random variables according to a normal Gauss distribution. They identify a point in the 2D space and the joint pdf is the product of the two 1D Gaussians:

$$p_{2G}(x_1, x_2) = p_G(x_1)p_G(x_2) = \frac{1}{2\pi}e^{-(x_1^2+x_2^2)/2}. \tag{30}$$

Now switch to polar coordinates: $x_1 = r\cos\theta$, $x_2 = r\sin\theta$, $dx_1dx_2 = rdrd\theta$. So

$$p_{2G}(r,\theta)drd\theta = \frac{d\theta}{2\pi}e^{-r^2/2}rdr \tag{31}$$

Now r and θ are independent random variables. θ is uniformly distributed in $[0, 2\pi]$. The pdf of r can be integrated and inverted $p(r) = re^{-r^2/2}$, $c(r) = 1 - e^{-r^2/2}$. This can be inverted as

$$r = \sqrt{-2\log(1-\xi)} = \sqrt{-2\log\xi} \tag{32}$$

So, sampling two uniform random numbers ξ_1 and ξ_2, one can produce two gaussian numbers: $x_1 = r\cos\theta = \sqrt{-2\log\xi_1}\cos(2\pi\xi_2)$
$x_2 = r\sin\theta = \sqrt{-2\log\xi_1}\sin(2\pi\xi_2)$
The procedure is exact and has 100% efficiency. For μ, σ, one just scales as $X \to \mu + x\sigma$.

Two-dim: $p(x,y)dxdy = (x+y)dxdy$ (x and y in [0,1]). Marginalized in x is $m(x) = x + \frac{1}{2}$ (can be integrated and inverted). The cumulative is $c(x) = \frac{1}{2}(x^2 + x)$, which can be inverted (second order equation). Now $p(y|x) = \frac{p(x,y)}{m(x)} = \frac{x+y}{x+\frac{1}{2}}$ (can also be integrated in dy and inverted: in this case x is a parameter).

3 Error estimation

"If you compute long enough, you should converge to the expected mean" (Law of Large Numbers): $\bar{x} \to \langle x \rangle$ ($\bar{x}$ is a random variable and it is a consistent estimator of $\langle x \rangle$ in the large-sample limit). The tally must make sense, i.e. the mean $\langle x \rangle$ of the distribution under study must exist. No guarantee that the convergence is "nice" and well-behaved.
If the variance exists, the Central Limit Theorem holds, and the width of the distribution of the mean is $\sigma/\sqrt{N}$ (and $\langle \bar{x} \rangle = \langle x \rangle$). The estimate of the mean is Gaussian-distributed (irrespectively of the shape of the parent distribution), provided the second moment exists.
If the third moment exists, there is a rule-of-thumb:

$$N \gg \sigma^6/\langle x^3 \rangle^2 \tag{33}$$

Usual recipe: score x and x^2 (better to score them separately) to estimate $\bar{x}$ and s^2

$$\bar{x} = \frac{1}{N}\sum_i x_i$$

$$s^2 = \frac{1}{N-1}\sum_i (x_i - \bar{x})^2 = \frac{1}{N-1}\sum_i x_i^2 - \bar{x}^2$$

$$s_{\bar{x}}^2 = \frac{s^2}{N}$$

Recipe #2: if not feasible, split N in n batches (e.g. $n = 30$) and apply the first recipe to the batches: the histories are not scored individually, but blocks of events are. Batches should be statistically independent.

To combine independent runs (not suggested). Better to do a global statistical analysis. However, simple mean of means and sum in quadrature of variances.

$$\bar{x} = \sum_k \frac{N_k}{N}\bar{x}_k \tag{34}$$

$$s_{\bar{x}}^2 = \sum_k \left(\frac{N_k}{N}\right)^2 s_{\bar{x}_k}^2$$

For scoring of binary quantities (yes/no), the binomial distribution has to be used and

$$s_{\bar{x}}^2 = \frac{\bar{x}(1-\bar{x})}{N-1}. \tag{35}$$

4 Monte Carlo integration

In general, solve

$$I = \int_{\mathcal{D}} H(u)du \tag{36}$$

For a 1-dim function $f(x)$ the numerical integration can be carried out with trivial methods (rectangles, trapezoids, Cavalieri-Simpson). An exact expansion would use Taylor's series, by dividing the intervals in N cells of width $\Delta u = (b-a)/N$ and center u_i

$$I = \int_a^b duH(u) = \sum_{i=1}^{N}\int_{u_i-\Delta u/2}^{u_i+\Delta u/2} H(u)du \tag{37}$$

From the Taylor's series

$$H(u) = H(u_i) + (u-u_i)\frac{dH(u)}{du}\Big|_{u_i} + \frac{(u-u_i)^2}{2}\frac{d^2H(u)}{du^2}\Big|_{u_i} + \ldots \tag{38}$$

which gives

$$I = \Delta u \sum_{i=1}^{N} \left(H(u_i) + \frac{(\Delta u)^2}{24} \frac{d^2 H(u)}{du^2}\Big|_{u_i} \right) \tag{39}$$

which is an other flavor of the Cavalieri-Simpson method. The first-order term cancels out because the integrand $H'(u_i)(u - u_i)$ is anti-symmetric in the integration range $u_i - \Delta u/2 \to u_i + \Delta u/2$. The second order term was calculated by keeping in mind that

$$\int_{u_i-\Delta u/2}^{u_i+\Delta u/2} \frac{H''(u_i)}{2}(u-u_i)^2 du = \frac{H''(u_i)}{2} \int_{u_i-\Delta u/2}^{u_i+\Delta u/2} (u-u_i)^2 du = \frac{H''(u_i)}{2} \frac{1}{12}(\Delta u)^2 \tag{40}$$

(integral = variance of a uniform distribution). The second-order term can be taken as an estimate of the error (ΔI). The relative precision is

$$\frac{\Delta I}{I} \propto (\Delta u)^2 \propto \frac{1}{N^2} \tag{41}$$

i.e. it scales quadratically with $1/N$.

Can also be done by Monte Carlo: take the function $f(x)$ within a box of basis ΔX (= integration range) and height ΔY (from the 0 to the max of the function within the range). Take for simplicity a positive $f(x)$, but this can be overcome by a simple shift of the axis. Shoot N points uniformly in $\Delta X \Delta Y$ and check how many points n fall below $f(x)$, i.e. $y < f(x)$. The area is simply

$$\int_{\Delta X} f(x)dx = \frac{n}{N} \Delta X \Delta Y \tag{42}$$

Even better, one can sample points uniformly in ΔX (or the integration range Ω for multi-dimensional space) and ("theorem of integral average")

$$I = V_\Omega \langle f \rangle_\Omega . \tag{43}$$

The mean $\langle f \rangle_\Omega$ of the function can be estimated statistically using the consistent estimator $\bar{f}$ (Law of Large Numbers), so

$$I \sim V_\Omega \bar{f} = V_\Omega \cdot \frac{1}{N} \sum_{i=1}^{N} f(x_i). \tag{44}$$

More generally, the integral can be seen as an expectation value

$$I = \int f(x) = \int p(x) g(x) = \langle g \rangle \tag{45}$$

where $p(x)$ is an arbitrary pdf and $g(x) = f(x)/p(x)$. If one samples x according to $p(x)$, the integral is just the average of $g(x)$. Each possible choice of $p(x)$ defines a different Monte Carlo algorithm to calculate the integral. One can simply take $p(x)$ as a uniform distribution $p(x) = 1/(b-a)$ (which is equivalent to the equation above). This is a "crude" Monte Carlo, which is not necessarily optimized. The variance-reduction techniques aim to find an appropriate $p(x)$ to optimize the calculation (i.e. minimize the overall calculation time for a given precision).

An advantage of the Monte Carlo approach is that it is possible to evaluate the (statistical) uncertainties on the estimate, using the Central Limit Theorem.

In 1D Monte Carlo integration is not a big gain with respect to numerical integration. In many dimensions it becomes more and more convenient. Now in 2D one has to create a square-mesh of $\sqrt{N} \times \sqrt{N}$ points. The mid-point summation yields:

$$I = \int du_1 du_2 H(u_1, u_2) = \sum_{i=1}^{\sqrt{N}} \sum_{j=1}^{\sqrt{N}} \int_{u_{i1}-\Delta u_1/2}^{u_{i1}+\Delta u_1/2} \int_{u_{j2}-\Delta u_2/2}^{u_{j2}+\Delta u_2/2} H(u_1, u_2) \tag{46}$$

Still can be done by using the Taylor expansion of $H(u_1, u_2)$:

$$\begin{aligned} H(u_1, u_2) = H(u_{i1}, u_{i2}) + (u_1 - u_{i1})\partial H(u_{i1}, u_{i2})/\partial u_1 + & \quad (47)\\ (u_2 - u_{i2})\partial H(u_{i1}, u_{i2})/\partial u_2 + & \quad (48)\\ \frac{(u_1 - u_{i1})^2}{2}\partial^2 H(u_{i1}, u_{i2})/\partial u_1^2 + & \quad (49)\\ \frac{(u_2 - u_{i2})^2}{2}\partial^2 H(u_{i1}, u_{i2})/\partial u_2^2 + & \quad (50)\\ (u_1 - u_{i1})(u_2 - u_{i2})\partial^2 H(u_{i1}, u_{i2})/\partial u_1 \partial u_2 & \quad (51) \end{aligned}$$

All linear and bi-linear terms vanish by symmetry [the factors are anti-symmetric with respect to the center of the cell], so only the quadratic terms on u_1 and u_2 survive. This is also true for more dimensions. The convergence of the integral (i.e. the relative error $\Delta I/I$, estimated as the ratio between the second-order term and the central value) is

$$\frac{\Delta I}{I} \propto \frac{1}{N^{2/\mathcal{D}}} \tag{52}$$

i.e. it gets *slower and slower* for increasing $\mathcal{D}$. In fact,

$$\frac{\Delta I}{I} \propto (\Delta u_1)^2 \propto (1/\sqrt[\mathcal{D}]{N})^2.$$

In 1D the convergence goes as $1/N^2$ and in two-dim goes as $1/N$. The central-value (= zeroth-order) calculation (i.e. use only the centers of the N cells) is equivalent to the rectangle method.

The convergence of the Monte Carlo integration goes with the Central Limit Theorem, i.e. with $1/\sqrt{N}$. Suppose that the time to process N Monte Carlo histories is $\alpha_{MC} N_h$ and the time to process N_c cells in numerical integration in $\mathcal{D}$ dimensions is $\alpha_{NMC} N_c$ (notice: the numerical integration might involve many matrix calculations, so it is not obvious that $\alpha_{MC} \gg \alpha_{NMC}$). Furthermore, the calculation N times of the integrand function $H(u)$ takes the same time in MC and numerical modes, so one can expect that α_{MC}/α_{NMC} is not too far from unity.

The ratio between the convergence rates vs. time t is

$$\frac{MC}{NMC} \propto \left(\frac{\alpha_{NMC}}{\alpha_{MC}}\right)^{1/2} t^{(4-\mathcal{D})/2\mathcal{D}} \tag{53}$$

i.e. Monte Carlo is more efficient for $\mathcal{D} > 4$. The ratio depends on $\frac{\alpha_{NMC}}{\alpha_{MC}}$. Generally, Monte Carlo methods are favored for rapidly-varying functions, while the non-MC methods are preferable for low-dimensional problems with flat/smooth functions. One could use a more accurate numerical integration (e.g. Cavalieri-Simpson with respect to the trapezoid), and MC is favored for $\mathcal{D} > 8$. However, high-order numerical quadrature formulas involve high-order derivatives and may have precision problems: if the function $H(u)$ cannot be calculated *exactly* at any point u (but e.g. has to be done by interpolation), the derivatives that are used in numerical integration might be screwed up by fake correlations (Monte Carlo approach does not use any derivative). Low-order quadrature formulas (i.e. trapezoids) are favored with respect to high-order (e.g. Cavalieri-Simpson).

5 Random numbers and precision problems

Estimate π by sampling random points (x, y) over a circle. Plotting the difference $(\bar{x} - \pi)$ vs. N. For large N one starts to see the periodicity of the random generator. An other clue of this behavior could be a "too good" convergence (i.e. always well above the 1 σ limit). False convergence.

Also possible problems due to numerical rounding, for instance $\sum_{i=1}^{N} \frac{1}{N}$ is obviously 1 mathematically. Numerically, one starts see deviations if using single-precision (round-off error).

6 Particle transport

In Monte Carlo codes of radiation transport, the history (track) of a particle is described as a *random sequence of free flights* (called "steps") that ends with an “interaction” event. In an interaction the particle changes its direction of movement, loses energy and occasionally produces secondary particles. Need an “interaction model” which contains the physics information, i.e. a set of (differential) cross sections required to decide which interaction takes place and how the final state is generated. Position and momentum are considered as independent variables in the Monte Carlo tracking, i.e. the quantum correlations are disregarded. This is valid only if the momentum is much larger than the uncertainty Δp from the Heisenberg principle.

6.1 Rotation

All interactions (= production of the final state) are sampled in a specific frame, where the particle travels in the positive z (or x) direction. The final state has to be transformed back into the original frame: need a rotation. Does not need a boost, because the energy of the particle is retained (the frames are not in relative motion: only the orientation of the axis and the position of the origin changes). Particle identified by position (3 coordinates) and direction (direction cosines, or Euler's angles): $\vec{x}$, $\vec{u}$. One can also use polar's angles:

$$\vec{u} = (u, v, w) = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta). \tag{54}$$

If $\vec{u}_0$ is the unit vector of the initial velocity, displacement in a step:

$$\vec{x} = \vec{x}_0 + \vec{u}_0 s \tag{55}$$

The calculation of the displacement s can be complicated if there is multiple scattering (e.g. charged particle) or a deflection by E or B fields.

Rotation is more complicated. The rotation matrix in general has the form

$$R(\theta,\phi) = \begin{pmatrix} \cos\theta\cos\phi & -\sin\theta & \sin\theta\cos\phi \\ \cos\theta\sin\phi & \cos\phi & \sin\theta\sin\phi \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{56}$$

so that

$$\vec{u}_{global} = R(\theta,\phi)\vec{u}_{local} \tag{57}$$

$R(\theta,\phi)$ is a orthogonal matrix, and its inverse corresponds to its transpose:

$$R^{-1}(\theta,\phi) = R^{T}(\theta,\phi) \tag{58}$$

Inverse matrices are the transformations to pass from the global coordinate frame to the local coordinate frame (which is the simplest one, used for tracking).
If a particle is scattered by Θ and Φ in the local system:

$$\vec{u} = R(\theta_0,\phi_0)R(\Theta,\Phi)R^{-1}(\theta_0,\phi_0)\vec{u}_0 \tag{59}$$

The term $R^{-1}(\theta_0, \phi_0)$ transforms the system into the one with direction along z, and $R(\theta_0, \phi_0)$ is the back-transformation to the global system. Notice that - by construction -

$$R^{-1}(\theta_0, \phi_0)\vec{u}_0 = \hat{k} = (0, 0, 1) \tag{60}$$

as the rotation $R^{-1}(\theta_0, \phi_0)$ brings the initial direction $\vec{u}_0$ to coincide with the z axis.
Numerically, the module of $\vec{u}$ must be checked and re-normalized to 1 periodically during the simulation, since it tends to move from 1 due to the numerical precision after many loops and rotations.

At the end, put rotation and translation all together. They *do not commute* in 3D, so the ordering is: displacement, rotation.

6.2 Transport in media

The job of a tracking simulation is to provide a set of displacements $s_1, s_2, \dots$ and a set of scattering angles $(\Theta_1, \Phi_1), (\Theta_2, \Phi_2) \dots$.

Interaction probability Take a semi-infinite medium (so to avoid complications with geometries and boundaries). The pdf of the distance z traveled between two following interactions $p(z)$ can be determined as follows. Firstly, let define the probability $F(z)$ to survive up to z, i.e. the probability that the particle does *not* interact between 0 and z:

$$F(z) = 1 - \int_0^z p(z')dz' = \int_z^\infty p(z')dz'. \tag{61}$$

Notice that the cumulative function is defined as

$$c(z) = \int_0^z p(z')dz' \tag{62}$$

so one has

$$F(z) = 1 - c(z). \tag{63}$$

The variation dF in the probability to survive after $z + dz$ is proportional to the probability to survive up to z, and decreases according to the probability $\mu(z)dz$ of interaction in $(z, z + dz)$, i.e.

$$dF(z) = -\mu(z)F(z)dz. \tag{64}$$

Interaction coefficient μ (L^{-1}) describes the change of the survival probability, and in general it is a function of the coordinates. Minus sign: the survival probability decreases with increasing z. Multiplication of $F(z)$: the rate is proportional to the number of surviving particles. Taking $F(0) = 1$, one can integrate:

$$F(z) = \exp(-\int_0^z \mu(z')dz'). \tag{65}$$

This solution is valid in general, for any arbitrary $\mu(z)$ depending on the z coordinate. The cumulative probability to interact within the distance z is $c(z) = 1 - F(z)$. The pdf $p(z)$ is hence

$$p(z) = \frac{d}{dz}c(z) = \mu(z)F(z) = \mu(z)\exp(-\int_0^z \mu(z')dz') \tag{66}$$

Notice: μ can be the combination of many interaction channels. One makes the assumption that the interaction channel which is chosen does not depend on what happened to the particle throughout the way (= the particle is memory-less).

Alternatively, the probability $p(z)dz$ to interact between z and $z + dz$ is the product of the probability $F(z)$ to survive up to z and the probability $\mu(z)dz$ to interact afterwards in dz. One has the differential equation

$$p(z)dz = F(z)\mu(z)dz$$

which - upon replacement of $F(z)$ - becomes:

$$p(z) = \mu(z)\int_z^\infty p(z')dz'. \tag{67}$$

The general solution with the boundary condition $p(\infty) = 0$ is

$$p(z) = \mu(z)\exp(-\int_0^z \mu(z')dz'), \tag{68}$$

as before.

If the medium is homogeneous and infinite, $\mu(z)$ is a constant,

$$\begin{aligned} p(z) &= \mu e^{-\mu z} \\ c(z) &= \int_0^z p(z')dz' = 1 - e^{-\mu z} \\ F(z) &= 1 - c(z) = e^{-\mu z} \end{aligned} \tag{69}$$

For finite medium $c(\infty) = 1 - p_s < 1$, because the particle can escape! Problem with the definition of pdf. Recover the proper normalization by adding a boundary at $z = z_b$: this is a θ-step, $\theta(z - z_b)$, in the cumulative distribution

$$c(z) = 1 - e^{-\mu z} + e^{-\mu z_b}\theta(z - z_b) \tag{70}$$

(for $\mu(z) = 0$ with $z > z_b$). The boundary behaves as a "black box" and absorbs all impinging particles (they are not tracked any longer). When taking the survival pdf, the θ-step becomes a δ function:

$$p(z) = \frac{d}{dz}c(z) = \mu e^{-\mu z} + e^{-\mu z_b}\delta(z - z_b) \tag{71}$$

Layered geometry Layered geometry with $\mu_1, \mu_2, \ldots \mu_N$ at the boundaries $b_1, b_2 \ldots b_N$. One can use conditional probabilities: $p(b_i|b_{i-1})$, i.e. the probability to survive to region i, given that it did not interact in region $i-1$. The path length in each region can be treated as *independent*, i.e. the tracking does not depend on the past story of the track (Markov process). For instance, two homogeneous regions:

$$\mu(z) = \theta(b - z)\mu_1 + \theta(z - b)\mu_2 \tag{72}$$

The general solution

$$p(z) = \mu(z)\exp(-\int_0^z \mu(z')dz') \tag{73}$$

still holds, and the pdf can be written as

$$p(z) = \theta(b - z)\mu_1 e^{-\mu_1 z} + \theta(z - b)\mu_2 e^{-\mu_1 b} e^{-\mu_2(z-b)} \tag{74}$$

or

$$p(z) = \begin{cases} \mu_1 e^{-\mu_1 z} & \text{if } z < b \\ \mu_2 e^{-\mu_1 b} e^{-\mu_2(z-b)} & \text{if } z > b \end{cases} \tag{75}$$

Notice that $p(z)$ is a well-behaved pdf, as $\int_0^\infty p(z)dz = 1$. Also, $e^{-\mu_1 b}$ is the probability that the particle survives the first layer and gets into the second. One can take the step length z_1 and z_2 in the two layers as *independent* random variables, i.e. sample z_1 according to the (truncated and re-normalized) pdf

$$p(z_1) = \mu_1 e^{-\mu_1 z_1} + e^{-\mu_1 b}\delta(z - b) \tag{76}$$

which includes also the boundary. If the particle survives the first layer (i.e. the sampling gives $z_1 = b$), it enters the second layer and one can sample z_2 (i.e. the distance traveled in the second layer, $z_2 = z - b$) from

$$p(z_2) = \mu_2 e^{-\mu_2 z_2} \tag{77}$$

Here z_1 and z_2 are independent. This is because the z_2 distribution does not depend on z_1 (the coefficient μ_1 appears only as a multiplicative constant). Once you pass from the first to the second layer, there is no memory of the fact that the particle was in the first layer. The argument applies for any arbitrary number of homogeneous layers, with different $\mu_1 \ldots \mu_N$. The paths traveled in each layer can be always seen as independent random variables.

In summary: the interaction of a particle depends only on the *local* scattering conditions. We can always sub-divide the space in regions with uniform interaction length. *If we go through the boundary, we stop at that boundary and re-sample according to the properties of the new medium.* This assumes that the boundaries are "sharp" (no transition) and passive (no transition radiation, plasmon excitation, etc.). The path length between the last real interaction in the first medium and the first real interaction in the second medium is $1/\mu_1 + 1/\mu_2$ (sic!). This can also be verified by taking a virtual boundary within a homogeneous medium: for the tracks which *cross the boundary* the average distance is $2/\mu$. Due to the Markov character of the process: the average distance between each interaction point and the (fixed) boundary is $1/\mu$, so the average distance between the two points before and after the crossing is $2/\mu$. One can stop the simulation at the boundary (fixed point in the track) and resume it later (the key here is the condition "which cross the boundary" so restricting the average only to a right-handed tail of the distribution). It is just like the radioactive decay law: if you shift the time origin, the distribution is always an exponential with the same τ. *Future does not depend on the past history.*

Path length and physical parameters The path length μ is related to the cross section σ. The cross section is such that

$$dp(z) = N\sigma dz \tag{78}$$

with N being the volume density of targets. Generalization of the classical "billiard ball" picture, with σ being the cross-sectional area of the scattering center. However, in the real life σ is a function of energy. Other problems: when the scattering centers are tightly packed (= very high density) or there are collective motions (= coherent scattering from several centres). If these cases are not considered:

$$\mu = N\sigma \tag{79}$$

and $1/\mu$ is the mean free path. The total cross section $\sigma(E)$ might be differential in energy and angle (i.e. the outgoing particle is not emitted isotropically and it has a different energy than the incoming one).

$$\mu(E, E', \Theta, \Phi) = \mu(E)p(E, E', \Theta, \Phi) \tag{80}$$

where $p(E, E', \Theta, \Phi)$ is a normalized pdf. In summary, only the integral cross section matters for μ, while the other dependencies are necessary to sample the final state.

Recipe for particle transport:

1. calculate the interaction coefficient as $\mu(E) = N\sigma(E)$ (if necessary, integrate $\sigma(E)$ from its differentials)
2. number density N can be derived by mass density ρ as $N = \frac{\rho}{A} N_A$ with A atomic weight and N_A Avogadro's number.
3. if $\mu(E)$ does not depend on the position, sample the step length (length of the free flight) from the exponential $s = -\frac{1}{\mu(E)} \log \xi$
4. transport the particle through the new position $\vec{x} = \vec{x}_0 + \vec{u}_0 s$
5. get the marginal probability distribution $d\sigma/dE'(E, E')$ (= integrate over angles) and sample the final energy E'.
6. get the conditional probability $p(E, \Theta, \Phi | E')$ given the energy E' and sample the scattering angles Θ and Φ.
7. rotate to get the new direction after the sampling.

In the case of compounds, one calculates μ as the weighted sum of the partial cross sections

$$\mu = \sum_i N_i \sigma_i \tag{81}$$

Since usually components are specified by fraction in mass (and not fraction in number of atoms) one has to retrieve N_i starting from mass density

$$\frac{\mu}{\rho} = \sum_i w_i \frac{\mu_i}{\rho_i} \tag{82}$$

(Notice: $\frac{\mu}{\rho}$ is often called “mass attenuation coefficient” and is tabulated. It is the inverse of the mean free path in g·cm^2.)

Sometimes $\mu(E)$ is the combination of many *competing* processes that can take place alternatively.

$$\mu = \sum_i \mu_i(E) \tag{83}$$

The fractional probability for each interaction channel is μ_i/μ, i.e. depends only on the attenuation coefficients (or on the cross sections), does not depend on *how* the particle arrived to that point. Discrete probability distribution. Need an other step to the recipe, between 4 and 5 (the sampling of s is unaffected), i.e. choose the interaction which the particle undergoes before sampling the final state. Notice that all processes contribute to μ and hence to the definition of the average path length s.

6.3 Markov chains

The fact that the propagation of a particle does not depend on the previous steps (i.e. on *how* the particle arrived to that point/state) means that the tracking Monte Carlo is a Markov process. "Future values of a random variable (e.g. interaction event) are statistically determined by present events and depend only on the event immediately preceding". One can stop the simulation at any stage and then resume it later without any bias or loss.

From Wikipedia:
A Markov chain named after Andrey Markov, is a mathematical system that undergoes transitions from one state to another, among a finite or countable number of possible states. It is a random process usually characterized as *memoryless*: the next state depends only on the current state and not on the sequence of events that preceded it. Transition matrix which describes the evolution of the system, i.e. $x_n = Mx_{n-1}$.

6.4 Examples

Isotropic scattering (e.g. low-energy neutrons, in the center of mass)

$$p(\Theta, \Phi) = \frac{1}{4\pi} \sin\Theta d\Theta d\Phi \tag{84}$$

Sampling: $\cos\Theta = -1 + 2\xi_1$, $\Phi = 2\pi\xi_2$.

P_1-wave (semi-isotropic):

$$p(\Theta, \Phi) = \frac{1}{4\pi}(1 + a\cos\Theta) \sin\Theta d\Theta d\Phi \tag{85}$$

with $|a| < 1$. The procedure is analytical, but a bit more complicated

$$\cos\Theta = \frac{2 - a - 4\xi_1}{1 + \sqrt{1 - a(2 - a - 4\xi_1)}} \tag{86}$$

and $\Phi = 2\pi\xi_2$.

Rutherford scattering (elastic scattering of a charged particle)

$$p(\Theta, \Phi) = \frac{a(2+a)}{4\pi} \frac{\sin\Theta d\Theta d\Phi}{(1 - \cos\Theta + a)^2} \quad 0 \le a < \infty \tag{87}$$

The sampling procedure is still analytical

$$\cos\Theta = 1 - 2a\frac{1 - \xi_1}{a + 2\xi_1} \tag{88}$$

and $\Phi = 2\pi\xi_2$. It gets back to an isotropic scattering for $a \to \infty$ and no-scattering $\delta(1-\cos\Theta)$ for $a \to 0$.

Cauchy distribution:

$$p(x) = \frac{1}{\pi}\frac{1}{1+x^2} \tag{89}$$

Cumulative function:

$$c(x) = \int_{-\infty}^{x} p(y)dy = \frac{1}{\pi}\arctan x + \frac{1}{2}$$

Inversion: as $x = \tan[\pi(\xi - \frac{1}{2})]$

6.5 A deterministic view

The particle transport and tracking might be seen from the point of view of transport equations (e.g. Maxwell-Boltzmann) in a space having 6 dimensions (position and momentum). Formally, the solution can be written by using the Green's function

$$\psi(x,p) = \int dx' \int dp' G(x,p,x',p',s)Q(x',p') \tag{90}$$

where $Q(x',p')$ is a source term and the Green's function $G(x,p,x',p',s)$ is an operation in the 6-dim space which propagates a particle from (x,p) to (x',p'). This includes the scattering and the interactions, and can be in principle calculated from the law of physics. Difficult to implement a method to integrate this equation, "phase space evolution" is iterative with the step length s. The step length has to be taken small in order to account for the inhomogeneities of the real geometry.

7 Mixed Monte Carlo

A detailed simulation (*all* interactions with *loss of energy and/or change of direction* are simulated explicitly) is affordable only if the number of steps per track is small ($<$ a few 100's). This means that it can be done at very low energy or very thin materials. Cross section for charged particles is much higher than for neutral particles. This is not a problem for γ-rays (mean free path is usually cm), but a significant issue for charged particles, which undergo ionization and bremsstrahlung. Furthermore, γ-ray interactions produce "catastrophic" interactions, i.e. there are large changes in energy/direction of the primary particle and production of energetic secondaries. The electron/positron interactions are typically soft: in most of

the cases, only a small fraction of the energy is lost at each interaction and there is a small angular displacement; low-energy secondaries are emitted. "Hard" interactions with a significant energy loss (i.e. ionization of an inner shell) or change of direction are relatively rare.
Employ *mixed* simulation scheme: simulate explicitly (i.e. force step) only if the loss of energy or the change of direction are above a given threshold. The effect of all sub-threshold interactions is described cumulatively (*condensed* simulation). Hard events occur much less frequently than soft events (= need much less steps to be produced and simulated) but have a major impact on the particle evolution.
An other possibility is a purely "condensed" simulation, in which all interactions are described cumulatively (i.e. no explicit simulation of hard events: everything is soft). Might have problems with the boundaries.

7.1 Delta interactions and soft energy losses

Notice: for electron transport things get more complicated because energy decreases along the path (mixed Monte Carlo scheme). In this case it is necessary to limit the free flight path s to a maximum value s_{max} due to the energy losses. Truncate the distribution of s: if the sampled s is greater than s_{max}, move the track by s_{max} and do nothing at the end of the step (δ interaction). This is not a bias in the simulation since physical pairs of (hard) interactions with distance $s > s_{max}$ can take place with n intermediate δ-interactions. The probability to have a δ interaction is

$$p_\delta = \int_{s_{max}}^{\infty} p(s)ds = e^{-\mu s_{max}} \tag{91}$$

If $s = ns_{max} + s'$ is made up by n δ-interactions plus one physical interaction (at $s' < s_{max}$ from the last fake interaction):

$$p(s) = p_\delta^n \mu e^{-\mu s'} = \mu e^{-\mu s} \tag{92}$$

that is the correct pdf.

Define a few parameters: mean free path between hard collisions (i.e. ionization or bremsstrahlung), above threshold W_0

$$\mu_h = N \int_{W_0}^{E} \frac{d\sigma}{dW}(E)dW. \tag{93}$$

Global stopping power due to soft collisions ($W < W_0$).

$$S_s = N \sum \int_0^{W_0} W \frac{d\sigma}{dW}(E)dW \tag{94}$$

(where the summation is over the processes involved, possibly with different cutoff energy) and straggling parameter

$$\Omega_s^2 = N \sum \int_0^{W_0} W^2 \frac{d\sigma}{dW}(E) dW. \tag{95}$$

Notice that μ_h, S_s and Ω_s^2 are all functions of the particle (initial) energy E.

One assumes that in the path s between two hard collisions, only soft collisions can take place. The energy loss in the step is w (a random variable), and the average energy loss is $\langle w \rangle = S_s s$, which must be much smaller than E. Important to know how the actual energy loss along the step w fluctuates around the mean value ("straggling"). One can calculate the second moment from Landau's theory

$$\langle w^2 \rangle = (S_s s)^2 + \Omega_s^2 s \tag{96}$$

and hence the variance

$$\mathrm{var}(w) = \langle w^2 \rangle - \langle w \rangle^2 = \Omega_s^2 s \tag{97}$$

Can be used for sampling. If the number of soft interactions in s is relatively large, the Central Limit Theorem holds, so Gaussian distribution. In some cases, the number of soft interactions is not sufficient for the Gaussian pdf: the solution of Penelope is to use an alternative pdf which has the same mean and variance (but "easy" shape): not affecting simulation results if the number of steps per track is sufficiently large. The energy "along step" can be considered as deposited locally in a random point within the step s (used e.g. for spatial dose calculations), called "hinge".

The detailed simulation is in principle possible for ionization (if the cut W_0 is smaller than the ionization potential of the most external shell, this is de-facto a detailed simulation), but not for bremsstrahlung, due to the infrared divergence of the cross section.

In some cases, the approximation $w \ll E$ does not hold: necessary to keep into account the fact that the cross section (integral and differential) changes along the step. Can expand the σ, S_s and Ω_s^2 at the first order, e.g.

$$\sigma(E_0 - w) \sim \sigma(E_0) - \left[\frac{\partial \sigma(E)}{\partial E}\right]_{E=E_0} w \tag{98}$$

Equations can be solved to calculate the corrected mean value and variance for w

$$\langle w \rangle = \frac{S_s(E_0)}{S_s'(E_0)}[1 - e^{-S_s'(E_0)s}] \sim S_s(E_0)s\left[1 - \frac{1}{2}S_s'(E_0)s\right] \tag{99}$$

(with S' = first derivative with respect to E). Similar trick to derive a corrected value of $\langle w^2 \rangle$ and of var(w).

The soft energy losses (and the energy variation along the step) affect also the total cross sections that are used for the sampling of the hard interactions (i.e. the mean free path changes along the step). Solution: *limit the average length of the step*, to make sure that the *average* energy loss along the step is small with respect to the kinetic energy at the beginning of the step. Define that you tolerate $\langle w \rangle / E < C$, with C small (e.g. 0.05). This limits the average step value. But s is sampled by a exponential distribution, so limit to s_{max}. Notice: the along-step energy loss is however applied even in δ interactions (there is the intermediate "hinge" where soft energy loss and deflection are applied). Discrete processes compete while continuous processes co-work.

7.2 Tracking loop for electrons and positrons

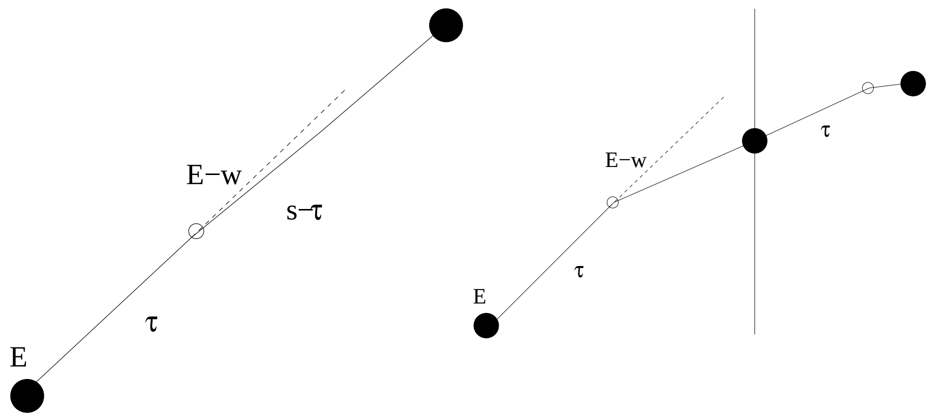

Figure 4: Left: normal step with the "hinge" in the middle. Right: step whose length is limited after the hinge by a geometry boundary.

Recipe used by Penelope for the (mixed) simulation of electrons and positrons

1. initialize the track by setting position, energy and direction
2. decide what is the maximum allowed average soft energy deposition $\langle w_{max} \rangle$ $(= C \cdot E)$ along the track and retrieve the hard attenuation coefficient μ_h for hard events. The step length can be limited by the requirement on $\langle w_{max} \rangle$ (δ interaction, in this case).
3. sample the distance s to the next hard event, by using the total hard cross section $s = -\frac{1}{\mu_h} \log \xi$. If $s > s_{max}$ the step is truncated to s_{max}.

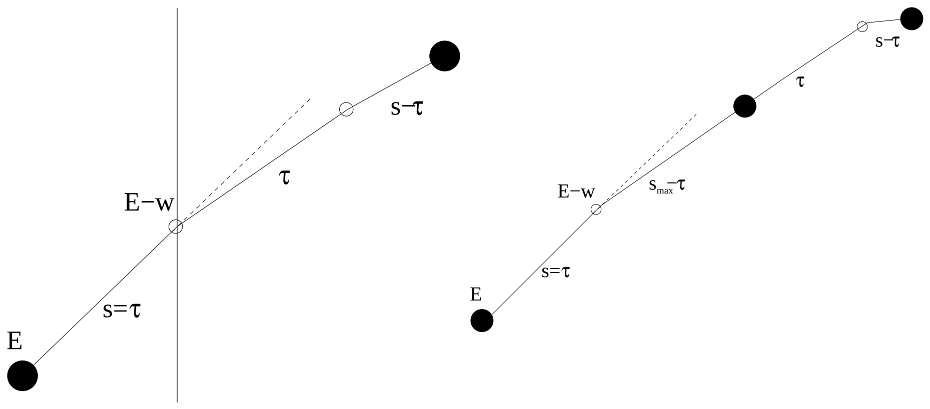

Figure 5: Left: step whose length is limited before the hinge by a geometry boundary. Right: Delta interaction, with truncation of the step length to s_{max}.

4. sample the position of the "hinge", randomly in the step. The hinge is the point where the along-step energy is released. Move the particle in that point. The hinge happens at distance $\tau = \xi s$.
5. if a boundary has been crossed, stop the step on the boundary, and redefine the hinge position ($\tau =$ traveled distance up to the boundary).
6. simulate cumulative energy loss and deflection at the hinge. Actions performed in random order. If residual energy $E - w$ is too small (below cut), stop tracking. Go to 2. for the step to the new material if the hinge is at a boundary.
7. complete the step s by completing the distance $(s - \tau)$ after the hinge (but using the new direction).
8. check again for the interface crossing (if so, stop step at the interface and redefine s as the traveled distance)
9. if $s > s_{max}$, simulate a delta interaction (i.e. do nothing) and go back to 2
10. simulate hard event: choose channel according to the cross section fractions and sample the final state. Update energy and direction. If secondary particles are created, store them in the stack
11. handle secondaries in the stack.

The tracking is stable under the variation of the simulation parameters (e.g. cuts, maximum step length, maximum allowed fractional energy loss along the step), mainly because of the corrections applied to account for the energy dependence along the step. The safe recipe for s_{max} is to have it < 0.1 of the minimum length scale which we are interested in. Much easier for γ-rays: only hard interactions can happen (no hinge, no delta interactions). The mixed simulation can be tens of times faster than the detailed simulation, while providing the same physics results. Some care has to be taken in selecting the proper values for the simulation parameters.

7.3 Effect of Multiple Coulomb Scattering (MCS)

"Along step" effects, which is a cumulative energy loss w due to soft interactions (must be $w \ll E$ to limit the dependency of the cross section along the step) and a cumulative displacement (angular displacement and lateral displacement). The effect of the MCS is also to make the trajectory not-straight: so first calculate the path length s (which is limited by the hard physics processes, or by the boundaries, or by the requirement that $\langle w \rangle \ll E$ in alongStep processes). The new position of the particle cannot simply be

$$\vec{x} = \vec{x}_0 + \vec{u}_0 s \tag{100}$$

The geometrical distance between $\vec{x}$ and $\vec{x}_0$ will be surely smaller than s, because of the deflection/folding caused by the multiple scattering.

8 Geometries and fields

Description of geometries, tracking in the presence of geometry boundaries and/or of fields.
Geometries defined as a set of volumes (with a possible hierarchy). Penelope and FLUKA: approach with quadric surfaces and planes. Geant4 has a more friendly interface with geometrical shapes and booleans. Recently, a very friendly graphical interface (FLAIR) made available for FLUKA. Interfaces between different media require the truncation of the step (a step can never cross a boundary: tracking is stopped on the interface and then resumed on the new medium).
Primary goals of the navigation:

- to know to which (unique) volume each point (x, y, z) belongs to. For a good-behaved geometry, volumes at the same level of the hierarchy cannot share a point (= no overlap). A daughter volume cannot protrude from the mother. Painstaking debugging necessary sometimes.
- to know what is the closest boundary and how far it is
- to know where is the boundary in a given direction

- to know the normal to the surface in each given point (e.g. for optical photons)
- be quick to do all that

Take care of numerical truncations (i.e. when we are very close to a surface, we might stuck on the wrong side of the interface) and introduce a tolerance.

Transport in *static* EM fields: coupled with tracking. Affect position, energy and direction of particles, so it must be integrated in the stepping. Loss/gain of energy in the case of electric fields. Changes only the description of the *free flight* part, but cross sections and interaction properties are unaffected. Usual general-purpose Monte Carlo codes are unable to treat strong fields, i.e. which accelerate electrons at rest and/or irradiate photons due to the fields. Use Lorentz's equation

$$\frac{dp}{dt} = e(E + v \times B) \tag{101}$$

Can divide the transverse and orthogonal components. Solution which is suitable for transport code: divide the trajectory in steps with nearly-constant acceleration, so to easily update velocity and position. Divide the chord in many straight steps. May have an impact on tracking (i.e. miss a volume or not). Potentially CPU-demanding.

9 Variance reduction

The statistical uncertainty can be somewhat optimized (while keeping constant the computing time) by using variance-reduction techniques. Very much problem-dependent, so it is impossible to deliver a general recipe. Importance of variance-reduction techniques should not be overemphasized: in most cases "analogue" simulations do the job within a reasonable time (if one considers the time which is necessary to invest to define, optimize and set up the variance reduction). The methods might decrease the statistical uncertainty of some observables and increase for other observables (trade-off).

Interaction forcing Useful when the process of interest has very low probability with respect to competitors. Solution: artificially increase the interaction probability of the process of interest, i.e. force the interaction to occur more frequently than in real life. Use an increased interaction coefficient $\mu_F > \mu$. Generation of the final state unchanged. To keep the simulation unbiased, it is necessary to apply a correction factor W (weight).

- Weight is 1 for primary particles and it is W/F ($< W$), with $F = \frac{\mu_F}{\mu}$ for the other secondaries produced in the forced interaction ($W < 1$).

Secondaries produced by other interactions (those not biased) are kept with the same W as the parent.

- Final state is generated to evaluate the energy loss and the possible emission of secondaries. But the parameters of the parent particles are updated only with probability $1/F$: in the other cases, a δ interaction occurs, i.e. no change of energy and direction for the projectile.
- weight W/F given to the energy deposited in forced interactions (notice: this violates energy conservation, due to fluctuations).

Splitting and Russian roulette Effective when the interest is localized in a certain region of the space (e.g. shielding problems). Favors a part of the volume with respect to the others and inhibits the radiation which leaves the volume. The region of interest can also be defined by cuts in E and direction (i.e. back-scattering from a surface: are important the particles on the impinging side of the boundary and with wrong direction). Particles start with weight $W = 1$. When they go in the "good" direction, split the particles in S particles ($S > 1$), each with weight W/S. When they leave the region of interest, kill the particles with a probability K (Russian roulette) and assign a weight $1/(1 - K) > 1$ (if survives).

Multiple secondaries When the iteraction of interest (e.g. bremsstrahlung) is sampled, generate N times the final state (instead that only once) and follow them all, each with weight $1/N$.

Other methods Simply avoid un-necessary calculations, e.g. kill particles that are not interesting and that are not contributing to the score of interest.

10 Monte Carlo codes

10.1 Zoology of Monte Carlo codes

- specialization for type of particles and/or energy range and/or physics domain. E.g. MCNP for neutrons, EGS and Penelope for EM interactions of electrons, positrons and gammas. Geant4 and FLUKA are (or, at least, aim to be) of general purpose, to cover a large set of particles, physics processes and experimental use cases.
- open source vs. libraries. Some codes are released with the source (= you can look what is inside, for instance how physics is treated, and potentially change it), some others are released only as pre-compiled libraries.

- programming language. The historical programming language for (particle) physics is FORTRAN (CERNLIB, PAW, etc.) so many Monte Carlo codes are written in this language. Starting to become a bit obsolete, and does not allow for easy "extensions" of functionalities. Geant4 is written in `C++`.

10.2 Geant4 tracking algorithm

Generic interface. Three classes of actions: at rest, along step, post step. (Along step: condensed, Post Step and At Rest: detailed). Each process can have all actions or only some of them.

1. If the particle stops (i.e. zero kinetic energy), each active atRest process proposes a step length (in time) based on the interaction it describes. And the process proposing the smallest step length (i.e. the shortest time) will be invoked. Notice: the proposed step lengths are randomized: $\frac{-1}{\mu}\log\xi$.
2. Each active discrete or continuous process must propose a step length based on the interaction it describes. The smallest of these step lengths is taken. Again, steps are randomized. Notice: a continuous process can limit the step (e.g. due to the fractional energy loss along path).
3. The geometry navigator calculates "Safety", the distance to the next volume boundary. If the minimum physical-step-length from the processes is shorter than "Safety", the physical-step-length is selected as the next step length. In this case, no further geometrical calculations will be performed.
4. If the minimum physical-step-length from the processes is longer than "Safety", the distance to the next boundary is re-calculated.
5. The smaller of the minimum physical-step-length and the geometric step length is taken.
6. *All* active continuous processes are invoked. Note that the particle's kinetic energy will be updated only after all invoked processes have completed. The change in kinetic energy will be the sum of the contributions from these processes.
7. The current track properties are updated before discrete processes are invoked. In the same time, the secondary particles created by processes are stored in SecondaryList. The updated properties are:
   - updating the kinetic energy of the current track particle
   - updating position and time

8. The kinetic energy of the particle is checked to see whether or not it has been terminated by a continuous process. If the kinetic energy goes down to zero, atRest processes will be applied at the next step, if applicable.

9. The discrete process is invoked (if it limited the step; otherwise: delta interaction). After the invocation,

   - the energy, position and time of the current track particle are updated, and
   - the secondaries are stored in SecondaryList.

10. The track is checked to see whether or not it has been terminated by the discrete process.

11. "Safety" is updated.

12. If the step was limited by the volume boundary, push the particle into the next volume.

13. Invoke the user intervention G4UserSteppingAction (user hook), Handle hit information. Save data to Trajectory.

14. Update the mean free paths of the discrete processes (new energy)

15. If the parent particle is still alive, reset the maximum interaction length of the discrete process which has occurred.

Notice: the step length s is sampled differently in Penelope and in Geant4. In Penelope, one takes the sum of the attenuation coefficients $\mu = \sum_i \mu_i$ and samples s according to the exponential $\mu e^{-\mu s}$. Then a new random number is drawn to decide which is the selected process.
In Geant4, each process proposes a step length s_i, sampled from $\mu_i e^{-\mu_i s_i}$ and s is chosen as $s = \min\{s_1 \ldots s_N\}$. In this case, the selected process is automatically determined. This is more efficient (and elegant) when there are many particles (each with its own list of physics processes) and when the number of processes may change (e.g. due to user settings).

10.3 Cuts and regions

Thresholds (detailed vs. condensed) are eventually necessary: a real-life simulation cannot be entirely detailed, except for very special cases. Question: how low should I go in energy? Trade-off: low thresholds correspond to a higher accuracy of the simulation (surely one wants to reach the desired accuracy, but do not over-do!), high thresholds to a shorter CPU time. Tracking a large number of low-energy secondary particles is the price to pay for a more precise simulation, but the question is: how "precise" should it be?

Tracking vs. production cuts. No tracking cut in Geant4 (all particles tracked down to zero energy), only *production* cut (only applies to bremsstrahlung and ionisation). Needs user explicit action to "kill" existing alive particles (e.g. thermal neutrons, etc.).

Geant4 uses a cut-per-range approach: cuts are not defined in energy, but as *cuts in range* (e.g. 1 mm). A secondary is explicitly generated (detailed simulation) only if it is able to travel for at least the specified range (energy large enough), otherwise it is considered as a local energy deposit. It is generated anyway if it could leave the current volume (even if sub-threshold): the cut does not affect the precision for the volume crossing. Corresponds to a different energy cut in each material of the geometry: the production energy threshold is automatically calculated by Geant4 for all materials, given the properties of the material and the types of particle. Advantage with respect to the energy cut when one has a set-up made out of very different materials (e.g. a Pb/Ar calorimeter): if the cut is "low" in energy, the tracking in Ar is appropriate, but a lot of time is spent to calculate unnecessary secondaries in Pb. If it is large, the tracking in Pb is ok, but it is not precise in Ar. With the cut-per-range, two different energy thresholds are used in the materials. One is setting the *spatial precision* of the simulation: the tracking will not be accurate on a length scale which is shorter than the cut (local energy deposit).

Geant4 allows to set different regions, each with a different set of cuts: have a more precise tracking where it is really needed (e.g. Si tracking devices) and a faster simulations in other less-interesting parts of the space (e.g. passive volumes, calorimeters).

References

[1] F. Salvat, J.M. Fernandez-Varea and J. Sempau, "PENELOPE, a code system for Monte Carlo simulation of electron and photon transport", PENELOPE User Manual, NEA (2008)

[2] A.F. Bjelajew, "Fundamentals of the Monte Carlo method for neutral and charged particle transport" (2001)

[3] F. James, "A review of pseudorandom number generators", Comp. Phys. Comm. **60** (1990) 329

[4] D.H. Lehmer, "Mathematical methods in large-scale computing units", Proceedings of a Second Symposium on Large-Scale Digital Calculating Machinery, 141 (1949)

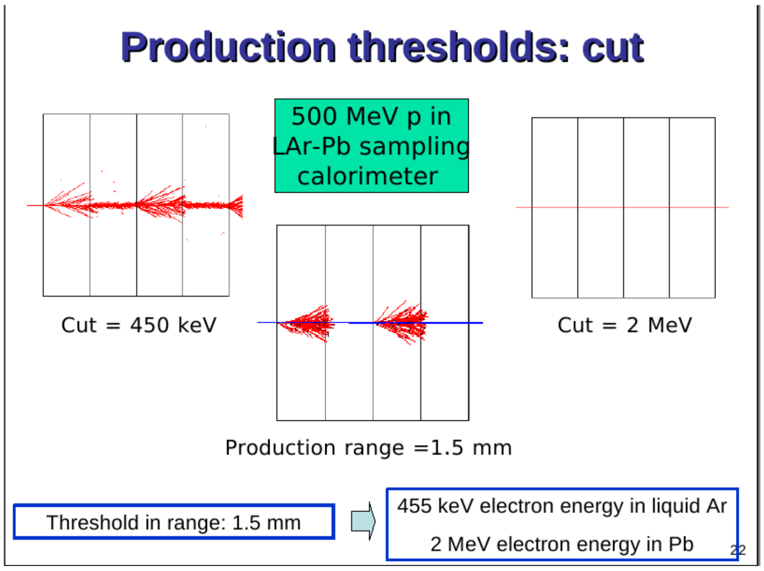
Production thresholds: cut
500 MeV p in LAr-Pb sampling calorimeter
Cut = 450 keV
Cut = 2 MeV
Production range =1.5 mm
Threshold in range: 1.5 mm
455 keV electron energy in liquid Ar
2 MeV electron energy in Pb
22

Figure 6: Example of cut-in-range

Contents