Göran Wahnström

MONTE CARLO
Lecture notes

Göteborg, 14 November 2017

The Monte Carlo method

Monte Carlo methods are a class of computer based techniques, generally based on extensive use of random number sequences. The name was invented by researchers in the 1940's working at Los Alamos and it refers to the Monte Carlo Casino in Monaco. One can distinguish between two types of problems that can be treated by Monte Carlo methods. One type is direct simulation of problems that are modelled as stochastic in nature and the other is problems that are deterministic but reformulated such that a stochastic approach can be used [1].

An example of the former type of problem is Brownian dynamics, which will be discussed later in the course. Equilibrium properties of interacting particles in classical statistical mechanics is an example of the latter type of problems. In 1953 Metropolis *et al.* [2] studied the equation of state of a system of particles treated as hard spheres in two dimensions. The obvious way to find out about the equilibrium properties is to solve for the dynamics of the system using Newton's equation of motion, and let it run until it reaches equilibrium. We call this molecular dynamics simulation. The great insight of Metropolis *et al.* [2] was that one does not need to solve for the dynamical evolution of the system, one can instead make use of a random walk process, a Markov chain, having the same equilibrium distribution.

The use of Markov chains has expanded considerably after the seminal work by Metropolis *et al.* [2] and the Markov chain Monte Carlo (MCMC) method has become an important technique in simulation, optimization and estimation [3]. For instance PageRank, the search ranking algorithm used by Google, is based on a Monte Carlo random walk technique. Monte Carlo methods can also be used to solve partial differential equations in high dimensions. Later in the course this type of application of the method will be discussed in connection to problems in quantum mechanics.

Contents

1 Probabilities

1.1 Probability functions

We start by presenting some basic concepts within probability theory. Consider a physical system. It can be in different *states*. The set of all possible states is the *sample space*. A *discrete* sample space contains either a finite or infinite number of distinct values (such as the "spin" configuration in an Ising model) while a *continuous* sample space contains an infinite number of continuous values (such as the positions of particles in a classical liquid). The result of an observation of the system, an "experiment", is called the outcome and is characterized by a single point in sample space, a *sample point*. We use upper case $(X, Y, \ldots)$ to denote a sample point.

The outcome cannot be predicted by certainty, only its probability. For continuous variables this is described by the *probability density function* $p(x)$ that gives the probability for all possible outcomes, with $p(x)dx$ equal to the probability that the outcome X is in the interval $x - dx < X \leq x$. The probability density function $p(x)$ has to be non-negative and integrate to unity,

$$\int_{-\infty}^{\infty} p(x)\,\mathrm{d}x = 1 \ . \tag{1}$$

It is also convenient to introduce the corresponding *cumulative distribution function* according to

$$F(x) = \int_{-\infty}^{x} p(x')\,\mathrm{d}x' \ , \tag{2}$$

where then $F(x)$ is equal to the probability that X will take a value less than or equal to x.

In the discrete case the corresponding probability density function is denoted *probability mass function* p_i, where p_i is the probability that the outcome is the state with index i. The values p_i have to be non-negative and normalized

$$\sum_{i=1}^{N} p_i = 1 \ , \tag{3}$$

where we assumed finite samle space with N distinct values. The corresponding cumulative distribution function is given by

$$F_i = \sum_{j=1}^{i} p_j \ . \tag{4}$$

The expected, or average, value of some arbitrary function $f(x)$ with respect to the probability density function $p(x)$ is given by

$$\langle f \rangle = \int_{-\infty}^{\infty} f(x)p(x)\,\mathrm{d}x \ . \tag{5}$$

The two most important are the *mean value*

$$\mathrm{E}[X] \equiv \mu = \langle x \rangle = \int_{-\infty}^{\infty} x p(x)\,\mathrm{d}x \tag{6}$$

and the *variance*, the second moment around the mean,

$$\mathrm{Var}[X] \equiv \sigma^2 = \left\langle (x - \langle x \rangle)^2 \right\rangle = \int_{-\infty}^{\infty} (x - \langle x \rangle)^2 p(x)\,\mathrm{d}x \ . \tag{7}$$

The square-root of the variance is the *standard deviation* σ. The variance can be evaluated from Eq. (7), which would require the pre-calculation of the mean $\langle x \rangle$. The pre-calculation can be avoided by rewriting Eq. (7) as

$$\sigma^2 = \left\langle x^2 \right\rangle - \langle x \rangle^2 = \int_{-\infty}^{\infty} x^2 p(x)\,\mathrm{d}x - \left[\int_{-\infty}^{\infty} x p(x)\,\mathrm{d}x \right]^2 \ , \tag{8}$$

which enables one to accumulate $\left\langle x^2 \right\rangle$ and $\langle x \rangle$ simultaneously during a numerical simulation. Corresponding expressions can be derived for the discrete case.

Some useful probability distributions are the *uniform* $p_u(x)$, the *exponential* $p_e(x)$ and the *Gaussian* $p_g(x)$ distributions. The uniform distribution is given by

$$p_u(x) = \begin{cases} 1/(b-a) & a \le x \le b \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

with the mean value $\mu = (a+b)/2$ and variance $\sigma^2 = (b-a)^2/12$, and the exponential by

$$p_e(x) = \begin{cases} \lambda \exp(-\lambda x) & x \ge 0 \\ 0 & x < 0 \end{cases} \tag{10}$$

with the mean value $\mu = 1/\lambda$ and variance $\sigma^2 = 1/\lambda^2$. The Gaussian distribution with mean value μ and variance σ^2 is given by

$$p_g(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2} \right] . \tag{11}$$

1.2 Central limit theorem

Consider now the situation where we have performed a set of N experimental observations or numerical simulations of a system. We denote the sequence of outcomes by $(X_1, X_2, \ldots, X_N)$ and we assume that they are *independent*. They are distributed according to the (unknown) probability density function $p(x)$ with the (unknown) mean value μ and (unknown) variance σ^2. The standard *estimate* of the mean value of X is the sum

$$S(X_1, \ldots, X_N) \equiv \frac{1}{N} \sum_{i=1}^{N} X_i \ , \tag{12}$$

but only in the limit $N \to \infty$ the true mean value μ is obtained. For a finite value of N we only get an approximate value. However, we would like to obtain some estimate of the error. This can be derived using the *central limit theorem*. Suppose that we continue the experiment until we have obtained M independent sums, S_1 to S_M, each composed of N samples. The set of sums $\{S_i\}$ are M new random variables. According to the central limit theorem these are distributed according to a Gaussian probability function, regardless of the underlying probability function $p(x)$ (provided the variance σ^2 is finite), with mean value μ and variance $\sigma_S^2 = \sigma^2/N$, *i.e.*

$$\begin{aligned} G(s) &= \frac{1}{\sqrt{2\pi\sigma_S^2}} \exp\left[-\frac{(s-\mu)^2}{2\sigma_S^2}\right] \\ &= \frac{1}{\sqrt{2\pi\sigma^2/N}} \exp\left[-\frac{(s-\mu)^2}{2\sigma^2/N}\right] . \end{aligned} \tag{13}$$

In an actual experiment or simulation the mean value and the variance are estimated by evaluating

$$\mu \simeq \frac{1}{N}\sum_{i=1}^{N} X_i \tag{14}$$

and

$$\sigma^2 \simeq \left[\frac{1}{N}\sum_{i=1}^{N} X_i^2 - (\frac{1}{N}\sum_{i=1}^{N} X_i)^2\right] , \tag{15}$$

respectively. Due to that the mean value μ is estimated the more correct expression for the variance of S is

$$\sigma_S^2 = \sigma^2/(N-1) . \tag{16}$$

For the Gaussian distribution, the probability of a measured result falling between $\pm\sigma_S$ of the exact μ is about 68%; the probability for falling within $\pm 2\sigma_S$ is about 95%. Therefore, if N is sufficiently large for the central limit theorem to hold, then the sum S in Eq.(12) provides an estimate of μ that has a 68% chance of being within $\pm\sigma_S$ and a 95 % chance of being within $\pm 2\sigma_S$ of the true mean value.

Central limit theorem The sum

$$I = \frac{1}{N}\sum_{i=1}^{N} f_i$$

is approximately Gaussian distributed

$$P(I) = \frac{1}{\sqrt{2\pi\sigma_I^2}} \exp\left[-\frac{(I-\mu)^2}{2\sigma_I^2}\right]$$

with mean value

$$\mathrm{E}[I] = \mu = \langle f \rangle$$

and variance

$$\mathrm{Var}[I] = \sigma_I^2 = \sigma_f^2/N = \left\langle (f - \langle f \rangle)^2 \right\rangle / N$$

The requirements are:

1. The variables $(f_1, \ldots, f_i, \ldots, f_N)$ have to be statistically independent.
2. The mean value $\mu = \langle f \rangle$ and the variance $\sigma_f^2 = \left\langle (f - \langle f \rangle)^2 \right\rangle$ have to exist.
3. N has to be sufficiently large.

Notice: This is independent on the actual distribution for f.

2 Random numbers

Random numbers are made available on most computers as part of the software or in common application libraries. The sequence of numbers produced by the computer cannot be truly random. Some underlying well-defined algorithm produces the numbers which therefore ought to be predictable. For that reason the numbers produced by a computer are often called *pseudo-random*. To test for randomness different statistical tests are available and good random number generators should pass these tests. A good random number generator should produce numbers that appear to be perfectly random, unless you happen to know both the algorithm and its internal state.

2.1 Uniform random numbers

The basic building block for most random number generators is a routine that produces a random number uniformly distributed in a specified range, typically between 0 and 1. These numbers are sometimes called *uniform deviates*. During the last decades the state of the art for generating uniform deviates has advanced considerably and different efficient generators are now available [4].

For a long time routines based on linear congruential generators dominated. These generates a sequence of integers $I_1, I_2, I_3, \ldots$ by the recurrence relation

$$I_{j+1} = (aI_j + c) \mod m \tag{17}$$

where a, c and m are integers and mod denotes the modulus operation, *i.e.* I_{j+1} is the remainder when the integer $(aI_j + c)$ is divided by m. The generator has to be initialized with a "seed" number I_0. By construction all integers in the sequence $I_1, I_2, I_3, \ldots$ are located between 0 and $m-1$ and the sequence will eventually repeat itself, with a period no greater than m. The desired uniform random number is obtained from

$$\xi_i = I_i/m. \tag{18}$$

If m, a and c are properly chosen one can prove using number theory that the period will be of maximal length, *i.e.* of length m, and all possible integers between 0 and $m-1$ occur once and only once. One such generator is obtained by choosing $a = 16807$, $c = 0$, and $m = 2^{31} - 1 = 2147483647$ [5]. Today the recommendation is to avoid linear congruential generators [4]. Other and better generators have been developed, often based on combining different unrelated methods [4].

2.2 Non-uniform random numbers

In many situation random numbers are needed that are not distributed uniformly but according to some probability density function $p(x)$. Here we

will present two methods to generate such non-uniform random numbers, the transformation and rejection method. Both methods are based on the input from a uniform random number generator, providing uniform random numbers ξ on [0,1]. For the non-uniform random number we will use the notation η.

2.2.1 Transformation method

Discrete case

We start with a discrete probability distribution function p_i, with $i = 1, \ldots, N$. Consider first a case with only two states with probabilities p_1 and p_2. To choose the states with correct probabilities using a uniform random number ξ we simply choose state 1 if $\xi < p_1$, otherwise we choose state 2. If there are three states with probabilities p_1, p_2, and p_3, then if $\xi < p_1$ we choose state 1, else if $\xi < p_1 + p_2$ we choose state 2, else we choose state 3. We can generalize this to N states. We then have to find the value of i that satisfies the condition

$$F_{i-1} \leq \xi \leq F_i \tag{19}$$

where F_i is the cumulative distribution function (cf Eq. (4))

$$F_i = \sum_{j=0}^{i} p_j$$

where we have defined $p_0 = 0$. According to Eq. (19) state i is chosen with the correct probability p_i.

Continuous case

In the case of a continuous probability distribution function $p(x)$ the cumulative probability function $F(x)$ becomes an integral and the two sums in Eq. (19) becomes identical and the inequalities become equalities,

$$F(x) = \xi \tag{20}$$

where $F(x)$ is defined in Eq. (2). By introducing the inverse function $F^{-1}(y)$ to $F(x)$ we can write

$$\eta = F^{-1}(\xi) \tag{21}$$

where η is a non-uniform random number that is distributed according to $p(x)$.

Transformation method A non-uniform random number η with probability distribution $p(x)$.

1. Determine $F(x) = \int_{-\infty}^{x} p(x')\,\mathrm{d}x'$
2. Determine the inverse $F^{-1}(y)$
3. Generate a uniform random number ξ
4. Obtain $\eta = F^{-1}(\xi)$

Uniform distribution A trivial example is to generate random numbers uniformly distributed on some interval [a,b]. In that case

$$p_u(x) = \begin{cases} 1/(b-a) & a \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$$

and

$$F(x) = \frac{x-a}{b-a}$$

By inverting $F(x)$ and using Eq. (21) we obtain

$$\eta = a + (b-a)\xi \tag{22}$$

Exponential distribution As a less trivial example consider the exponential distribution

$$p_e(x) = \begin{cases} \lambda \exp(-\lambda x) & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{23}$$

The cumulative distribution function is given by

$$F(x) = 1 - \exp(-x/\lambda)$$

and by inverting this function and using Eq. (21) we obtain

$$\eta = -\lambda \ln(1-\xi)$$

ξ is a uniform random number and therefore also $(1-\xi)$. A non-uniform random number η with exponential distribution and average value λ can therefore be obtained from the expression

$$\eta = -\lambda \ln(\xi) \tag{24}$$

where ξ is a uniform random number.

Gaussian distribution Another important probability distribution function is the Gaussian distribution

$$p_g(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \tag{25}$$

with mean value μ and variance σ^2. In this case the cumulative distribution function is given by the error function $\mathrm{erf}(x)$ according to

$$F(x) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x-\mu}{\sqrt{2}\sigma}\right)\right]$$

but its inverse is not known in closed form. However, it is known for the two dimensional case. Consider the joint probability function of two independent Gaussians with zero mean and unit variance

$$p(x)p(y) = \frac{1}{2\pi} e^{-x^2/2} e^{-y^2/2}\,\mathrm{d}x\,\mathrm{d}y$$

and introduce polar coordinates

$$\left\{\begin{array}{rcl} x &=& \rho\cos\theta \\ y &=& \rho\sin\theta \end{array}\right.$$

That transforms the joint probability distribution to

$$p(x)p(y)\,\mathrm{d}x\,\mathrm{d}y = \frac{1}{2\pi} e^{-\rho^2/2} \rho\,\mathrm{d}\rho\,\mathrm{d}\theta$$

which is further simplified to

$$p(x)p(y)\,\mathrm{d}x\,\mathrm{d}y = e^{-u}\,\mathrm{d}u\,\mathrm{d}a$$

using the substitution

$$\left\{\begin{array}{rcll} u &=& \rho^2/2\ , & 0 < u < \infty \\ a &=& \theta/2\pi\ , & 0 < a < 1 \end{array}\right.$$

The two variables u and a are independent, the joint probability distribution function is a product of two functions, e^{-u} and 1, respectively. This implies that u is exponential distributed while a is uniformly distributed. Using two uniform random numbes ξ_1 and ξ_2, we have

$$\left\{\begin{array}{rcl} u &=& -\ln\xi_1 \\ a &=& \xi_2 \end{array}\right.$$

Two independent gaussian random numbers η_1 and η_2, with zero mean and unit variance, can therefore be obtained from the expressions

$$\left\{\begin{array}{rcl} \eta_1 &=& \sqrt{-2\ln\xi_1}\ \cos(2\pi\xi_2) \\ \eta_2 &=& \sqrt{-2\ln\xi_1}\ \sin(2\pi\xi_2) \end{array}\right.$$

They can be transformed to Gaussian random numbers with mean value μ and variance σ^2 according to

$$\begin{cases} \eta_1 &= \mu + \sigma \sqrt{-2 \ln \xi_1} \cos(2\pi\xi_2) \\ \eta_2 &= \mu + \sigma \sqrt{-2 \ln \xi_1} \sin(2\pi\xi_2) \end{cases}$$

This way of generating Gaussian random numbers is known as the Box-Müller method.

2.2.2 Rejection method

Another method that can be used both for discrete and continuous random numbers is the *rejection method*. It does not require that the inverse of a cumulative distribution function can be readily computed as required by the transformation method. It is in that respect more general and it has a simple geometric interpretation.

Consider a continuous probability density $p(x)$ on a finite interval [a,b]. Choose a value p_{max} such that

$$p_{max} \geq p(x) \ , \quad \forall x$$

Generate a uniform random number ξ_1 on [0,1] and determine the trial value $x_{try} = a+(b-a)\xi_1$. Generate another uniform random number ξ_2 and accept the trial value x_{try} if $\xi_2 \leq p(x_{try})/p_{max}$ and set $\eta = x_{try}$, otherwise reject the trial value x_{try} and repeat the procedure. The random number η will then be given by the probability distribution $p(x)$. The rejection method is attributed to von Neumann and it is for instance used in the Metropolis algorithm.

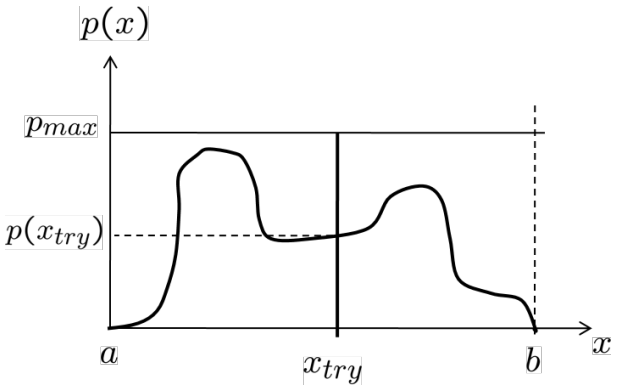


Illustration of the rejection method.

Rejection method A non-uniform random number η with probability distribution $p(x)$ on [a,b].

1. Choose a value p_{max} such that $p_{max} \geq p(x)$, $a < x < b$
2. Generate a uniform random number ξ_1 and determine a trial value $x_{try} = a + (b - a)\xi_1$
3. Generate another uniform random number ξ_2 and set $\eta = x_{try}$, but only if $\xi_2 \leq p(x_{try})/p_{max}$
4. Otherwise reject x_{try}, go back to 2. and try again.

Repeat the procedure until sufficient many random numbers η have been generated.

The technique can be made applicable also for a probability distribution $p(x)$ defined on an infinite interval. One then has to find a comparison function $p_{max}(x)$ with the property

$$p_{max}(x) \geq p(x) \ , \quad \forall x$$

and with $\int p_{mac}(x)\,\mathrm{d}x$ finite. One then has to sample points according to the comparison function $p_{max}(x)$ and then the method follows the recipe above.

The method can also be extended to higher dimensions. One example is choosing random direction on a sphere in three dimensions.

Random direction Random vectors on the surface of a sphere in three dimensions.

1. Generate three uniform random numbers ξ_1, ξ_2 and ξ_3.
2. Calculate $\eta_i = 1 - 2\xi_i$ and form the som $\eta^2 = \eta_1^2 + \eta_2^2 + \eta_3^2$.
3. If $\eta^2 < 1$ take $\hat{\eta} = (\eta_1/\eta, \eta_2/\eta, \eta_3/\eta)$ as a unit vector, else reject the vector and return to 1.

3 Monte Carlo integration

3.1 Monte Carlo integration with uniform sampling

To illustrate the basic idea behind Monte Carlo integration we will first consider a one-dimensional integral

$$I = \int_0^1 f(x)\,\mathrm{d}x \ . \tag{26}$$

Conventional numerical integration methods are based on evaluation of the integrand at particular values x_i. The integral is then obtained as some weighted sum over the corresponding values of the integrand, $f_i \equiv f(x_i)$. Often the x-values are chosen equally spaced while more elaborate methods also optimize the location of the x-values.

Here we will introduce a different approach, *Monte Carlo integration*. The integral can be written as an average over f

$$I = \langle f \rangle_u \ = \int_0^1 f(x)\,\mathrm{d}x \ , \tag{27}$$

where the subscript u indicates that the corresponding probability density function is a uniform distribution, here on the interval [0,1]. Instead of evaluating the integrand at prescribed values of x we can simply choose the x-values *randomly* with equal probability on the interval [0,1]. The integral is then approximated by the average value

$$I_N = \frac{1}{N}\sum_{i=1}^{N} f(x_i) = \frac{1}{N}\sum_{i=1}^{N} f_i \ . \tag{28}$$

To make the method useful one has to be able to estimate the error. This can be done using the central limit theorem. The variables $(f_1, \ldots, f_i, \ldots, f_N)$ are statistically independent provided the random number generator is adequate. An estimate of the variance is obtained by evaluating

$$\sigma_f^2 = \left\langle (f - \langle f \rangle_u)^2 \right\rangle_u \simeq \left[\frac{1}{N}\sum_{i=1}^{N} f_i^2 - \left(\frac{1}{N}\sum_{i=1}^{N} f_i \right)^2 \right] \tag{29}$$

and hence the integral can be approximated by

$$I = I_N \pm \sigma_I = I_N \pm \frac{\sigma_f}{\sqrt{N}} \ . \tag{30}$$

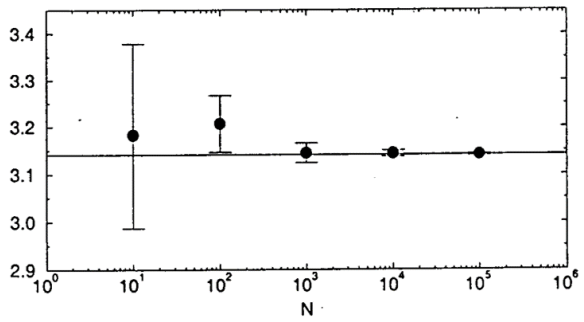
Here one standard deviation is used. It is implies that the correct value I is within the interval $\pm\sigma_f/\sqrt{N}$ with 68% probability. Using two standard deviations $\pm 2\sigma_f/\sqrt{N}$ the corresponding probability is 95%.

Example MC1 As an example consider the integral

$$\int_0^1 \frac{4}{1+x^2}\,\mathrm{d}x = \pi \ .$$

The table shows the result N using different number of evaluation of the integrand and the data with error bars (one standard deviation) is shown in the figure. The calculated result is equal to the exact value within a few (usually less than one) standard deviations and the integration becomes more precise as N increases.

	N	I_N	σ_f	$\sigma_I = \sigma_f/\sqrt{N}$
MC	10^1	3.18266	0.61954	0.19592
MC	10^2	3.20677	0.60315	0.06032
MC	10^3	3.14463	0.65030	0.02056
MC	10^4	3.14380	0.63989	0.00640
MC	10^5	3.14096	0.64438	0.00204
exact		3.14159	0.64310	



The uncertainty in the estimate of the integral decreases very slowly with the number of points, as $\mathcal{O}(N^{-1/2})$. This is to be contrasted with more conventional methods like the trapezoidal formula, where the error scales as $\mathcal{O}(N^{-2})$ and considerably less computational time is required to obtain the same accuracy. However, the key point in Monte Carlo integration is that the error is independent on the dimension of the integral. This is not the case with the more conventional, "grid based", methods.

Consider the evaluation of an integral in d dimensions. Suppose that you are willing to invest a given amount of computational time, a certain number N of evaluations of the integrand. Using a conventional method each dimension of the d-dimensional integral is than broken up into $\sim N^{1/d}$ intervals with spacing $h \sim N^{-1/d}$. Using the trapeziodal rule the error for each cell volume h^d in the integration region is $\mathcal{O}(h^{d+2})$, so that the total

error is $N\mathcal{O}(h^{d+2}) = \mathcal{O}(N^{-2/d})$. For large d this decreases very slowly with increasing N. This can be compared with the Monte Carlo method where the error scales as $\mathcal{O}(N^{-1/2})$. Assuming that the prefactors in these estimates are similar we see that Monte Carlo integration becomes more efficient for $d > 4$. The details depends on the conventional quadrature scheme, but the key point is the very different way in which the two errors scale with increasing N for large d.

Monte Carlo integration - uniform sampling Consider the integral

$$I = \int_0^1 f(x)$$

1. Choose N points x_i at random with uniform probability within the integration interval [0,1].
2. Determine the mean value

$$I_N = \frac{1}{N}\sum_{i=1}^{N} f_i$$

and the variance

$$\sigma_f^2 = \left[\frac{1}{N}\sum_{i=1}^{N} f_i^2 - \left(\frac{1}{N}\sum_{i=1}^{N} f_i\right)^2\right]$$

3. Approximate the value of the integral as

$$I = I_N \pm \frac{\sigma_f}{\sqrt{N}}$$

3.2 Variance reduction and importance sampling

The accuracy of Monte Carlo integration increases with the number of samping points N. From Eq. (30) we notice that the accuracy also increases if the variance σ_f of the integrand could be decreased. This can be done using *importance sampling*. Consider some function $p(x)$ that is positive and normalized to 1,

$$\int_a^b p(x)\,\mathrm{d}x = 1 \text{ and } \; p(x) > 0 \;\text{ on }\; [a,b]\;,$$

i.e. a probability density function. Rewrite the integral as

$$I = \int_a^b f(x)\,\mathrm{d}x = \int_a^b \frac{f(x)}{p(x)}p(x)\,\mathrm{d}x = \int_a^b g(x)p(x)\,\mathrm{d}x = \langle g\rangle_p \qquad (31)$$

where then

$$g(x) \equiv \frac{f(x)}{p(x)}$$

The integral in Eq. (31) can be evaluated using the Monte Carlo technique by sampling x not uniformly but according to the probability density function $p(x)$, indicated by the notation $\langle \ldots \rangle_p$. The average is obtained as

$$I_N = \frac{1}{N} \sum_{i=1}^{N} g(x_i) = \frac{1}{N} \sum_{i=1}^{N} g_i$$

and the variance σ_g^2 correspondingly. If the function $p(x)$ behaves approximately as $f(x)$, *i.e.* is large where $f(x)$ is large and small where $f(x)$ is small, then the new integrand $g(x)$ is made more smooth and the variance is reduced, $\sigma_g \ll \sigma_f$.

Example MC2 Consider again the integral in Example MC1

$$\int_0^1 \frac{4}{1+x^2}\,\mathrm{d}x = \pi \ .$$

If we introduce the normalized probability density function

$$p(x) = \frac{4-2x}{3}$$

as an importance sampling function the result below is obtained. The sampling can conveniently be done using the Transformation method, introduced in Sec. (2.2.1). The error bars correspond to one standard deviation and we notice that the method is about 10 times more efficient compared with uniform sampling (cf. Example MC1).

	N	I_N	σ_f	$\sigma_I = \sigma_f/\sqrt{N}$
MC	10^1	3.18155	0.06746	0.02133
MC	10^2	3.14107	0.08327	0.00833
MC	10^3	3.13931	0.08102	0.00256
MC	10^4	3.14119	0.08072	0.00081
MC	10^5	3.14200	0.08012	0.00025
exact		3.14159	0.08002	

We have introduced a more efficient integration technique by reducing the variance. This is accomplished by sampling x non-uniformly. The sampling points are concentrated around more "important" values of x, where $p(x)$ and (hopefully) $f(x)$ is large, and less computing power is spent on calculating the integrand for "unimportant" values of x where $p(x)$ and $f(x)$

are small. The technique is therefore called *importance sampling* and it reduces the variance. However, it is based on that one can find a way to sample points according to the chosen function $p(x)$.

Monte Carlo integration - importance sampling Consider the integral

$$I = \int_a^b f(x)\,\mathrm{d}x = \int_a^b \frac{f(x)}{p(x)} p(x)\,\mathrm{d}x = \int_a^b g(x)p(x)\,\mathrm{d}x$$

with

$$\int_a^b p(x)\,\mathrm{d}x = 1$$

1. Choose N points x_i at random with probability $p(x)$ within the integration interval [a,b].
2. Determine the mean value

$$I_N = \frac{1}{N}\sum_{i=1}^{N} g_i$$

and the variance

$$\sigma_g^2 = \left[\frac{1}{N}\sum_{i=1}^{N} g_i^2 - \left(\frac{1}{N}\sum_{i=1}^{N} g_i\right)^2\right]$$

3. Approximate the value of the integral as

$$I = I_N \pm \frac{\sigma_g}{\sqrt{N}}$$

Notice that $\sigma_g \ll \sigma_f$ if $p(x)$ "mimics" $f(x)$.

The real power of Monte Carlo integration is in connection to high dimensional integrals. Metropolis *et al.* [2] considered a problem in classical statistical mechanics, the equilibrium properties of N interacting particles. If we consider a system at temperature T the average value of some quantity A that depends on the positions of the particles $(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N)$ is given by the integral

$$\langle A \rangle = \int \mathrm{d}\boldsymbol{r}_1 \ldots \mathrm{d}\boldsymbol{r}_N A(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N) P(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N) \tag{32}$$

where

$$P(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N) = \frac{\exp[-V(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N)/k_B T]}{\int \mathrm{d}\boldsymbol{r}_1 \ldots \mathrm{d}\boldsymbol{r}_N \exp\left[-V(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N)/k_B T\right]} \tag{33}$$

and $V(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N)$ is the potential energy for the system as function of the

positions of the particles and k_B Boltzmann's constant. Straightforward evaluation of the integral in Eq. (32) using some conventional method is completely out of question. Assuming a system with N=100 particles in three dimensions corresponds to evaluating a 300 dimensional integral. Say that you take 10 points in each dimension which implies 10^{300} evaluation of the integrand. Even if you have access to the fastest computer in the world that perform say 100 PFLOPS (10^{17} operations per second) you will only be able to perform 10^{35} floating point operations during the age of the universe. 10^{35} evaluations are totally negligible compared to what you need, 10^{300} evaluations.

Even if you could perform the integral with the above gridspacing the result would most likely be quite useless with large statistical uncertainties due to too large gridspacing. The integrand is a rapidly varying function with respect to the particle positions due to the Boltzmann factor $\exp[-V(\boldsymbol{r}_1,\ldots,\boldsymbol{r}_N)/k_BT]$. For instance, for a liquid of 100 particles modelled as hard spheres and at a temperature close to the freezing point only 1 out of 10^{260} configurations would be non-zero [6] due to the Boltzmann factor $\exp[-V(\boldsymbol{r}_1,\ldots,\boldsymbol{r}_N)/k_BT]$.

However, this problem was solved numerically by Metropolis *et al.* 1953 [2] using Monte Carlo integration and by introducing a new sampling technique, importance sampling. They stated that "instead of choosing configurations randomly, then weighting them with $\exp(-E/kT)$, we choose configurations with a probability $\exp(-E/kT)$ and weight them evenly". The key contribution by Metropolis *et al.* [2] was to introduce a general method to choose configurations from some general probability distribution function in high dimensions, in their case $P(\boldsymbol{r}_1,\ldots,\boldsymbol{r}_N)$ in Eq. (33) (in 2 dimensions). The method was based on a random walk procedure and it has been extended, generalized and applied in many other research areas. It is now known as the *Markov Chain Monte Carlo* (MCMC) technique.

4 Markov Chain Monte Carlo

4.1 Markov process

We start with some basic properties of Markov processes. Consider a physical system that can be in different states. For simplicity assume a discrete space and that the number of states is finite, equal to M. Denote the different states by Ω_m, with $m = 1, \ldots, M$. The number of states can be very large. For instance, for a simple "magnetic" system (the Ising model) in 3 dimensions with 10×10×10 lattice points the number of states is equal to $M = 2^{1000} \simeq 10^{300}$.

Consider now a stochastic process, a sequence of states generated in a stochastic way. Enumerate the sequence using $s = 0, 1, 2, \ldots$. The process is called a Markov process, or *Markov chain*, if the outcome at any step s only depends on the present state. It has no memory. The sequence is in that case uniquely defined by the conditional probabilities

$$w_{nm} = w_{n \leftarrow m} = \mathsf{Prob}\,(n, s+1 \mid m, s) \tag{34}$$

together with the specification of the initial state. The quantity w_{nm} is the probability to make a transition from m to n, provided the system is in state m at step s. These probabilities, which are independent on s, define a matrix, the *transition matrix* $\mathbf{W}$, and they fulfil the following two conditions

$$0 \leq w_{nm} \leq 1 \qquad \forall\ n \text{ and } m \tag{35}$$

and

$$\sum_{n=1}^{M} w_{nm} = 1 \qquad \forall\ m \tag{36}$$

The last condition simply implies that with unit probability the system will be in one of its M allowed states at the next step. In a sense, the Markov chain is the probabilistic analogue to trajectories in classical mechanics. The "time"-evolution of the Markov process is determined by a stochastic matrix, while the classical trajectories are given by Newton's equation of motion, which is deterministic. However, both are characterized by a lack of memory, i.e. the immediate future is uniquely determined by the present, regardless of the past.

Example MC3 Consider the daily weather in Göteborg. It can be in three different "states": (1) sunny, (2) cloudy, or (3) rainy. Construct the transition matrix based on the following observations. A sunny day is never followed by another sunny day. Rainy or cloudy weather is equally probable after a sunny day. A rainy or cloudy day is followed by 50% probability by another day with the same weather. If, on the other hand, the weather

is changing from cloudy or rainy weather, the following day will be sunny only in half of the cases. From this the following transition matrix can be constructed.

$$\mathbf{W} = \begin{bmatrix} 0 & 0.25 & 0.25 \\ 0.5 & 0.5 & 0.25 \\ 0.5 & 0.25 & 0.5 \end{bmatrix}$$

Notice that $\sum_{n=1}^{3} w_{nm} = 1$ for $m = 1$, 2, and 3.

Denote the probability that the system is in state Ω_m at step or "time" s as $p_m(s)$. The probability distribution may then be represented by the column vector

$$\mathbf{P}(s) = \begin{bmatrix} p_1(s) \\ \vdots \\ p_M(s) \end{bmatrix} \tag{37}$$

with the normalization $\sum_{m=1}^{M} p_m(s) = 1$. At each time point s the system may move from a state Ω_m to a state Ω_n with probability w_{nm}. Hence, the probability distribution will evolve as

$$p_n(s+1) = \sum_{m=1}^{M} w_{nm} p_m(s)$$

or in matrix form

$$\mathbf{P}(s+1) = \mathbf{W}\,\mathbf{P}(s) \tag{38}$$

Using this notation the probability distribution evolves from the initial distribution as follows: $\mathbf{P}(1) = \mathbf{W}\,\mathbf{P}(0)$ and then $\mathbf{P}(2) = \mathbf{W}\,\mathbf{P}(1) = \mathbf{W}^2\,\mathbf{P}(0)$. After s steps $\mathbf{P}(s)$ is related to $\mathbf{P}(0)$ by

$$\mathbf{P}(s) = \mathbf{W}^s\,\mathbf{P}(0) \tag{39}$$

It is likely that when s becomes large the initial information contained in $\mathbf{P}(0)$ will fade away and under quite general conditions [7] one can show that $\mathbf{P}(s)$ approaches a "time"-independent constant distribution

$$\mathbf{P}(s \to \infty) = \mathbf{P}^{\mathrm{st}} \tag{40}$$

This distribution is a fixed point distribution of the transition matrix $\mathbf{W}$ and is called the *stationary distribution*. It must satisfy the equation

$$\mathbf{P}^{\mathrm{st}} = \mathbf{W}\,\mathbf{P}^{\mathrm{st}} \tag{41}$$

This is an eigenvalue equation and it implies that the stationary distribution corresponds to the eigenvector with eigenvalue $\lambda = 1$. It is easy to show that all transition matrices $\mathbf{W}$ have at least one eigenvalue that is equal to 1 and, hence, a corresponding stationary distribution (see App. A). The absolute value of all other eigenvectors are less than 1, $|\lambda_i| \leq 1$, and they determine how fast $\mathbf{P}(s)$ approaches the stationary distribution $\mathbf{P}^{\mathrm{st}}$ (see App. A).

Example MC4 Consider again the daily weather in Göteborg in example MC3. We can now solve for the stationary distribution using Eq. (41) $p_n^{\rm st} = \sum_{m=1}^{3} w_{nm} p_m^{\rm st}$, under the constraint that $\sum_{n=1}^{3} p_n^{\rm st} = 1$. We get

$$\mathbf{P}^{\rm st} = \begin{bmatrix} 0.2 \\ 0.4 \\ 0.4 \end{bmatrix}$$

We can also study the convergence of the Markov chain towards the stationary distribution using Eq. (39). The table below gives the result for two different initial distributions. The convergence is slightly faster for the initial distribution that is "closer" to the stationary distribution, but in both cases the convergence is quite rapid.

s	p_1	p_2	p_3
0	1.00000	0.00000	0.00000
1	0.00000	0.50000	0.50000
2	0.25000	0.37500	0.37500
3	0.18750	0.40625	0.40625
4	0.20312	0.39844	0.39844
5	0.19922	0.40039	0.40039
6	0.20020	0.39990	0.39990
7	0.19995	0.40002	0.40002
8	0.20001	0.39999	0.39999
9	0.20000	0.40000	0.40000
10	0.20000	0.40000	0.40000
∞	0.20000	0.40000	0.40000

s	p_1	p_2	p_3
0	0.50000	0.00000	0.50000
1	0.12500	0.37500	0.50000
2	0.21875	0.37500	0.40625
3	0.19531	0.39844	0.40625
4	0.20117	0.39844	0.40039
5	0.19971	0.39990	0.40039
6	0.20007	0.39990	0.40002
7	0.19998	0.39999	0.40002
8	0.20000	0.39999	0.40000
9	0.20000	0.40000	0.40000
10	0.20000	0.40000	0.40000
∞	0.20000	0.40000	0.40000

Consider now some quantity f that depends on the state Ω_m of the system, $f_m = f(X_m)$. It could, for instance, be the pressure in a classical liquid or the "magnetization" in an Ising model. Assuming that the probability for the different states $\Omega_m, (m = 1, \ldots M)$ is given by the stationary distribution $\mathbf{P}^{\rm st}$ the average of f can be written as the *ensemble average*

$$\langle f \rangle_{ens} = \frac{1}{M} \sum_{i=1}^{M} f(X_m) \tag{42}$$

Example MC5 We can construct a "sunindex" $f^{\odot}$ with the property that $f^{\odot}$=10 for a sunny(1) day, $f^{\odot}$=5 for a cloudy(2) day, and $f^{\odot}$=0 for a rainy(3) day. The average sunindex for Göteborg is

$$\left\langle f^{\odot} \right\rangle_{ens} = 10 \cdot \frac{1}{5} + 5 \cdot \frac{2}{5} + 0 \cdot \frac{2}{5} = 4$$

which is slightly less than the sunindex for a cloudy day.

4.2 Random walk in state space

The number of states M in actual applications of the Markov chain method is often enormous. The ensemble average in Eq. (42) then becomes totally impractical. Instead one focuses the attention on a single random walker, not on the probability distribution function. Assume that the single walker is in state k at time s, $X_k^{(s)}$. In the next step, time $s+1$, the walker will be in state l, $X_l^{(s+1)}$, with probability w_{lk}. In this way we can generate a sequence of sample points

$$X^{(0)}, X^{(1)}, X^{(2)}, \ldots, X^{(s)}, X^{(s+1)}, \ldots, X^{(N)}$$

and we can perform a *time average* along this chain

$$\langle f \rangle_{time} = \frac{1}{N} \sum_{s=1}^{N} f(X^{(s)}) \tag{43}$$

In the limit $N \to \infty$ we expect the time-average in Eq. (43) is equal to the ensemble average in Eq. (42). Such transition matrices are said to be *ergodic*. The matrices have to be irreducible and aperiodic [7]. Irreducible implies that the model only has one unique stationary distribution and aperiodic means that you do not get stuck into a periodic orbit.

Example MC6 Consider again the daily weather in Göteborg in example MC3. We can now simulate the weather using the probabilities w_{nm} introduced in example MC3. We denote a sunny, cloudy, and rainy day with $X = S$, $X = C$, and $X = R$, respectively. If we start with a day with sunny weather, $X = S$, we get the following sequence using random numbers from a random number generator

s	0	1	2	3	4	5	6	7	8	9	10	11	12	...
X	S	R	C	C	S	C	R	S	R	C	C	C	S	...

The sequence can be used to determine average properties using the time average in Eq. (43). For the sunindex $f^{\odot}$, introduced in Example MC5, we

get

$$\left\langle f^{\odot}\right\rangle_{time} = \frac{1}{10}\sum_{s=1}^{10} f^{\odot}(X^{(s)}) = 4.50$$

using the 10 first days. By expanding to 100 and 1000 days, respectively, we get

$$\begin{aligned}\left\langle f^{\odot}\right\rangle_{time} &= \frac{1}{100}\sum_{s=1}^{100} f^{\odot}(X^{(s)}) = 3.95\\ \left\langle f^{\odot}\right\rangle_{time} &= \frac{1}{1000}\sum_{s=1}^{1000} f^{\odot}(X^{(s)}) = 4.01\end{aligned}$$

In an actual simulation the time average in Eq. (43) is used. The number of configurations N has to be large but still N is much smaller than M. If the time average should be accurate it is important that relevant configurations are sampled. If the starting point $X^{(0)}$ is very unlikely it can take many steps before a more probable part of the distribution is sampled. There is thus a need to "equilibrate" or "burn-in" the Markov chain by steeping through, and discarding, a certain number, say N_{eq}, of sampling points. In many cases this "equilibration" of the system can be a substantial part of the total simulation. The subsequent sampling points are then used to determine different quantities according to

$$\langle f\rangle = \frac{1}{N}\sum_{s=N_{eq}+1}^{N_{eq}+N} f(X^{(s)}) \tag{44}$$

It is crucial to be able to estimate error bounds associated with the mean value defined in Eq. (44). However, the values along the Markov chain $f_s = f(X^{(s)})$ can be highly correlated and are not statistically independent and the error cannot be estimated as in Eq. (30). What is often done in practice is to use states along the Markov chain separated by some fixed number n_s, chosen so that there is effectively no correlation between the states used. The number n_s is called statistical inefficiency. In Appendix B two ways to estimate n_s are presented. The effective number of configurations to be used in the averaging then becomes equal to N/n_s. Using the central limit theorem we can then write

$$f = \langle f\rangle \pm \frac{\sigma}{\sqrt{N/n_s}} \tag{45}$$

where $\sigma^2 = \left\langle f^2\right\rangle - \langle f\rangle^2$ and the error bar corresponds to one standard deviation.

4.3 Basic idea

We have now shown how a stationary distribution $\mathbf{P}^{\text{st}}$ can arise using a Markov process defined by a transition matrix $\mathbf{W}$. Our aim is to use the obtained stationary distribution $\mathbf{P}^{\text{st}}$ in a simulation study. However, we do not yet have control over $\mathbf{P}^{\text{st}}$. It is a consequence of a given transition matrix $\mathbf{W}$.

If we would like to generate a particular distribution, say $\mathbf{P}$, we need to invert the above procedure to find the appropriate $\mathbf{W}$ for the desired $\mathbf{P}$. The size of the vector $\mathbf{P}$ is M while $\mathbf{W}$ is a matrix with size $M \times M$. This implies that there are many different matrices $\mathbf{W}$ that will generate the same probability distribution $\mathbf{P}$ and there is a large freedom in construction $\mathbf{W}$. The probability transition matrix $\mathbf{W}$ has to fulfil the two conditions in Eqs. (35) and (36), be ergodic, and satisfy

$$\mathbf{P} = \mathbf{W}\,\mathbf{P}$$

or

$$p_n = \sum_{m=1}^{M} w_{nm} p_m \tag{46}$$

By using $\sum_{m=1}^{M} w_{mn} = 1$ this last condition can be written as

$$\sum_{m=1}^{M} w_{mn} p_n = \sum_{m=1}^{M} w_{nm} p_m$$

A sufficient but not necessary condition is to require that each term in the two sums are equal, *i.e*

$$w_{mn} p_n = w_{nm} p_m \tag{47}$$

This is called *detailed balance*. It has a simple physical interpretation. Eq. (47) states that the probability for a transition from n to m, provided the system is located in n, is equal to the probability for the reversed transition, a transition from m to n, provided the system is in m. If that is the case, nothing will occur in average and, hence, $\mathbf{P}$ is a stationary distribution. The weaker condition in Eq. (46) has been used in few cases [8, 9], but in practise, the more restrictive detailed balance condition in Eq. (47) is used in constructing suitable transition matrices $\mathbf{W}$.

Basic idea If you can find a matrix $\mathbf{W}$ with the following properties

$$0 \leq w_{nm} \leq 1 \quad \forall\ n \text{ and } m \tag{48}$$

$$\sum_{n=1}^{M} w_{nm} = 1 \quad \forall\ m \tag{49}$$

$$w_{nm} \text{ is ergodic} \tag{50}$$

$$w_{mn} p_n = w_{nm} p_m \tag{51}$$

then the Markov process will, in the long run, produce states distributed according to the probability distribution $\mathbf{P}$.

Example MC7 The weather in Göteborg

$$\mathbf{P}^{\text{st}} = \begin{bmatrix} 0.2 \\ 0.4 \\ 0.4 \end{bmatrix}$$

was generated in Example MC4 using the transition matrix introduced in Example MC3. One can construct other matrices $\mathbf{W}$ that give the same weather. The matrix has to fulfil detailed balance, *i.e.* $w_{21} = 2w_{12}$, $w_{31} = 2w_{13}$, and $w_{32} = w_{23}$ as well as the normalization $w_{11} + w_{21} + w_{31} = 1$, $w_{12} + w_{22} + w_{32} = 1$, and $w_{13} + w_{23} + w_{33} = 1$. We have 9 variables w_{nm} and only 6 equations. Many solutions can be found. For instance, by choosing $w_{21} = 0.4$, $w_{31} = 0.2$, and $w_{32} = 0.5$, we obtain the transition matrix

$$\mathbf{W} = \begin{bmatrix} 0.4 & 0.2 & 0.1 \\ 0.4 & 0.3 & 0.5 \\ 0.2 & 0.5 & 0.4 \end{bmatrix}$$

which also will give the "correct" weather in Göteborg.

4.4 The Metropolis algorithm

We now would like to construct the transition matrix $\mathbf{W}$ with $\mathbf{P}$ as the stationary distribution. This implies that we have to find a matrix $\mathbf{W}$ that fulfils the conditions Eqs (48)-(51).

In the Metropolis algorithm a transition step is split into two parts

$$w_{nm} = \tau_{nm}\, \alpha_{nm} \quad (n \neq m) \tag{52}$$

where $\tau_{nm} = \tau_{n \leftarrow m}$ is the probability to make a trial change from m to n and $\alpha_{nm} = \alpha_{n \leftarrow m}$ is the probability to accept the trial change, and

$$w_{mm} = 1 - \sum_{n(\neq m)} w_{nm} \tag{53}$$

In the original Metropolis algorithm [2] α_{nm} is given by

$$\alpha_{nm} = \begin{cases} 1 & \text{if } p_n \geq p_m \\ p_n/p_m & \text{if } p_n < p_m \end{cases} \tag{54}$$

which also can be written as

$$\alpha_{nm} = \min\left[1, \frac{p_n}{p_m}\right] \tag{55}$$

Detailed balance, Eq. (51), is fulfilled if the probability for the trial change is symmetric, *i.e.*

$$\tau_{nm} = \tau_{mn} \tag{56}$$

and the condition in Eq. (53) implies that Eq. (49) is fulfilled. Eq. (53) implies that moves that are not accepted are rejected and the system remains at the state m for at least one more step.

A key element in the Metropolis algorithm is that in generating the configurations only knowledge of the relative probabilities is needed, not the absolute probabilities. That is crucial. The relative probabilities can often be computed, but in practise, not the absolute probabilities.

To implement the Metropolis algorithm one has to specify the matrix τ_{nm}, which is designed to take the system from one state m to a trial state n. It is a probability and hence $\sum_n \tau_{nm} = 1$. The only constraint is that it is symmetric, $\tau_{nm} = \tau_{mn}$, and one has considerable freedom in finding an appropriate matrix τ_{nm}. Often it contains one or several adjustable parameters which can be tuned to optimize the sampling. The tuning can be done during the simulation. If nearly all states are accepted the sampling is not efficient, the changes introduced by τ_{nm} are "too small". On the other hand, if very few trial states are accepted the changes are "too large" and the efficiency is reduced. Optimum sampling rate depends on the application, but generally the acceptance ratio should be about 20 % to 50 % [6, 10].

The matrix τ_{nm} is in most cases constructed such that the trial state Ω_n is located in the "vicinity" of Ω_m. The reason for this is that one would like the Markov chain to "guide" the system into regions with high probability. Actually, it is this fact that makes the method efficient for high dimensional problems. If one would simple choose Ω_n totally at random one would never find the rare regions with high probability. However, a drawback is that the different states along the Markov chain become highly correlated and the estimate of error bounds has to be made carefully.

To ensure a proper sampling τ_{nm} has to be chosen such that all relevant states are tested. Mathematically this is expressed by Eq. (50), $\mathbf{W}$ should be ergodic. However, here one is not so much helped by the mathematical formulation but one has to rely more on physical intuition.

The Metropolis algorithm Consider a system that can be in different states Ω_m with corresponding probabilities p_m.
Decide how to make trail changes $\tau_{nm} = \tau_{n\leftarrow m}$ and establish an initial configuration $\Omega(s=0)$.
To advance the Markov chain one step, from configuration $\Omega(s) = \Omega_m$ to $\Omega(s+1)$

1) Choose a trail state Ω_t according to τ_{tm}
2) calculate the ratio $q = p_t/p_m$
3) generate a random number ξ between 0 and 1
   if $q \geq \xi$
     accept the change and let $\Omega(s+1) = \Omega_t$
   otherwise
     count the old state once more and let $\Omega(s+1) = \Omega_m$

Repeat step 1)-3) many times.
Throw away a sufficient number of states in the beginning. Determine average quantities with proper error bars.

A more symmetric expression for α_{nm} has been suggested [11]

$$\alpha_{nm} = \frac{p_n}{p_n + p_m} \tag{57}$$

and is often referred to as Barker sampling [12]. Eq. (57) also fulfil detailed balance provided τ_{nm} is symmetric and one has to allow for the possibility of no transition according to Eq. (53).

4.5 The Metropolis-Hastings algorithm

The Metropolis algorithm has also been generalized to the case when τ_{nm} is not equal to τ_{mn} by Hastings [13]. In the context of Monte Carlo simulations it is often called *smart Monte Carlo* [14] or *biased Monte Carlo* [6]. In that case, the acceptance criterion for a trial change $m \to n$ must be replaced by

$$\alpha_{nm} = \begin{cases} 1 & \text{if } \tau_{mn}p_n \geq \tau_{nm}p_m \\ (\tau_{mn}p_n/\tau_{nm}p_m) & \text{if } \tau_{mn}p_n < \tau_{nm}p_m \end{cases} \tag{58}$$

or equivalently

$$\alpha_{nm} = \min\left[1, \frac{\tau_{mn}p_n}{\tau_{nm}p_m}\right] \tag{59}$$

Also in this case one has to allow for no transition as expressed in Eq. (53).

A Markov Chain

In this appendix we will derive a more explicit expression for the Markov chain time evolution of the probability distribution $\mathbf{P}(s)$. We start with the expression for the evolution given by Eq. (39)

$$\mathbf{P}(s) = \mathbf{W}^s \, \mathbf{P}(0)$$

where $\mathbf{W}$ is the transition matrix. It is a $M \times M$ matrix with real components w_{nm} that fulfill the conditions $0 \leq w_{nm} \leq 1$ and $\sum_{n=1}^{M} w_{nm} = 1$. The evolution equation can be solved formally by introducing the corresponding eigenvectors and eigenfrequencies to $\mathbf{W}$. Due to that in general $\mathbf{W}$ is non-symmetric the left and right eigenvectors will be different. Using the Dirac vector notation we write

$$\mathbf{W}|\chi_i\rangle = \lambda_i |\chi_i\rangle \tag{60}$$

$$\langle\chi_i|\mathbf{W} = \langle\chi_i|\lambda_i \tag{61}$$

where $\langle\chi_i|$ and $|\chi_i\rangle$ are the left and right eigenvectors, respectively, λ_i the eigenvalues and $i = 1, \ldots, M$. The eigenvectors are orthogonal

$$\langle\chi_i|\chi_j\rangle = \delta_{i,j} \tag{62}$$

and they form a complete set

$$\sum_{i=1}^{M} |\chi_i\rangle\langle\chi_i| = \mathbf{I} \tag{63}$$

where $\mathbf{I}$ is the identity matrix [15].

The eigenvalues λ_i are given by the values of λ which satisfy the characteristic equation

$$\det(\mathbf{W} - \lambda\mathbf{I}) = 0 \tag{64}$$

We can prove that $\lambda = 1$ is always an eigenvalue. The determinant is unchanged if two rows are added together. By adding all rows together we get one row with the components $\sum_{n=1}^{M} [w_{nm} - \lambda\delta_{nm}] = 1 - \lambda$. By choosing $\lambda = 1$ we get a row of zeros, the determinant is then equal to zero and, hence, $\lambda = 1$ is a solution to Eq. (64). We choose this eigenvalue to have index 1, *i.e.*

$$\lambda_1 = 1 \tag{65}$$

By comparing Eq. (60) with Eq.(41) we notice that the corresponding right eigenvector is equal to the stationary distribution

$$|\chi_1\rangle = \mathbf{P}^{\mathrm{st}} \tag{66}$$

and by inspecting Eq. (61) we find that

$$\langle \chi_1 | = [1\ 1\ 1\ \ldots\ 1] \tag{67}$$

One can prove that $|\lambda_i| \leq 1$ for all i's [15]. (Otherwise, by repeatedly applying $\mathbf{W}$, $\mathbf{P}$ would increase indefinitely, which is not possible.) We assume here that only one eigenvalue is equal to 1, *i.e.* the stationary distribution is unique. This is the case for an ergodic transition matrix.

By using the eigenvectors and eigenfrequencies the transition matrix can now be written as

$$\mathbf{W} = \sum_{i=1}^{M} \lambda_i |\chi_i\rangle\langle\chi_i| = \sum_{i=1}^{M} \lambda_i \mathbf{B}^{(i)}$$

and we define a matrix $\mathbf{B}^{(i)}$ according to

$$\mathbf{B}^{(i)} = |\chi_i\rangle\langle\chi_i| \tag{68}$$

For s number of steps we then obtain

$$\mathbf{W}^s = \sum_{i=1}^{M} \lambda_i^s \mathbf{B}^{(i)}$$

and finally

$$\begin{aligned} \mathbf{P}(s) &= \sum_{i=1}^{M} \lambda_i^s\ \mathbf{B}^{(i)}\ \mathbf{P}(0) \\ &= \mathbf{P}^{\mathrm{st}} + \sum_{i=2}^{M} \lambda_i^s\ \mathbf{B}^{(i)}\ \mathbf{P}(0) \end{aligned} \tag{69}$$

Independent on the initial distribution $\mathbf{P}(0)$, the probability distribution approaches the stationary distribution $\mathbf{P}^{\mathrm{st}}$ at long times, for sufficiently number of steps s,

$$\mathbf{P}(s \to \infty) = \mathbf{P}^{\mathrm{st}} \tag{70}$$

and the rate of approach is determined by the eigenvalues λ_i with $2 \leq i \leq M$, which all have absolute values less than one, $|\lambda_i| < 1$.

Example MC8 Going back to the daily weather in Göteborg in examples MC3 and MC4, we can now give the explicit solution to the evolution equation. First we have to determine all eigenvalues. These are given by the equation $\det(\mathbf{W} - \lambda\mathbf{I}) = 0$. We obtain

$$\lambda_1 = 1 \ , \ \lambda_2 = 1/4 \ , \ \lambda_3 = -1/4$$

Next we need the eigenvectors. They can be obtained by solving Eq. (60) and Eq. (61) for the right and left eigenvectors, respectively, and the result is

$$|\chi_1\rangle = \begin{bmatrix} 1/5 \\ 2/5 \\ 2/5 \end{bmatrix} \qquad |\chi_2\rangle = \begin{bmatrix} 0 \\ 1/2 \\ -1/2 \end{bmatrix} \qquad |\chi_3\rangle = \begin{bmatrix} 1/5 \\ -1/10 \\ -1/10 \end{bmatrix}$$

$$\langle\chi_1| = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \qquad \langle\chi_2| = \begin{bmatrix} 0 & 1 & -1 \end{bmatrix} \quad \langle\chi_3| = \begin{bmatrix} 4 & -1 & -1 \end{bmatrix}$$

The $\mathbf{B}$ matrices can then be constructed

$$\mathbf{B}^{(2)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1/2 & -1/2 \\ 0 & -1/2 & 1/2 \end{bmatrix} \qquad \mathbf{B}^{(3)} = \begin{bmatrix} 4/5 & -1/5 & -1/5 \\ -2/5 & 1/10 & 1/10 \\ -2/5 & 1/10 & 1/10 \end{bmatrix}$$

and finally we get for the evolution of the probability distribution the following explicit expression

$$\mathbf{P}(s) = \begin{bmatrix} 0.2 \\ 0.4 \\ 0.4 \end{bmatrix} + \left(\frac{1}{4}\right)^s \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1/2 & -1/2 \\ 0 & -1/2 & 1/2 \end{bmatrix} \mathbf{P}(0) + \left(-\frac{1}{4}\right)^s \begin{bmatrix} 4/5 & -1/5 & -1/5 \\ -2/5 & 1/10 & 1/10 \\ -2/5 & 1/10 & 1/10 \end{bmatrix} \mathbf{P}(0)$$

This expression can be used to derive the same numerical sequences as in example MC4.

B Error estimate

It is important to find error bounds associated with evaluated quantities in a simulation. Consider a variable f. Assume that M measurements have been made $\{f_i\}$ and denote the average as

$$I = \frac{1}{M}\sum_{i=1}^{M} f_i. \tag{71}$$

We would like to determine the error bounds for I, its variance. If the values $\{f_i\}$ are independent on each others, *i.e.* uncorrelated data, the variance for I is given by $\mathrm{Var}[I] = \frac{1}{M}\mathrm{Var}[f]$ where

$$\mathrm{Var}[f] = \sigma^2(f) = \left\langle (f - \langle f\rangle)^2 \right\rangle = \left\langle f^2 \right\rangle - \langle f\rangle^2 . \tag{72}$$

However, in a simulation subsequent data are often highly correlated. The variance of I will depend on the number of independent samples M_{eff} generated by the simulation. We can introduce the *statistical inefficiency* n_s according to

$$M_{\mathrm{eff}} = M/n_s.$$

The variance of I can then be written as

$$\mathrm{Var}[I] = \frac{1}{M_{\mathrm{eff}}}\mathrm{Var}[f] = \frac{n_s}{M}\mathrm{Var}[f] \tag{73}$$

and the problem is reduced to determine n_s. We will consider two methods, one based on a direct evaluation of the corresponding correlation function and one based on data blocking.

Correlation function The variance of I can be written as

$$\begin{aligned}
\mathrm{Var}[I] &= \left\langle \left(\frac{1}{M}\sum_{i=1}^{M} f_i - \langle f\rangle \right)^2 \right\rangle \\
&= \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} \left[\langle f_i f_j\rangle - \langle f\rangle^2 \right].
\end{aligned}$$

If the data are uncorrelated we have that $\langle f_i f_j\rangle - \langle f\rangle^2 = [\langle f^2\rangle - \langle f\rangle^2]\delta_{ij}$ and $\mathrm{Var}[I] = \frac{1}{M}\mathrm{Var}[f]$. If the data are correlated we introduce the correlation function

$$\Phi_k = \frac{\langle f_i f_{i+k}\rangle - \langle f\rangle^2}{\langle f^2\rangle - \langle f\rangle^2}. \tag{74}$$

This is normalized such that

$$\Phi_{k=0} = 1.$$

We also assume that we study a stationary system and hence $\Phi_k = \Phi_{-k}$. For large k, $k > M_c$, Φ_k will decay to zero,

$$\Phi_{k>M_c} \to 0.$$

We assume that the total length of the simulation is considerably longer, $M > M_c$. By introducing $k = i - j$ we can now write the variance as

$$\begin{aligned}
\mathrm{Var}[I] &= \frac{1}{M^2} \sum_{i=1}^{M} \sum_{k=-(M-1)}^{M-1} \left(1 - \frac{\mid k \mid}{M}\right) \left[\langle f_i f_{i+k} \rangle - \langle f \rangle^2\right] \\
&= \mathrm{Var}[f] \, \frac{1}{M^2} \sum_{i=1}^{M} \sum_{k=-(M-1)}^{M-1} \left(1 - \frac{\mid k \mid}{M}\right) \Phi_k \\
&= \mathrm{Var}[f] \, \frac{1}{M^2} \sum_{i=1}^{M} \sum_{k=-M_c}^{M_c} \Phi_k \\
&= \mathrm{Var}[f] \, \frac{1}{M} \sum_{k=-M_c}^{M_c} \Phi_k
\end{aligned}$$

and hence

$$n_s = \sum_{k=-M_c}^{M_c} \Phi_k. \tag{75}$$

By comparing with the definition of a relaxation time τ_{rel}, $\Phi(t) = \exp(-t/\tau_{rel})$, we find that the statistical inefficiency n_s is equal to 2 times the relaxation time

$$n_s = 2\tau_{rel} \tag{76}$$

If we assume that the correlation function decays exponentially, $\Phi_k = \exp(-k/\tau_{rel})$, we notice that

$$\Phi_{k=n_s} = e^{-2} = 0.135 \sim 0.1$$

The statistical inefficiency can then be determined as the "time" when the corresponding correlation function has decayed to about 10% of its initial value.

Block averaging Another way to determine the statistical inefficiency n_s is to use so called *block averaging*. Divide the total length M of the simulation into M_B blocks of size B,

$$M = BM_B.$$

Determine the average in each block

$$F_j = \frac{1}{B} \sum_{i=1}^{B} f_{i+(j-1)B} \quad \text{for } j = 1, \ldots, M_B \tag{77}$$

and the corresponding variance Var[F]. If the block size B is larger than n_s $\{F_j\}$ will be uncorrelated and hence

$$\mathrm{Var}[I] = \frac{1}{M_B}\mathrm{Var}[F] \quad \text{if } B > n_s.$$

However, if the block size is smaller than s we have that

$$\mathrm{Var}[I] > \frac{1}{M_B}\mathrm{Var}[F] \quad \text{if } B < n_s$$

and we obtain the following relation for n_s

$$\begin{aligned} \mathrm{Var}[I] &\geq \frac{1}{M_B}\mathrm{Var}[F] \\ \frac{n_s}{M}\mathrm{Var}[f] &\geq \frac{B}{M}\mathrm{Var}[F] \\ n_s &\geq \frac{B\mathrm{Var}[F]}{\mathrm{Var}[f]} \end{aligned}$$

We can then obtain the statistical inefficiency

$$n_s = \lim_{B\ large} \frac{B\mathrm{Var}[F]}{\mathrm{Var}[f]} \tag{78}$$

by plotting $B\mathrm{Var}[F]/\mathrm{Var}[f]$ as function of the block size B. In Fig. 1 we show a typical result from a simulation.

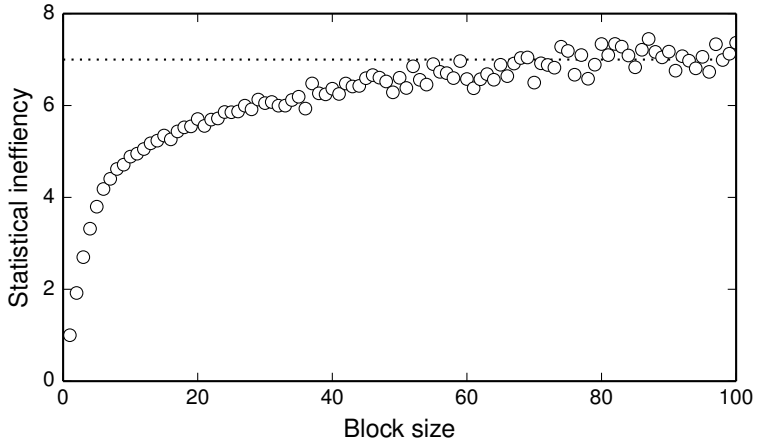


Figure 1: Illustration of the calculation of n_s using block averaging. The figure shows the approach to the plateau value $s = 7$ when the block size B is increased.

References

[1] J. M. Hammmersley and D. C. Handscomb, *Monte Carlo Methods*, Methuen, London, 1964.

[2] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21** 1087 (1953).

[3] *Handbook of Markov Chain Monte Carlo*, Eds. S. Brooks, A. Gelman, G. L. Jones, X.-L. Meng, Chapman and Hall, CRC, 2011.

[4] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes, The Art of Scientific Computing*, 3rd ed., Cambridge, 2007.

[5] S. K. Park and K. W. Miller, Communications of the ACM **31**, 1192 (1988).

[6] D. Frenkel and B. Smit, *Understanding Molecular Simulation*, 2nd ed., Academic Press, 2002.

[7] W. Feller, *An Introduction to Probability Theory and Its Applications*, Wiley, 1957.

[8] V. I. Manousiouthakis and M. W. Deem, J. Chem. Phys. **110**, 2753 (1999).

[9] H. Suwa and S. Todo, Phys. Rev. Lett. **105** 120603 (2010).

[10] G. O. Roberts, A. Gelman, W. R. Gilks, Ann. Appl. Probab. **7** 110 (1997).

[11] W. W. Wood and J. D. Jacobson, Proc. of the Western Joint Computer Conference (San Fransisco) 261 (1959).

[12] A. A. Barker, Aust. J. Phys. **18** 119 (1965).

[13] W. K. Hastings, Biometrika **57**, 97 (1970).

[14] M. P. Allen and D. J. Tildesley, *Computer Simulations of Liquids*, Clarendon Press, 1989.

[15] L. E. Reichl, *A Modern Course in Statistical Physics*, 2nd ed., Wiley, 1998.