

Monte Carlo Methods in Statistical Physics: II. Dynamic MC and correlated data

Andrea Pelissetto

Andrea.Pelissetto@roma1.infn.it

Dipartimento di Fisica
Università di Roma “La Sapienza”

Let us go back to our original problem.

Example: A gas. Consider a gas (fluid) of N molecules in a box of volume V . The **configurations** of the gas are specified by the set of positions $\mathbf{r}_1, \mathbf{r}_2, \dots$ (each vector defines a position inside the box) and of momenta $\mathbf{p}_1, \mathbf{p}_2, \dots$. The **configuration space** is the phase space available to the system.

For instance, we wish to compute the average **energy at constant volume** defined by

$$E = \frac{1}{Z} \int [dp][dr] H e^{-\beta H},$$

where the integral is over the whole configuration space (all values of the momenta and all positions inside the box), $\beta = 1/(k_B T)$ and Z is the **partition function**.

The integral over momenta is simple (H is quadratic in the momenta).

Thus, we need only to compute the **configurational energy**

$$E_{\text{conf}} = \frac{1}{Q} \int [dr] U e^{-\beta U},$$

where U is the potential energy and Q is the **configurational partition function**.

In principle, one could imagine of using the following MC algorithm to compute E_{conf} :

1. If the box is cubic, of size L , let $\mathbf{r}_1 = (LU_1, LU_2, LU_3)$, where U_i are random numbers uniformly distributed in $[0,1]$. Repeat the same for all N particles;
2. Compute $n_i = U e^{-\beta U}$ and $d_i = e^{-\beta U}$
3. Repeat steps 1 and 2 n_{iter} times. An estimate of E_{conf} is $\sum_i n_i / \sum_i d_i$.

This algorithm is **correct**, but, for any reasonable n_{iter} , the **error** on the estimate is **enormous**.

The method does NOT work for the very same reason why thermodynamics works.

Consider a system of N particles at fixed temperature T . The energy distribution is peaked around the mean energy E_{conf} .

The **width** of the distribution is of order $1/\sqrt{N}$.

It implies that the only configurations that contribute to the integral are those for which $E_{\text{conf}} - w/\sqrt{N} < U < E_{\text{conf}} + w/\sqrt{N}$.

But, our algorithm generates configurations with any value of U : most of them will be outside the relevant interval (which shrinks as N increases) and thus they will not be relevant for the computation of E_{conf} . The result will depend on the very few configurations that belong to the relevant interval and thus the error will be large.

WAY OUT: use **importance sampling**. Configurations are not generated randomly, but according to the Gibbs measure.

Unfortunately, it is very difficult to do it directly, except in a very few systems: dilute polymers (growth algorithms, dimerization), Gaussian models, percolation...

We need a new method: the **dynamic Monte Carlo method**, which mathematically is nothing but a **Markov process**.

A **Markov chain** is specified by a **state space** S and by a **transition matrix** P_{xy} , $x, y \in S$ such that

$$P_{xy} \geq 0, \quad \sum_{y \in S} P_{xy} = 1$$

Two basic features:

- 1) the probability to be in y at time $t + 1$ depends only on the position at time t ;
- 2) the probabilities are time independent.

Another example: Let us consider two Ising spins s_1, s_2 which can assume the values ± 1 . The dynamics consists in choosing randomly one spin and flipping it with probability p .

State space: the space of all possible spin configurations. Since each spin assumes two values, there are four possibilities: $(s_1, s_2) = (1, 1), (1, -1), (-1, 1), (-1, -1)$.

Transition matrix $P_{s_1s_2,s'_1s'_2}$.

1) Since at each step only one spin is changed, $P_{s_1s_2,s'_1s'_2} = 0$ if $s_1 \neq s'_1$ and $s_2 \neq s'_2$.

2) Suppose now that $s_1 = s'_1$ and let us compute the probability that $s_2 \neq s'_2$. This occurs when s_2 is chosen (it occurs with probability $1/2$) and then flipped (probability p). Hence $P_{s_1s_2,s_1s'_2} = p/2$ for $s_2 \neq s'_2$.

3) Analogously $P_{s_1s_2,s'_1s_2} = p/2$ for $s_1 \neq s'_1$.

4) Finally, $P_{s_1s_2,s_1s_2}$ can be computed from $\sum_{y \in \mathcal{S}} P_{xy} = 1$:
 $P_{s_1s_2,s_1s_2} = 1 - p$.

relevant family of Markov chains

Among all Markov chains we will consider processes which satisfy two conditions:

1) **Ergodicity** (in the mathematical literature it is called irreducibility). We can go from any state to any state:

For any x, y , there is $n > 0$ such that $P_{xy}^n > 0$

2) **Aperiodicity** (technical, not very interesting condition): **the greatest common divisor of the set of integers n such that $P_{xx}^n > 0$ is 1.**

1) Write the transition matrix for the spin system:

$$\begin{bmatrix} 1-p & p/2 & p/2 & 0 \\ p/2 & 1-p & 0 & p/2 \\ p/2 & 0 & 1-p & p/2 \\ 0 & p/2 & p/2 & 1-p \end{bmatrix}$$

2) Using the stationarity condition, show that, for any $p > 0$, the equilibrium distribution is $\pi = (1/4, 1/4, 1/4, 1/4)$.

3) Show that the Markov chain is ergodic for any $p > 0$.

4) Show that the Markov chain is aperiodic for $p < 1$, periodic (of period 2) for $p = 1$.

fundamental theorem

If P is irreducible and aperiodic:

a)

$$\lim_{n \rightarrow \infty} P_{xy}^n = \pi_y$$

with π_y nonnegative.

b) If π_x is not identically zero, it satisfies

$$\sum_x \pi_x = 1.$$

Hence π_x is a probability distribution on the state space.

c) If π_x is not identically zero, it satisfies the **stationarity condition**

$$\sum_x \pi_x P_{xy} = \pi_y.$$

Hence π_x is the **equilibrium distribution**.

fundamental theorem

- d) (**Uniqueness**) π_x is the unique probability distribution satisfying the stationarity condition.
- e) (**Ergodic theorem**) Consider the process $X_0 \rightarrow X_1 \rightarrow X_2 \dots \rightarrow X_N$ generated by P . Then

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N f(X_n) = \sum_x f(x) \pi_x$$

for any function $f(x)$ defined on the state space, irrespective of X_0 .

In Markov-chain theory the emphasis is on P which determines the equilibrium distribution π .

In our Monte Carlo applications we work in the opposite way.

The equilibrium distribution π is known: π should be identified with the (normalized) Gibbs measure.

Then, we devise a transition matrix P such that π satisfies the stationarity condition for P .

The uniqueness theorem guarantees that π is the equilibrium distribution of the process.

Averages over the Gibbs measure can be computed by using averages over the Markov process (ergodic theorem).

IMPORTANT: The Gibbs measure is defined by $\exp(-\beta H)$. In order to identify it with π , we should multiply it by a normalization constant, the **partition function**:

$$\pi = \frac{1}{Z} e^{-\beta H}, \quad Z = \sum_{x \in \mathcal{S}} e^{-\beta H(x)}$$

The determination of the quantity Z requires complex techniques. However, Z is **NOT** needed to devise the transition matrix P .

The stationarity condition can be written as

$$\sum_x \pi_x P_{xy} = \pi_y \Rightarrow \sum_x \frac{e^{-\beta H(x)}}{Z} P_{xy} = \frac{e^{-\beta H(y)}}{Z} \Rightarrow \sum_x e^{-\beta H(x)} P_{xy} = e^{-\beta H(y)}.$$

The partition function Z drops out!

There is an infinite number of matrices P that satisfy the stationarity condition. It is often easier to look for a matrix P which satisfies the stronger condition

$$\pi_x P_{xy} = \pi_y P_{yx}$$

for any x, y . This condition is called **reversibility condition** or **detailed-balance condition**.

Let us prove that if P is reversible, then it satisfies the stationarity condition. Summing over x the detailed-balance condition and using the fact that $\sum_x P_{yx} = 1$ we have

$$\sum_x \pi_x P_{xy} = \sum_x \pi_y P_{yx} \quad \Rightarrow \quad \sum_x \pi_x P_{xy} = \pi_y.$$

Static MC algorithms

The Monte Carlo algorithms we have considered before are **static** algorithms.

They can also be put in the present framework.

At each iteration a point in the sample space is chosen with probability π_x .

Hence, in these algorithms $P_{xy} = \pi_y$, independently of x .

As we discussed, the ergodic theorem states that

time averages = sample averages

One can prove that:

1) For any starting condition the time average

$$\frac{1}{N+1} \sum_{n=0}^N f(X_n)$$

is a **biased** estimate of the sample average $\sum_x \pi_x f(x)$. The bias is of order $1/N$.

2) The bias vanishes if one **starts in equilibrium**.

We wish to compute for $n \rightarrow \infty$ the average

$$I_N = \left\langle \frac{1}{N+1} \sum_{n=0}^N f(X_n) - \sum_x \pi_x f(x) \right\rangle_0$$

where $\langle \cdot \rangle_0$ indicates the average over all processes that start in X_0 .

The probability of being in point y at $t = n$ is P_{0y}^n , so that

$$\langle f(X_n) \rangle_0 = \sum_y f(y) P_{0y}^n$$

Therefore

$$\begin{aligned} I_N &= \frac{1}{N+1} \sum_{n=0}^N \sum_y [P_{0y}^n f(y) - \pi_y f(y)] \\ &= \frac{1}{N+1} \sum_y f(y) \sum_{n=0}^N [P_{0y}^n - \pi_y] \end{aligned}$$

Now, introduce the projector $\Pi_{xy} = \pi_y$. It has the properties:

$$(\Pi P)_{xy} = \sum_z \Pi_{xz} P_{zy} = \sum_z \pi_z P_{zy} = \pi_y = \Pi_{xy}$$

$$(P \Pi)_{xy} = \sum_z P_{xz} \Pi_{zy} = \sum_z P_{xz} \pi_y = \pi_y = \Pi_{xy}$$

$$(\Pi \Pi)_{xy} = \sum_z \Pi_{xz} \Pi_{zy} = \sum_z \pi_z \pi_y = \pi_y = \Pi_{xy}$$

Then, for $k \neq 0$, since P and Π commute, we can write

$$(P - \Pi)^k = \sum_{n=0}^k \binom{k}{n} (-1)^{k-n} P^n \Pi^{k-n} = P^k + \Pi \sum_{n=0}^{k-1} \binom{k}{n} (-1)^{k-n} = P^k - \Pi$$

Therefore

$$I_N = \frac{1}{N+1} \sum_y f(y) \left[I - \Pi + \sum_{n=1}^N (P - \Pi)_{0y}^n \right]$$

Now, a general theorem states that the eigenvalues λ_i of an irreducible aperiodic transition matrix satisfy the conditions:

- 1) $|\lambda_i| \leq 1$;
- 2) there is only one eigenvalue on the unit circle: it has $\lambda_i = 1$, the left eigenvector is π_x , the right eigenvector is $(1, \dots, 1)$.

al conditions: the proof. 3

Now, Π is the projector onto the eigenvector of eigenvalue 1, so that all eigenvectors of $P - \Pi$ lie strictly inside the unit circle.

Consequences:

- 1) $(P - \Pi)_{0y}^n$ behaves as $|\lambda_2|^n$ as $n \rightarrow \infty$, where λ_2 is the second-largest (in absolute value) eigenvalue. Since $|\lambda_2| < 1$, the matrix element vanishes exponentially for $n \rightarrow \infty$.
- 2) The matrix $I - (P - \Pi)$ has an inverse (there are some caveats for infinite-dimensional systems, which have however little practical interest).

Therefore, discarding exponentially small terms we can write

$$I_N \approx \frac{1}{N+1} \sum_y f(y) \left[I - \Pi + \sum_{n=1}^{\infty} [(P - \Pi)_{0y}^n] \right] = \frac{1}{N+1} \sum_y f(y) [(I - P + \Pi)^{-1} - \Pi]_{0y}$$

Therefore, I_N is a **biased** estimate of the sample average. The bias is of order $1/N$.

Let us now suppose that we pick X_0 with probability π_x . Then

$$\bar{I}_n = \sum_{z \in S} \pi_z \left\{ \frac{1}{N+1} \sum_{y \in S} f(y) \sum_{n=0}^N [P_{zy}^n - \pi_y] \right\}$$

$$\bar{I}_n = \frac{1}{N+1} \sum_{y \in S} f(y) \sum_{n=0}^N \left[\sum_{z \in S} (\pi_z P_{zy}^n - \pi_z \pi_y) \right] = 0$$

because of the stationarity condition and of $\sum_{z \in S} \pi_z = 1$.

There is no bias if we start in equilibrium.

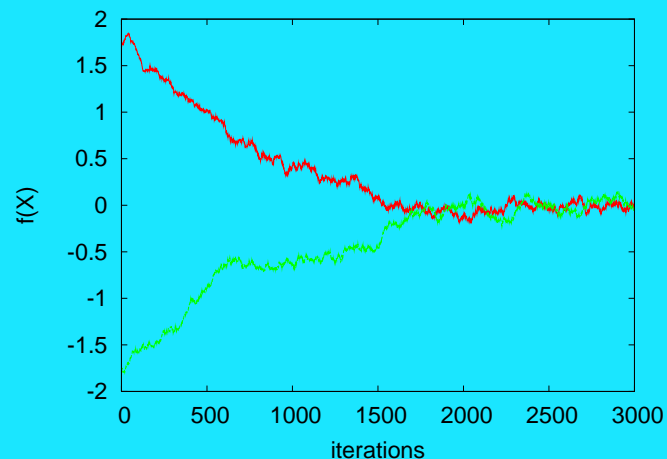
thermalization: practical considerations

Initial-condition effects vanish as $1/N$.

Statistical fluctuations vanish as $1/\sqrt{N}$.

Initial conditions introduce a bias in the estimates which is particularly important for short runs and which become irrelevant as the number of iterations increases.

Contrary to what one might think, in high-precision Monte Carlo studies (of non disordered systems!!!) there is no need to be very careful about thermalization.



To avoid the bias one looks at the data and discard all data that do not appear in equilibrium. For instance, in the case of the figure, one can discard 2000 iterations.

Useful check: repeat the simulation starting from two very different configurations and check when the two runs provide results that roughly agree.

For a spin system, one can perform a simulation that start from a disordered configuration (at $t = 0$ the spins have random direction) and a second one starting from an ordered configuration (at $t = 0$ all spins are parallel).

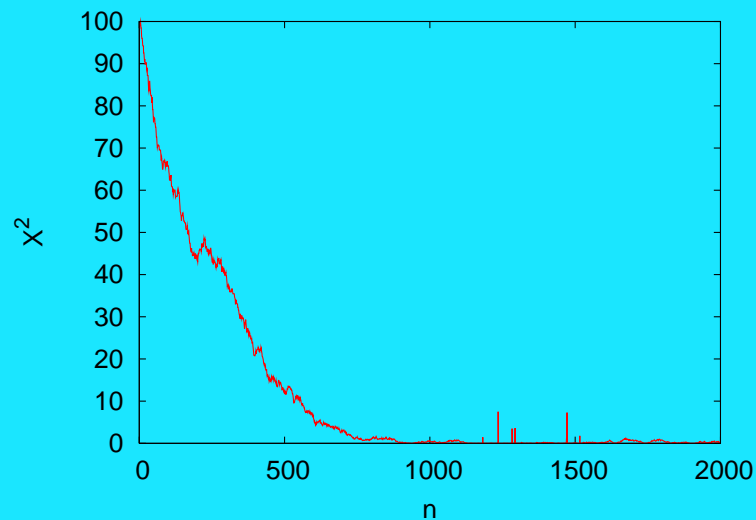
To clarify the role of the initial conditions let consider a simple example. We consider the Gaussian probability $e^{-x^2} / \sqrt{\pi}$ on the real line.

A simple dynamic Monte Carlo algorithm for this distribution is the following (as we shall see this is the **Metropolis** algorithm):

1. If X_n is the value of x at the n -th iterations, propose $Y = X_n + p(U - 1/2)$, where p is a constant and U a random number uniformly distributed in $[0, 1]$.
2. If $|Y| \leq |X_n|$ then $X_{n+1} = Y$. If $|Y| > |X_n|$ we draw a random number V uniformly distributed in $[0, 1]$. If $U \leq \exp(X_n^2 - Y^2)$, then $X_{n+1} = Y$; otherwise $X_{n+1} = X_n$.

The algorithm is correct for any $p > 0$. In the following we shall use the value $p = 0.2$. Moreover, we will start very far from equilibrium taking $X_0 = 10$.

Exercise: p can be optimized by finding the value that provides the smallest error.



We report here the quantity X^2 . Its average should converge to $1/2$. Equilibrium is reached after 1000 iterations. There is no bias if we discard the first 1000 iterations.

Now we report the average

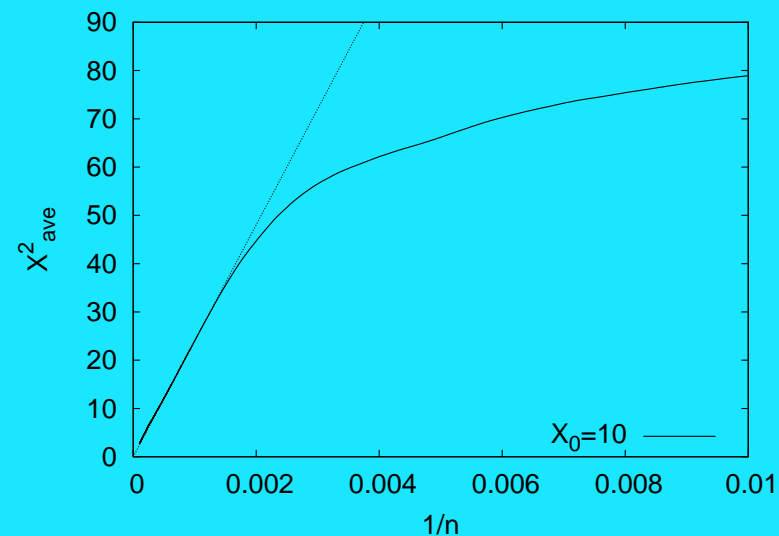
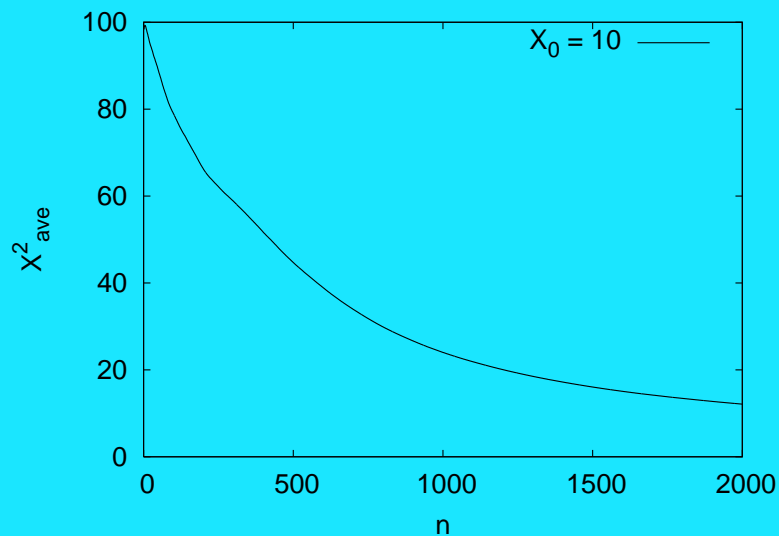
$$X_{\text{ave}}^2(n) = \frac{1}{n+1} \sum_{i=0}^n X_i^2$$

in which we keep all data.

Now we report the average

$$X_{\text{ave}}^2(n) = \frac{1}{n+1} \sum_{i=0}^n X_n^2$$

in which we keep all data.



The estimate $X_{\text{ave}}^2(n)$ converges to $1/2$ with $1/n$ corrections as expected (this behavior sets in after 1000 iterations).

As in the case of the numerical distributions we are interested in computing the error on the mean, i.e.

$$\Delta^2 = \left\langle \left(\frac{1}{N+1} \sum_{n=0}^N f(X_n) - \sum_x \pi_x f(x) \right)^2 \right\rangle_0$$

where, as before, the average is taken over all possible histories that start at X_0 .

THEOREM: The limit

$$\lim_{N \rightarrow \infty} N\Delta^2$$

is finite and independent of the initial configuration.

Thus, also in the case of dynamic Monte Carlo fluctuations (errors) decrease as $1/\sqrt{N}$.

We wish now to estimate Δ^2 from the data. If

$$F = \sum_x \pi_x f(x)$$

we have

$$\Delta^2 = \frac{1}{(N+1)^2} \sum_{n=0}^N \sum_{m=0}^N \langle (f(X_n) - F)(f(X_m) - F) \rangle_0$$

Now we define the **autocorrelation function** $C_f(n, m)$:

$$C_f(n, m; X_0) = \langle (f(X_n) - F)(f(X_m) - F) \rangle_0$$

so that

$$\Delta^2 = \frac{1}{(N+1)^2} \sum_{n=0}^N \sum_{m=0}^N C_f(n, m; X_0)$$

The autocorrelation function depends on X_0 , while the large- N limit of Δ does not.

We now assume we start in equilibrium.

The **equilibrium autocorrelation function** is time-translation invariant, hence depends only on $|n - m|$:

$$C_f(n, m; \text{eq}) = C_f(|n - m|).$$

$$\Delta^2 = \frac{1}{(N+1)^2} \sum_{k=-N}^N (N+1-k) C_f(|k|) = \frac{1}{N+1} \sum_{k=-N}^N \left(1 - \frac{k}{N+1}\right) C_f(|k|)$$

The function $C_f(|k|)$ decays exponentially. Hence for $N \rightarrow \infty$ we obtain

$$\Delta^2 = \frac{1}{N+1} \left[C_f(0) + 2 \sum_{k=1}^{\infty} C_f(k) \right].$$

Observations:

- 1) $C_f(0)$ is the variance, the only term present in the static algorithms we considered before.
- 2) We define an **integrated autocorrelation time** $\tau_{\text{int},f}$ as

$$\tau_{\text{int},f} = \frac{1}{2} + \sum_{k=1}^{\infty} \frac{C_f(k)}{C_f(0)}.$$

The error becomes

$$\Delta^2 = \frac{C_f(0)}{N+1} 2\tau_{\text{int},f}.$$

3) The definition of $\tau_{\text{int},f}$ can be understood by considering the simple case in which $C_f(k) = C_f(0) \exp(-k/\tau)$:

$$\tau_{\text{int},f} = -\frac{1}{2} + \frac{1}{1 - e^{-1/\tau}} \approx \tau$$

where the last equality holds for $\tau \gg 1$.

The determination of $\tau_{\text{int},f}$ is a tricky business, which is crucial in dynamic studies of the critical behavior. However, in order to compute the error bars, we can avoid it.

We wish now to show how $C_f(n, m)$ can be related to the transition matrix P .

In terms of the transition matrix we can write (for $n < m$)

$$C_f(n, m) = \langle (f(X_n) - F)(f(X_m) - F) \rangle_0 = \sum_{xy} (f(x) - F)(f(y) - F) P_{0x}^n P_{xy}^{m-n}.$$

If we start in equilibrium

$$C_{f,\text{eq}}(n, m) = \sum_z \pi_z \sum_{xy} (f(x) - F)(f(y) - F) P_{zx}^n P_{xy}^{m-n} = \sum_{xy} \pi_x (f(x) - F)(f(y) - F) P_{xy}^{m-n},$$

so that the autocorrelation function depends only on the difference $m - n$. Using the stationarity condition we can rewrite it as

$$\begin{aligned} & \sum_{xy} \pi_x f(x) (f(y) - F) P_{xy}^{m-n} - F \sum_{xy} \pi_x P_{xy}^{m-n} (f(y) - F) \\ &= \sum_{xy} \pi_x f(x) (f(y) - F) P_{xy}^{m-n} - F \sum_y \pi_y (f(y) - F) \\ &= \sum_{xy} \pi_x f(x) (f(y) - F) P_{xy}^{m-n} \\ &= \sum_{xy} \pi_x f(x) f(y) P_{xy}^{m-n} - F \sum_x \pi_x f(x) \sum_y P_{xy}^{m-n} = \sum_{xy} \pi_x f(x) f(y) P_{xy}^{m-n} - F^2 \end{aligned}$$

Defining the projector $\Pi_{xy} = \pi_y$ we have

$$\begin{aligned}
 \sum_{xy} \pi_x f(x) f(y) P_{xy}^{m-n} - F^2 &= \sum_{xy} \pi_x f(x) f(y) P_{xy}^{m-n} - \sum_{xy} \pi_x \pi_y f(x) f(y) = \\
 &= \sum_{xy} \pi_x f(x) f(y) [P_{xy}^{m-n} - \Pi_{xy}] \\
 &= \sum_{xy} \pi_x f(x) f(y) [P - \Pi]_{xy}^{m-n}
 \end{aligned}$$

As we already discussed all eigenvalues of $P - \Pi$ lie in the unit circle (this is correct for a finite system; for an infinite one some caveats are needed). Hence, the autocorrelation function decays exponentially as $|\lambda_2|^{m-n}$ as $m-n \rightarrow \infty$, where λ_2 is the second largest (in absolute value) eigenvalue of P .

Estimating the error

The correct estimation of the statistical error is difficult. Here I will assume that we have a very large statistical sample. In this case, there is a very simple method.

To illustrate we present an **EXAMPLE**.

Suppose $X_0 \rightarrow X_1 \rightarrow X_2 \rightarrow \dots$ is the set of configurations generated in the Monte Carlo.

The **measures** are $f_n = f(X_n)$.

Assume for the example that the autocorrelation function $C_f(n)$ is simply

$$C_f(n) = \exp(-n/\tau) \quad \tau = 10$$

The error is

$$\Delta^2 = \frac{1}{N} \left[C_f(0) + 2 \sum_k C_f(k) \right] = 20.0/N$$

while the variance is $C_f(0) = 1$. As expected $\Delta^2/C_f(0) = 20 = 2\tau$.

Then, we introduce some new variables:

$$\begin{aligned} Y_n^{(0)} &= f_n \\ Y_n^{(1)} &= \frac{1}{2}(Y_{2n}^{(0)} + Y_{2n+1}^{(0)}) = \frac{1}{2}(f_{2n} + f_{2n+1}) \\ Y_n^{(2)} &= \frac{1}{2}(Y_{2n}^{(1)} + Y_{2n+1}^{(1)}) \\ &\dots\dots \end{aligned}$$

For instance

$$Y_0^{(1)} = \frac{1}{2}(f_0 + f_1), \quad Y_1^{(1)} = \frac{1}{2}(f_2 + f_3), \quad Y_2^{(1)} = \frac{1}{2}(f_4 + f_5), \dots$$

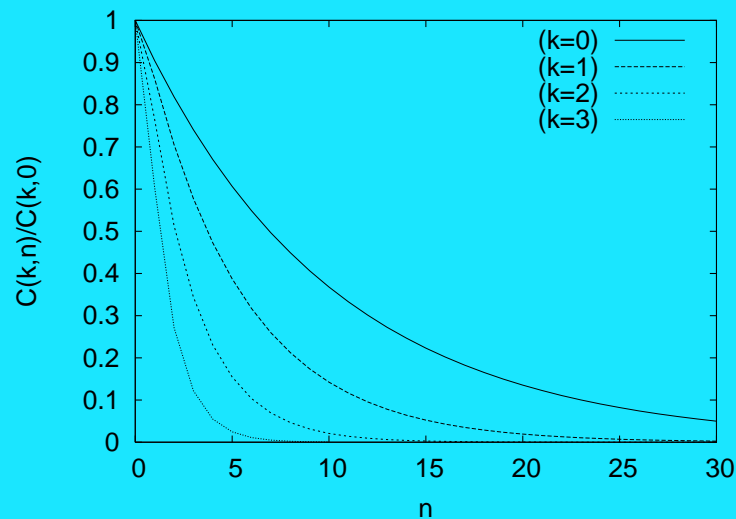
$$Y_0^{(2)} = \frac{1}{2}(Y_0^{(1)} + Y_1^{(1)}) = \frac{1}{4}(f_0 + f_1 + f_2 + f_3),$$

$$Y_1^{(2)} = \frac{1}{2}(Y_2^{(1)} + Y_3^{(1)}) = \frac{1}{4}(f_4 + f_5 + f_6 + f_7),$$

The average of each set $Y_k^{(n)}$ gives the same result:

$$\sum_x \pi_x f(x) \approx \frac{1}{N} \sum f_n = \frac{1}{N/2} \sum Y_n^{(1)} = \frac{1}{N/4} \sum Y_n^{(2)} = \frac{1}{N/8} \sum Y_n^{(3)} = \dots$$

But the autocorrelation functions decays faster.



We report $C(k,n)/C(k,0)$ which is the normalized autocorrelation function of $Y_n^{(k)}$. The decay is clearly faster.

We expect that for $k \gg \tau$ there is essentially no autocorrelation so that we can estimate the error by simply taking the variance.

The error formula for $Y_n^{(k)}$ is

$$\Delta^2 = \frac{2^k}{N} [C(k, 0) + 2 \sum_n C(k, n)]$$

Therefore, a practical method consists in increasing k until $2^k C(k, 0) \approx 2^{k+1} C(k+1, 0)$ (the autocorrelation function is negligible). Then

$$\Delta^2 = 2^k C(k, 0) / N.$$

Let us see how it works in our specific example.

k	$2^k C(k, 0)$	$2^k [C(k, 0) + 2C(k, 1)]$
1	1.90	5.19
2	3.55	8.98
3	6.26	13.84
4	10.05	18.00
5	14.03	19.77
6	16.90	20.01
7	18.46	20.02
8	19.23	20.02

The quantity $2^k C(k, 0)$ converges towards 20.0, and thus it provides the correct error.

NOTE: CONVERGENCE IS SLOW!

For $k = 6$, i.e. $2^k = 64 \approx 6\tau$, the error is underestimated by 17%. Moreover, $2^k C(k, 0)$ is not yet k independent even for $k = 8$ (i.e. when we average data on time intervals of $2^8 = 256 \approx 25\tau$)

A SUGGESTION: Use $\Delta^2 = 2^k [C(k, 0) + 2C(k, 1)]$: convergence is significantly faster. Even with this trick, we need k such that $2^k \approx 10\tau$. Since we need at least 100 $Y_n^{(k)}$ in order to estimate the variance and the one-step autocorrelation function with reasonable accuracy, the method requires more than $100 \times 10\tau$, i.e. 1000 independent configurations.

If we have some idea of the autocorrelations we can just measure every $\sim \tau$ iterations.

EXAMPLE: again the case $C(k) = \exp(-k/\tau)$, $\tau = 10$.

If we measure every n iterations, the rescaled variance $N\Delta^2$ (N is the total number of **iterations**, not measures) is given by

$$n = 1 \quad \Delta^2 = 20.0/N$$

$$n = 10 \quad \Delta^2 = 21.6/N$$

$$n = 20 \quad \Delta^2 = 26.3/N$$

$$n = 30 \quad \Delta^2 = 33.1/N$$

$$n = 40 \quad \Delta^2 = 41.5/N$$

Even measuring every $20 = 2\tau$ iterations we only loose a factor of $\sqrt{26/20} = 1.14$ on the errors. But, $n = 40$ is clearly too much: half of the statistics is lost.

Metropolis algorithm

The Metropolis algorithm is a general purpose algorithm, which can be applied to essentially any problem.

Let us recall our problem: given a probability distribution π on a state space S , we wish to determine a transition matrix P which has π as equilibrium distribution.

In practice, we require P to satisfy the detailed-balance condition $\pi_x P_{xy} = \pi_y P_{yx}$.

The Metropolis transition matrix is the product of the proposal transition matrix $P^{(0)}$ and of the acceptance matrix A . Explicitly

$$\begin{aligned} P_{xy} &= P_{xy}^{(0)} A_{xy} & x \neq y \\ P_{xx} &= 1 - \sum_{y \neq x} P_{xy} = P_{xx}^{(0)} + \sum_{y \neq x} P_{xy}^{(0)} (1 - A_{xy}). \end{aligned}$$

The matrix A satisfies:

- 1) since $P_{xy} \geq 0$, we must have $A_{xy} \geq 0$;
- 2) we require $A_{xy} \leq 1$, to guarantee $P_{xx} \geq 0$.

Moreover, necessary (but not sufficient) condition for P to be ergodic is that $P^{(0)}$ is ergodic.

The detailed-balance condition requires that

$$\frac{A_{xy}}{A_{yx}} = \frac{\pi_y P_{yx}^{(0)}}{\pi_x P_{xy}^{(0)}}$$

for all pairs $x \neq y$.

The Metropolis choice consists in taking

$$A_{xy} = F(R), \quad R = \frac{\pi_y P_{yx}^{(0)}}{\pi_x P_{xy}^{(0)}}$$

where $F(x)$ is a function in $[0, 1]$ such that

$$\frac{F(x)}{F(1/x)} = x, \quad F(x) = xF(1/x).$$

The fastest dynamics is obtained by taking the largest possible P_{xy} for $x \neq y$, hence the largest A_{xy} .

To obtain the maximal $F(x)$ note that, since $F(1/x) \leq 1$ we have $F(x) \leq x$, and therefore $F(x) \leq \min(x, 1)$.

The choice

$$F(x) = \min(x, 1)$$

is the Metropolis choice, which is therefore the optimal choice.

Algorithm: one Metropolis iteration works as follows.

1. At iteration n the system is in the state point x_n .
2. We generate a new state point y using the proposal matrix $P^{(0)}$.
3. We compute the ratio $R = \pi_y P_{yx}^{(0)} / (\pi_x P_{xy}^{(0)})$.
4. If $R \geq 1$ we accept the proposal and set $x_{n+1} = y$.
Otherwise, we generate a number U uniformly distributed in $[0, 1]$. If $U \leq R$, the proposal is accepted and $x_{n+1} = y$; in the opposite case a null transition is performed and $x_{n+1} = x_n$.

In many applications $P^{(0)}$ is symmetric and therefore $R = \pi_y/\pi_x$.

In statistical mechanics $\pi_x = e^{-\beta E_x}/Z$. In this case, for a symmetric proposal, the last two steps of the algorithm can be rewritten as follows:

3. Compute $\Delta E = E_y - E_x$;
4. If $\Delta E \leq 0$ accept the proposal and set $x_{n+1} = y$. Otherwise, generate a number U uniformly distributed in $[0, 1]$. If $U \leq \exp(-\Delta E)$, the proposal is accepted and $x_{n+1} = y$; in the opposite case a null transition is performed and $x_{n+1} = x_n$.

Exercise: Verify that $F(x) = x/(1+x)$ can also be used to define the acceptance matrix.

Exercise: Metropolis for the Ising model

The ferromagnetic Ising model Hamiltonian is

$$H = - \sum_{\langle ij \rangle} s_i s_j$$

where $s_i = \pm 1$ are defined on the sites of a lattice (for instance a cubic lattice) and the sum is over all nearest-neighbor pairs $\langle ij \rangle$.

One iteration of the algorithm works as follows:

1. choose randomly a site i ;
2. compute $R = \sum_j s_j$, where the sum runs over all nearest neighbors j of site i , and $\Delta = 2s_i R$.
3. If $\Delta \leq 0$, flip the spin at site i : the new configuration $\{s'_j\}$ is given by $s'_j = s_j$ for $j \neq i$, $s'_i = -s_i$. If $\Delta > 0$, generate a number U uniformly distributed in $[0, 1]$. If $U < e^{-\beta\Delta}$, flip the spin as explained before; otherwise perform a null transition

Show this is a correct algorithm.