

# Chapter 1

## Markov Chains

# Understanding the Metropolis-Hastings algorithm for simulating the Ising model


**Goal of this tutorial** is for students to get familiar with the basis of the theory behind the Metropolis-Hastings algorithm, and with its application to the numerical simulation of the Ising model. After this tutorial, students should:

- be more familiar with the notion of microscopic states VS macroscopic properties of a complex systems;
- become more confident on how to write a continuous time Markov chain;
- know what is the condition of *detailed balance*.
- become more familiar with the Boltzmann distribution and with the Ising model.
- understand where the Metropolis-Hastings algorithm comes from and know how to apply it for the Ising model.
- know how to compute ensemble averages from a Metropolis simulation of the system.

The Metropolis-Hastings algorithm is a well-known Markov Chain Monte Carlo (MCMC) algorithm that is commonly used to sample from probability distribution for which direct sampling is difficult (such as multi-variate probability distributions, like the Boltzmann distribution in Eq. (1.2)).

**Important comment about the notations:** Note that to match the notation used in the lectures and to avoid confusion with the external magnetic field (which is often denoted  $h$  or  $H$ ), we denote the energy as  $E(s)$  instead of  $H(s)$ . In the literature, you may conventionally find the energy expressed as the Hamiltonian of the system and denoted  $H(s)$ .

## 1.1 Markov chain for sampling from an equilibrium system

 One of the simplest way to study the behavior of a stochastic system is to resort to numerical simulation. To study the possible phases of a system at equilibrium, we must look at its macroscopic properties, which are obtained by computing the ensemble averages (or “thermal average”) of different quantities characterizing the system. By definition, the **ensemble average** of a quantity  $A(s)$  is given by the average of  $A(s)$  over all the states  $s$  accessible to the system at equilibrium:

$$\langle A(s) \rangle = \sum_s A(s) P(s) . \quad (1.1)$$

In this equation, the sum is over all the states  $s$  of the system, and  $P(s)$  is the probability to observe the system in the state  $s$  at equilibrium, which is often written under the form:

$$P(s) = \frac{\exp(-\beta E(s))}{\mathcal{Z}}, \quad (1.2)$$

where  $E(s)$  is the **energy**<sup>a</sup> of the system when it is in the state  $s$ , the parameter  $\beta = 1/T$  is the inverse temperature, and  $\mathcal{Z}$  is the **partition function**, which is obtained from the normalization of  $P(s)$ . The parameter  $\beta$  (or  $T$ ) can also be seen as controlling the noise level of the system: the larger is  $\beta$ , the smaller is the noise. In practice, it is not computationally efficient (and, most of the time, impossible) to compute all the terms of the sum in Eq. (1.1). Just the computation of the partition function itself would be already computationally demanding.

<sup>a</sup>In physics,  $E(s)$  is also known as the Hamiltonian of the system and is often denoted  $H(s)$  (instead of  $E(s)$ ).

**Q1.** For instance, in the context of the Ising model, the simplest quantity to look at is the local average magnetisation  $m_i = \langle s_i \rangle$ . What are the possible microstates  $s$  of a system of  $n$  spins in the Ising model? Using the definition of the ensemble average in Eq. (1.1), what is the expression of  $m_i$ ? What is  $A(s)$  in this case? what is  $E(s)$ ? and how is  $Z$  computed?

**Q2.** Consider an Ising model on a 1d-lattice with 20 sites: how many terms are there in the sum of Eq. (1.1) in that case? How many terms are there in the sum to compute  $Z$ ?

**What can we do?** If one was able to generate many random states of the system with their respective probabilities<sup>1</sup>  $P(s)$ , then an **approximation of the ensemble average** would be given by:

$$\langle A(s) \rangle \simeq \frac{1}{N} \sum_{r=1}^N A(s^{(r)}), \quad (1.3)$$


where  $N$  is the total number of random states that were sampled, and  $s^{(r)}$  denotes the  $r$ -th sampled states.  $A(s^{(r)})$  is simply the value of the function  $A(s)$  when the system is in state  $s^{(r)}$ . The number of terms needed to compute  $\langle A \rangle$  can now be way smaller. However, this is not sufficient, as in order to sample states from  $P(s)$  one would still need to compute  $Z$  (which still a sum over a very large number of terms).

**Q3.** Let's try to clarify Eq. (1.3). Consider a dataset composed of  $N$  random states of the system sampled from  $P(s)$ . The right-hand side of Eq. (1.3) corresponds to the average value of  $A(s)$  over the dataset, let us denote it  $\langle A \rangle_{data}$ :

$$\langle A \rangle_{data} = \frac{1}{N} \sum_{r=1}^N A(s^{(r)}), \quad (1.4)$$

whereas the left-hand side of Eq. (1.3) is the thermal average of  $A(s)$  defined in Eq. (1.1). Let's try to clarify Eq. (1.3) with a few additional steps. How can one define the empirical probability  $P_{data}(s)$  to observe a given state  $s$  in the data? Can you re-write  $\langle A \rangle_{data}$  using  $P_{data}(s)$ ? What happens to  $\langle A \rangle_{data}$  when the number of datapoints become very large?

### 1.1.1 Defining a Stochastic Markovian process

 The trick proposed by Metropolis, Rosenbluth and Teller in 1953 consists of introducing a stochastic Markovian process between successive configurations,  $s^{(r)} \rightarrow s^{(r+1)}$ , that converges towards the desired equilibrium distribution. One can then obtain many samples of the desired distribution by recording states from the random process at equilibrium. The more steps are included, the more closely the distribution of the sample matches the actual desired distribution.

<sup>1</sup>i.e., randomly generate more often states that are more likely to occur (with a larger  $P(s)$ ), and less often states that are less likely to occur (with a smaller  $P(s)$ ).

**Introducing a Markovian process.** To do so, we must first introduce a numerical time  $t$ , which starts at  $t = 0$  at the beginning of the simulation. We then denote by  $p(s, t)$  the probability that the system is in the configuration  $s$  at time  $t$  during the simulation. To introduce a stochastic process, we must define the probabilities with which the system jumps from state to state during a very small time interval  $[t, t + dt)$  (for instance, take  $dt = 1/n$ , where  $n$  is the number of spins in the system). Besides, defining a *Markovian process* means that the probability of jumping from a state  $s$  to a state  $s'$  during  $dt$  only depends on the state  $s$  where the system was at time  $t$ , and not on the previous states of the system (the process is memory-less). We thus define the probability that the system jumps from  $s$  to  $s'$  during  $dt$  as:  $w(s \rightarrow s') dt$ .

**Q4.** Can you show that the probability to find the system in state  $s$  at time  $t + dt$  is given by:

$$p(s, t + dt) = p(s, t) + \sum_{s' \neq s} p(s', t) w(s' \rightarrow s) dt - \sum_{s' \neq s} p(s, t) w(s \rightarrow s') dt, \quad (1.5)$$

where the sums are over all states  $s'$  of the system that are different from  $s$ ? Deduce the equation for the time evolution of  $p(s, t)$ .

### 1.1.2 Convergence to equilibrium and Detailed Balance equation

 **Driving the system towards the desired equilibrium.** The stationary solution of the previous equation is:

$$\sum_{s'} p_{st}(s') w(s' \rightarrow s) = \sum_{s'} p_{st}(s) w(s \rightarrow s'), \quad (1.6)$$

where  $p_{st}(s)$  are the stationary probabilities of the states of the system for the simulated process. As we would like this process to converge towards the desired probability distribution  $p_{st}(s) = P(s)$  given by Eq. (1.2), this latter equation therefore gives us a set of constraints that must be satisfied by  $w(s \rightarrow s')$ .

A simple solution is given by:


$$P(s') w(s' \rightarrow s) = P(s) w(s \rightarrow s'), \quad (1.7)$$

which is known as the condition of **detailed balance**. It expresses that at stationarity, there is the same probability to observe the system transition from a state  $s$  to a state  $s'$  then to observe the reverse transition. Note that this is not necessarily the unique solution, but in general, it is difficult to prove that a system that doesn't satisfy detailed balance will converge to equilibrium, which is why many Monte Carlo algorithms use detailed balance.

**Q5.** Can you show that Eq. (1.7) implies that  $w(s \rightarrow s')$  does not depend on the partition function  $Z$ , but only on the Boltzmann factor  $\exp(-\beta E(s))$ ? We recall that the desired equilibrium distribution  $P(s)$  is given by Eq. (1.2).

This is great, because it means that if we find a solution  $w(s \rightarrow s')$  that satisfies Eq. (1.7), we will be able to design a Markovian stochastic process that converges towards the desired equilibrium  $P(s)$  without the need to sample the partition function! Once the process is at equilibrium, it will stay there (by definition), and we will be able to get many samples of the system at equilibrium and therefore compute the thermal average of any quantity  $A(s)$  using Eq. (1.3).

## 1.2 The Metropolis–Hastings algorithm

 A Monte Carlo algorithm works as follows. We start from a randomly sampled initial state (out-of-equilibrium) at time  $t = 0$ . We then consider very short time increments  $dt$  (typically, for an Ising model it is common to take  $dt = 1/n$ , where  $n$  is the total number sites), and during each time step  $dt$  we perform the two operations:

- from a current state  $s$  of the system, sample a random new state  $s'$ ;
- the new state is accepted with probability  $\Pi(s \rightarrow s')$ .

The choice for  $\Pi(s \rightarrow s')$  introduced by Metropolis et al. is:

$$\Pi(s \rightarrow s') = \begin{cases} 1, & \text{if } E(s') \leq E(s), \\ \exp[-\beta(E(s') - E(s))], & \text{otherwise.} \end{cases} \quad (1.8)$$

Note that each step of the numerical simulation corresponds to a very short time increment  $dt$ , which means that  $\Pi(s \rightarrow s')$  is the probability that the systems goes from  $s$  to  $s'$  during  $dt$ , and therefore  $\Pi(s \rightarrow s') = w(s \rightarrow s') dt$ .

**Q6.** Can you show that this choice of  $\Pi(s \rightarrow s')$  satisfies the detailed balance equation (1.7)?

**Q7.** Can you give a general interpretation of the stochastic process defined by Eq. (1.8)? We recall that:

$$\exp[-\beta(E(s') - E(s))] = \frac{P(s')}{P(s)}. \quad (1.9)$$

### 1.3 Application to the Ising model

For a trial state  $s'$  to be accepted, this state needs to be “close” to the previous state  $s$  in energy, because the acceptance probability is proportional to the exponential of the energy difference between the two states. If this difference is large and positive, the probability of acceptance becomes very small and the system can stay “trapped” for a long time in a local minimum. For the Metropolis algorithm, the single spin flip is generally used which leads to a dynamics where energy changes are reasonable. Because the dynamics must remain stochastic, a spin must be chosen randomly for each trial configuration, and not according to a regular sequence.

**Q8.** What is the energy difference,  $\Delta E = E(s') - E(s)$ , between a state  $s$  and a new state  $s'$  in which the spin  $s_k$  at site  $k$  has been flipped? Show that one can re-write the ratio of the state probabilities as:

$$\frac{P(s')}{P(s)} = \exp\left(-\beta h_k^{loc} s_k\right), \quad (1.10)$$

where  $s_k$  is the value of the spin at site  $k$  in the current state  $s$ , and where  $h_k^{loc}$  is a local field to be specified.

**Q9.** If a spin is flipped, what is the energy of a new configuration  $E(s')$  as a function of the old one? What is the value of the magnetization of the whole system  $M(s')$  in the new state as a function of the old one?

Note that in the case where the old configuration is kept, instantaneous quantities are unchanged, but time must be incremented.

**Q10.** Using these recursive relations provides a computationally efficient way to keep track of the energy  $E(s)$  and total magnetization  $M(s)$  of the system during the simulation. For the Ising model, all the thermodynamic quantities of interest can be expressed as statistical moments of  $E(s)$  and  $M(s)$  (i.e. as the average of powers of  $E$  and  $M$ ), which means that the computation of thermodynamic quantities can be easily performed (not computationally demanding). For instance, can you recall the expression of the specific heat and that of the susceptibility as functions of  $E(s)$  and  $M(s)$ ?

**Q11.** The simulation starts from a random initial state (out-of-equilibrium), we then impose a fixed temperature  $T$  (i.e. a fixed value of  $\beta$ ), and let the system evolves towards equilibrium using a Metropolis algorithm. We would like to record the average value of the magnetization  $\langle M \rangle = \sum_s M(s)$  in the stationary state. Can you describe the different steps of the program?

**Q12. 2D Ising model.** In 1944, Onsager obtained an exact solution for the 2D Ising model with external magnetic field. He showed that there exists a ferromagnetic transition at a critical temperature:

$$T_c = \frac{2J}{\log(1 + \sqrt{2})} \simeq 2.2691 J. \quad (1.11)$$

You can find [here](#) an online implementation of the Metropolis algorithm for a 2D Ising model. For the Ising model, it is known that the phase transition happens at  $T_c \simeq 2.27\dots$ . Play with the parameters of the algorithm. How does the systems look like when you take  $T > T_c$ ? How does it looks like when you take  $T < T_c$ ? Can you observe a phase transition? Can you see appearing magnetic domains for  $T < T_c$ ? How does it looks like when you take  $T = 0$ ? Can you explain why?

**Bonus question:** If you have some spare time, feel free to implement your own numerical simulation of the Ising model!

## 1.4 Solutions

**Q1:** We consider a system of Ising spins. We denote by  $s_i$  is the value  $\pm 1$  of the spin located at the site  $i$ , and by  $\mathbf{s}$  the state of all the spins of the system. The local magnetization  $m_i$  is the ensemble average of  $A(\mathbf{s}) = s_i$ , which can be written as:

$$m_i = \langle s_i \rangle = \sum_{\mathbf{s}} s_i P(\mathbf{s}). \quad (1.12)$$

For the Ising model, the *state probability* (i.e. the probability that the system is in a given state  $\mathbf{s}$ ) is defined by:

$$P(\mathbf{s}) = \frac{\exp(-\beta E(\mathbf{s}))}{Z}, \quad \text{where } E(\mathbf{s}) = -h \sum_i s_i - J \sum_{\langle i,j \rangle} s_i s_j. \quad (1.13)$$

In this equation,  $h$  is a parameter that corresponds to an external magnetic field applied to all the spins of the system, the sum over  $\langle i, j \rangle$  denotes a sum over neighboring pairs of spins  $s_i$  and  $s_j$ , and  $J$  is a coupling parameter that corresponds to the strength of the coupling between neighboring spins  $s_i$  and  $s_j$ . The partition function  $Z$  is then obtained from the normalisation of  $P(\mathbf{s})$  (i.e.  $\sum_{\mathbf{s}} P(\mathbf{s}) = 1$ ), which gives that:

$$Z = \sum_{\mathbf{s}} \exp(-\beta E(\mathbf{s})). \quad (1.14)$$

**Q2:** A system with 20 spins can be in  $2^{20}$  different states, which means that there are  $2^{20} \simeq 10^6$  terms in the sum in Eq. (1.1). Idem for the sum to compute  $Z$ . These are a lot of terms to sum over to compute  $Z$ , or to compute any ensemble average (and 20 spins is only a small system!). This also means that the distribution  $P(\mathbf{s})$  is a distribution over  $\sim 10^6$  elements, so sampling from that distribution is not very practical, and many of the states will actually have a very small probability. Ideally, we would like to avoid having to compute these huge sums to compute observable of the systems. Ideally, it would also be good to be able to take into account mostly the contributions of the most likely states to the sum.

**Q3:** The empirical probability  $P_{data}(\mathbf{s})$  to observe the state  $\mathbf{s}$  in the data is given by the ratio of the number of times  $K(\mathbf{s})$  that the state  $\mathbf{s}$  appears in the data to the total number of datapoints:

$$P_{data}(\mathbf{s}) = \frac{K(\mathbf{s})}{N}. \quad (1.15)$$

We can re-write  $\langle A \rangle_{data}$  as:

$$\langle A \rangle_{data} = \frac{1}{N} \sum_{r=1}^N A(\mathbf{s}^{(r)}) = \frac{1}{N} \sum_{\mathbf{s}} K(\mathbf{s}) A(\mathbf{s}) = \sum_{\mathbf{s}} \frac{K(\mathbf{s})}{N} A(\mathbf{s}) \quad (1.16)$$

$$= \sum_{\mathbf{s}} P_{data}(\mathbf{s}) A(\mathbf{s}), \quad (1.17)$$

where we replaced the sum over all the datapoints by a sum over all the states weighted by the number of times each state occurs. Note that most of the times, many states are never observed in the dataset, and for these states  $K(\mathbf{s}) = 0$ . As the number of datapoints  $N$  increases, the empirical distribution  $P_{data}(\mathbf{s})$  converges to the true distribution  $P(\mathbf{s})$ , and we have that:

$$\langle A \rangle_{data} \xrightarrow{N \rightarrow \infty} \langle A \rangle, \quad (1.18)$$

in which we recall that  $\langle A \rangle = \sum_{\mathbf{s}} P(\mathbf{s}) A(\mathbf{s})$  and  $\langle A \rangle_{data} = \sum_{\mathbf{s}} P_{data}(\mathbf{s}) A(\mathbf{s})$ .

**Q4:** The probability to find the system in state  $\mathbf{s}$  at time  $t + dt$  is given by:

$$p(\mathbf{s}, t + dt) = \underbrace{\sum_{\mathbf{s}' \neq \mathbf{s}} p(\mathbf{s}', t) w(\mathbf{s}' \rightarrow \mathbf{s}) dt}_{(1)} + \underbrace{p(\mathbf{s}, t) \left( 1 - \sum_{\mathbf{s}' \neq \mathbf{s}} w(\mathbf{s} \rightarrow \mathbf{s}') dt \right)}_{(2)}, \quad (1.19)$$

where

- (1) is the probability that the system was in state  $s'$  at time  $t$  and transition from  $s'$  to  $s$  during  $dt$ .
- (2) is the probability that the system was in state  $s$  at time  $t$  and stayed in  $s$  during  $dt$  (i.e. did not transition to another state during  $dt$ ).

Re-organizing the terms gives Eq. (1.19).

**Q5:** Replacing the expression of  $P(s)$  in Eq. (1.7) gives:

$$\frac{w(s \rightarrow s')}{w(s' \rightarrow s)} = \frac{P(s')}{P(s)} = e^{-\beta(E(s') - E(s))}. \quad (1.20)$$

which doesn't depends on  $Z$  anymore. This implies that  $w(s \rightarrow s')$  does not depend on the partition function  $Z$ .

**Q6.** We can decompose the problem in two cases:

- if  $E(s') \leq E(s)$ , then:

$$w(s \rightarrow s') = 1 \quad \text{and} \quad w(s' \rightarrow s) = e^{-\beta(E(s) - E(s'))} = \frac{P(s)}{P(s')}, \quad (1.21)$$

- or  $E(s') > E(s)$ , then:

$$w(s \rightarrow s') = e^{-\beta(E(s') - E(s))} = \frac{P(s')}{P(s)} \quad \text{and} \quad w(s' \rightarrow s) = 1. \quad (1.22)$$

In both case, we recover the detailed balance Eq. (1.7).

**Q7.** According to Eq. (1.8), if the system is more likely to be in the new state  $s'$  than in the current state  $s$  at equilibrium, then the algorithm will move the system to the new state with probability 1. If instead the new state is less likely than the current state at equilibrium, then the algorithm will still allow a transition of to the new state, but with a probability that is proportional to the ratio of the probabilities of the two states at equilibrium. Thus, if  $s'$  is very unlikely compared to  $s$ , then the ratio will be close to zero and a transition to  $s'$  will be unlikely, and if  $s'$  is almost as probable as  $s$ , then the ratio will be close to 1 and a transition to  $s'$  will still have a high chance to occur.

Another way to see this, is that: The stochastic process is driving the system towards lower energy states, while still allowing small moves towards higher energy states. This in fact prevents the system from getting stuck in local minima.

**Q8.** One can write the two states  $s = (s_1, \dots, s_k, \dots, s_n)$  and  $s' = (s_1, \dots, -s_k, \dots, s_n)$ , which differ only by a flip of  $s_k$ . Let us separate the contributions of the spin  $s_k$  from the rest in  $E(s)$ :

$$E(s) = - \left( h \sum_{i \neq k} s_i + J \sum_{\langle i, j \rangle, i, j \neq k} s_i s_j \right) - \left( h s_k + J \sum_{i: \langle i, k \rangle} s_i s_k \right). \quad (1.23)$$

Idem for  $E(s')$ :

$$E(s) = - \left( h \sum_{i \neq k} s_i + J \sum_{\langle i, j \rangle, i, j \neq k} s_i s_j \right) - \left( h (-s_k) + J \sum_{i: \langle i, k \rangle} s_i (-s_k) \right), \quad (1.24)$$

$$= - \left( h \sum_{i \neq k} s_i + J \sum_{\langle i, j \rangle, i, j \neq k} s_i s_j \right) + \left( h s_k + J \sum_{i: \langle i, k \rangle} s_i s_k \right). \quad (1.25)$$

The energy difference  $\Delta E = E(s') - E(s)$  is then:

$$\Delta E = +2 \left( h s_k + J \sum_{i: \langle i, k \rangle} s_i s_k \right), \quad (1.26)$$

$$= h_k^{loc} s_k, \quad \text{where } h_k^{loc} = 2h + 2J \sum_{i: \langle i, k \rangle} s_i. \quad (1.27)$$

The quantity  $h_i^{loc}$  is a local field created by the 4 neighbors of the spin  $s_k$  (the notation “ $i :< i, k >$ ” indicates a summation over the sites  $i$ , such that  $i$  is a neighbor of  $k$ ). As  $P(s')/P(s) = \exp(-\beta(E(s') - E(s))) = \exp(-\beta\Delta E)$ , one directly recovers Eq. (1.10).

**Q9.** Since  $\Delta E = E(s') - E(s)$ , we have that the new energy is expressed as a function of the old energy by:

$$E(s') = E(s) + \Delta E = E(s) + h_k^{loc} s_k \quad (1.28)$$

where  $s_k$  is spin that is flipped from the old to the new configuration. To compute the new magnetization  $M(s')$ , we recall that the total average magnetization is given by:

$$M(s) = \sum_{i=1}^N s_i \quad (1.29)$$

Flipping the spin  $s_k$  changes the magnetization as follows

$$M(s') = -s_k + \sum_{i \neq k} s_i = -2s_k + \sum_i s_i = M(s) - 2s_k, \quad (1.30)$$

where  $M(s)$  is the total average magnetization of the original state.

**Q10.** **This question is in fact discussed in the next tutorial (TCS T2 – exercise 1, Q5).** The specific heat is given by the derivative of the average energy  $\langle E \rangle$  with respect to the temperature  $T$  at fixed external field  $h$ :

$$c = \frac{1}{N} \left( \frac{\partial \langle E \rangle}{\partial T} \right)_h. \quad (1.31)$$

The fluctuation-dissipation theorem connects  $c$  with the variance of the total energy  $E$  of the system (see proof in tutorial 2, exercise 1, Q5):

$$\frac{\langle E^2 \rangle - \langle E \rangle^2}{N} = k_B T^2 c. \quad (1.32)$$

Similarly, the susceptibility is given by the derivative of the average total magnetization with respect to the strength of the external magnetic field at fixed value of the temperature  $T$ :

$$\chi = \frac{1}{N} \left( \frac{\partial \langle M \rangle}{\partial h} \right)_T. \quad (1.33)$$

The fluctuation-dissipation theorem connects  $\chi$  with the variance of  $M$  (see proof in tutorial 2, exercise 1, Q5):

$$\frac{\langle M^2 \rangle - \langle M \rangle^2}{N} = k_B T \chi. \quad (1.34)$$

**Q11.** We consider a system with  $n$  spins. In the algorithm, the system initially starts from a random initial state  $s$ , out-of-equilibrium. During each small time step  $dt = 1/n$ , we perform the following operations:

- choose a random spin  $s_k$  (we will consider the candidate state  $s'$  corresponding to flipping the spin  $s_k$  in the state  $s$ );
- using Eq. (1.27), compute the difference in energy  $\Delta E$  between the two states  $s$  and  $s'$ ;
- if  $\Delta E \leq 0$  the new state  $s'$  is accepted: we flip  $s_k$  and compute the new magnetisation  $M(s')$  using Eq. (1.30);
- else (i.e.,  $\Delta E > 0$ ) the new state  $s'$  is accepted with probability  $\Pi(s \rightarrow s') = \exp(-\beta\Delta E)$ .

It is typical to consider a unit time to be of the order of  $n dt = 1$ , i.e. after we have performed  $n$  times the Monte Carlo steps described above. Before we start recording the average magnetization  $\langle M \rangle$ , we must first wait that the system has reached equilibrium. For instance, we may wait of the order of 5000 unit time before we start recording the magnetization to compute the averages (see additional comment below). We can then start recording the successive values of  $M(s)$  to compute  $\langle M \rangle$  using Eq. (1.3). To avoid too much correlation between the successive values of  $M$  that are recorded, one can also record values every 10 unit time.



**Important comments:** Computation of a thermal average only starts when the system reaches equilibrium, namely when  $p(s, t) \simeq p_{st}(s)$ . Therefore, in a Monte Carlo run, there are generally two parts: The first starting from an initial configuration, where a run is performed in order to lead the system close to equilibrium; the second where the system evolves in the vicinity of equilibrium with the computation of thermodynamic quantities. To estimate the relaxation time towards equilibrium, a naive approach consists of following the evolution of the instantaneous energy of the system and in considering that equilibrium is reached when the energy is stabilized around a quasi-stationary value.