# 3

# Monte Carlo Sampling and Markov Chains

## 3.1 Introduction

Monte Carlo methods indicate a broad class of numerical algorithms that are based upon repeated random *sampling* to obtain the solution of several mathematical and physical problems. The typical issue is about the calculation of large sums or integrals and the revolutionary idea is that we do not perform an *exact* enumeration/integration but instead we generate random *samples*, which are then added together to approximate the exact result. Therefore, the concept of sampling is pivotal in any Monte Carlo approach: its meaning is just to produce random examples of configurations (e.g., $N$ classical particles distributed in a box) that are used to give an accurate estimate of the exact quantity under examination (e.g., the internal energy at fixed temperature).

The roots of the Monte Carlo approaches date back to Enrico Fermi's attempts while studying neutron diffusion in the early thirties. He did not publish anything on the subject, but he got the credits for the idea that a time-independent Schrödinger equation can be interpreted in terms of a system of particles performing a random walk (Metropolis and Ulam, 1949). Few years after, a fundamental step forward has been done by Stanislaw Ulam, when he was working on nuclear-weapon projects at the Los Alamos National Laboratory. His first thoughts about Monte Carlo methods were suggested by a question that occurred when he was convalescing from an illness and playing solitaire. Ulam tried to calculate the likelihood of winning based on the initial layout of the cards. After exhaustive combinatorial calculations, he decided to go for a more practical approach of trying out many different layouts and observing the number of successful games. At that time, a new era of fast computers was beginning and John von Neumann understood the relevance of Ulam's suggestion and proposed a statistical approach to solve the problem of neutron diffusion in a fissionable material. Ulam and von Neumann worked together to develop algorithms including importance sampling and rejection sampling, which are at the basis of any modern Monte Carlo technique. In addition, von Neumann developed

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

a way to obtain pseudo-random numbers by using the so-called middle-square digit method, since he realized that using a truly random number was extremely slow. Ulam and von Neumann required a code name for their secret project carried on with these stochastic methods. Then, Nicholas Metropolis, who was also working at the Los Alamos National Laboratory, suggested the name Monte Carlo, which refers to the Monte Carlo Casino in Monaco where Ulam's uncle used to gamble with cards (Metropolis and Ulam, 1949). A nice recollection of the early days of the Monte Carlo methods has been reported by Metropolis (1987).

Although the first tests of the Monte Carlo methods were done on a variety of problems in neutron transport, the real breakthrough came in 1952, when a new computer, called MANIAC, became operational at the Los Alamos laboratories (Anderson, 1986): Metropolis, together with Arianna and Marshall Rosenbluth, and Augusta and Edward Teller, studied the equation of state of the two-dimensional motion of hard spheres (Metropolis et al., 1957). They developed a strategy to enhance the computational efficiency in describing systems at thermal equilibrium, i.e., obeying the Boltzmann distribution function. According to this strategy, if a statistical move of a particle resulted in a decrease in the total energy, the new configuration was accepted. By contrast, if there was an increase in the total energy, the new configuration was accepted only if it survived a game of chance biased by a Boltzmann factor. Otherwise, the new configuration was taken equal to the old one. The so-called Metropolis algorithm, based upon random walks, is one of the most pervasive numerical algorithms used in computational approaches and was included in the top-ten list of "the greatest influence on the development and practice of science and engineering in the 20th century" (Dongarra and Sullivan, 2000). The fast improvement of computers made possible, in recent years, to reach incredible achievements in many branches of science, including mathematics, physics, chemistry, but Monte Carlo methods are frequently used also in economical sciences to predict the behavior of stock markets.

In this section, we give the justification of the Monte Carlo sampling to compute integrals, which is the simplest application of the Monte Carlo approach. Indeed, suppose that we must compute the following $d$-dimensional integral:

$$\mathcal{I} = \int \mathbf{dx}\, F(\mathbf{x}), \tag{3.1}$$

where $F(\mathbf{x})$ is a generic function of the vector $\mathbf{x}$ with $d$ components. Then, without loss of generality, we can always split $F(\mathbf{x})$ into a probability density $\mathcal{P}(\mathbf{x})$ (with $\mathcal{P}(\mathbf{x}) \geq 0$ and $\int \mathbf{dx}\, \mathcal{P}(\mathbf{x}) = 1$) and a function $f(\mathbf{x}) = F(\mathbf{x})/\mathcal{P}(\mathbf{x})$:

$$\mathcal{I} = \langle f(\mathbf{x}) \rangle = \int \mathbf{dx}\, f(\mathbf{x})\, \mathcal{P}(\mathbf{x}), \tag{3.2}$$

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

which is nothing else than the expectation value of the random variable $f(\mathbf{x})$ over the distribution function $\mathcal{P}(\mathbf{x})$, i.e. the multi-dimensional generalization of Eq. (2.29). Then, the central limit theorem implies that the *deterministic* integral $\mathcal{I}$ is equal to the *stochastic* random variable computed as the average value of $f(\mathbf{x})$ over a large number of samplings:

$$\langle f(\mathbf{x})\rangle = \int \mathbf{dx}\, f(\mathbf{x})\ \mathcal{P}(\mathbf{x}) \approx \frac{1}{N}\sum_i f(\mathbf{x}_i), \tag{3.3}$$

where the values of $\mathbf{x}_i$ in the sum of the r.h.s. are distributed according to the probability density $\mathcal{P}(\mathbf{x})$. Indeed, for large $N$, the variable:

$$\bar{f} = \frac{1}{N}\sum_i f(\mathbf{x}_i) \tag{3.4}$$

is normally (Gaussian) distributed, with mean equal to $\langle f(\mathbf{x})\rangle$ and variance $\sigma^2/N$, where $\sigma^2 = \langle f^2(\mathbf{x})\rangle - \langle f(\mathbf{x})\rangle^2$. Therefore, for $N \to \infty$, the random variable $\bar{f}$ tends to the deterministic number $\langle f(\mathbf{x})\rangle$ (since fluctuations decrease to zero with $1/\sqrt{N}$). This is the meaning for Eq. (3.3), in which the l.h.s. is a deterministic number and the r.h.s. is a random variable.

In summary, the validity of the stochastic calculation (the numerical simulation) is based upon the fact that, whenever the number of samplings $N$ is large enough, then the error due to statistical fluctuations goes to zero, implying that the errorbars of the simulations can be kept under control. In this sense, we have that:

$$\int \mathbf{dx}\, f(\mathbf{x})\ \mathcal{P}(\mathbf{x}) \approx \langle\langle f(\mathbf{x}_i)\rangle\rangle, \tag{3.5}$$

where $\langle\langle \dots \rangle\rangle$ denotes the statistical average over many independent samples (distributed according to $\mathcal{P}(\mathbf{x})$). We emphasize that any stochastic average without errorbars is completely meaningless, since one would not have any idea of the accuracy of the simulation. While the estimation of the integral is given by $\bar{f}$, the errorbar can be obtained from the estimator of $\sigma^2$, see Eq. (2.61):

$$s^2 = \frac{1}{N}\sum_i \left[f(\mathbf{x}_i) - \bar{f}\right]^2. \tag{3.6}$$

The main issues of the stochastic calculation are (i) to generate configurations $\mathbf{x}_i$ that are distributed according to the desired probability density $\mathcal{P}(\mathbf{x})$ and then (ii) compute the function $f(\mathbf{x}_i)$ for all these configurations.

Whenever it is possible to generate configurations with the probability density $\mathcal{P}(\mathbf{x})$, we talk about *direct sampling*; in this case, all configurations are independent from each other. Unfortunately, this is only possible in a few cases for very simple probability densities that depend upon few variables $\mathbf{x}$ (e.g., the case with

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

$d = 1$). In the general case, we are not able to directly sample the probability density and indirect ways of obtaining such configurations must be devised. This is the case of the so-called *Markov chains*, in honor of the Russian mathematician Andrei Andreievich Markov. In the following, we will present in some detail both the direct sampling and the theory of Markov chains. In particular, while the latter approach is very general and can be applied in a large variety of cases, the former one can be implemented to sample discrete probabilities, where the total number of possible outcomes can be enumerated and stored in the computer (while it is rarely used to sample continuous variables in $d > 1$).

## 3.2 Reweighting Technique and Correlated Sampling

Before discussing the direct sampling, we briefly illustrate the concept of *reweighting*, which allows us to obtain the average of a function $f(\mathbf{x})$ over the probability $\mathcal{Q}(\mathbf{x})$, once a sampling over $\mathcal{P}(\mathbf{x})$ has been performed. In fact, let us suppose that we have two probabilities $\mathcal{P}(\mathbf{x})$ and $\mathcal{Q}(\mathbf{x})$ that are defined in terms of their corresponding weights $\mathcal{W}_p(\mathbf{x})$ and $\mathcal{W}_q(\mathbf{x})$, respectively:

$$\mathcal{P}(\mathbf{x}) = \frac{\mathcal{W}_p(\mathbf{x})}{\int \mathbf{dx}\, \mathcal{W}_p(\mathbf{x})}, \tag{3.7}$$

$$\mathcal{Q}(\mathbf{x}) = \frac{\mathcal{W}_q(\mathbf{x})}{\int \mathbf{dx}\, \mathcal{W}_q(\mathbf{x})}. \tag{3.8}$$

Then, from general grounds, we have that:

$$\frac{\int \mathbf{dx}\, f(\mathbf{x})\mathcal{W}_q(\mathbf{x})}{\int \mathbf{dx}\, \mathcal{W}_q(\mathbf{x})} = \frac{\int \mathbf{dx}\, f(\mathbf{x})\mathcal{R}(\mathbf{x})\mathcal{W}_p(\mathbf{x})}{\int \mathbf{dx}\, \mathcal{R}(\mathbf{x})\mathcal{W}_p(\mathbf{x})}, \tag{3.9}$$

where $\mathcal{R}(\mathbf{x}) = \mathcal{W}_q(\mathbf{x})/\mathcal{W}_p(\mathbf{x})$ is the ratio between the two weights. Therefore, the statistical sampling over the new probability $\mathcal{Q}(\mathbf{x})$ can be expressed in terms of samplings over $\mathcal{P}(\mathbf{x})$:

$$\langle\langle f(\mathbf{x})\rangle\rangle_{\mathcal{Q}} = \frac{\langle\langle f(\mathbf{x})\mathcal{R}(\mathbf{x})\rangle\rangle_{\mathcal{P}}}{\langle\langle \mathcal{R}(\mathbf{x})\rangle\rangle_{\mathcal{P}}}, \tag{3.10}$$

where $\langle\langle \dots \rangle\rangle_{\mathcal{Q}}$ and $\langle\langle \dots \rangle\rangle_{\mathcal{P}}$ denote the statistical samplings over $\mathcal{Q}(\mathbf{x})$ and $\mathcal{P}(\mathbf{x})$, respectively.

Then, the same sampling (i.e., set of configurations $\{\mathbf{x}_i\}$) obtained from $\mathcal{P}(\mathbf{x})$ can be used to evaluate averages over $\mathcal{Q}(\mathbf{x})$. This way of proceeding gives rise to the concept of *correlated sampling*, since two quantities are evaluated with the same set of configurations. Of course, in order to have an accurate statistics (i.e., small errorbars) on the reweighted quantity $\langle\langle f(\mathbf{x})\rangle\rangle_{\mathcal{Q}}$, the two weights must

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

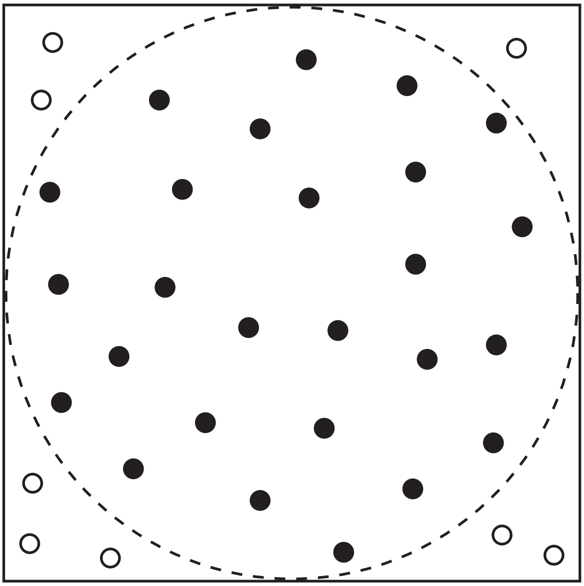

Figure 3.1 Direct sampling Monte Carlo to compute $\pi$. We shoot random bullets in the square, for a large number $N$ of trials (i.e., $N \to \infty$) the ratio between the number of bullets inside the circle and the total number of trials converges to $\pi/4$; see text.

be quite similar, otherwise the configurations $\{\mathbf{x}_i\}$ would fall in regions where $\mathcal{Q}(\mathbf{x})$ is small and, therefore, irrelevant for the final result.

## 3.3 Direct Sampling

Here, in order to discuss the *direct sampling* method, we start form a simple example. Suppose that we want to compute $\pi$ by a stochastic approach, then we can draw a circle with radius $r$ and a square that exactly contains it, namely with side $L = 2r$, see Fig. 3.1. Then, we can randomly shoot bullets inside the square, counting every trial; each time a bullet falls inside the circle we increase by one the number of "hits." By keeping track of trials and hits, we can perform a direct sampling Monte Carlo calculation: the ratio between hits and trial is approaching, for a large number of trials $N$, the ratio of the areas of the circle and the square:

$$\frac{\pi}{4} = \frac{\int_{\text{circle}} dx\, dy}{\int_{\text{square}} dx\, dy} = \int_{\text{square}} dx\, dy\, f(x, y)\, \mathcal{P}(x, y), \tag{3.11}$$

with

$$\mathcal{P}(x, y) = \frac{1}{\int_{\text{square}} dx\, dy}, \tag{3.12}$$

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

and

$$f(x,y) = \begin{cases} 1 & \text{if } (x,y) \text{ is inside the circle,} \\ 0 & \text{otherwise.} \end{cases} \tag{3.13}$$

The function $\mathcal{P}(x,y)$ is non negative and normalized to unity, thus it may well represent a probability function. Shooting bullets in the square implies to generate a couple of random numbers (one for $x$ and the other for $y$) that are uniformly distributed between 0 and $L$:

$$\frac{\pi}{4} \approx \frac{1}{N} \sum_i f(x_i, y_i). \tag{3.14}$$

This is the first (and simplest) example of how we can use a stochastic (i.e., Monte Carlo) approach to compute integrals. In general, any (multidimensional) integral such as Eq. (3.1) can be, in principle, computed by taking random numbers that are uniformly distributed in the whole interval of integration and then computing the value of the function in these points, i.e., $f(\mathbf{x}_i)$. However, if $f(\mathbf{x})$ is a sharply peaked function, this uniform way of sampling the interval is not efficient, since we would compute many points $\mathbf{x}_i$ for which the function $f(\mathbf{x})$ gives a negligible contribution to the integral $\mathcal{I}$, while only few points would give a finite contribution. In this case, some tricks must be employed in order to improve the sampling procedure, which we will discuss in the following.

## 3.4 Importance Sampling

Whenever the function $f(\mathbf{x})$ that must be evaluated has sharp peaks, a uniform sampling is not efficient, because it would lead to a considerable waste of time, spending efforts to visit regions that give a negligible contribution to the final result. A simple way to overcome this problem is to consider the so-called *importance sampling*. The idea is very general and is used in several Monte Carlo calculations. Nevertheless, it can be explained in the simple case of a one-dimensional integral. Indeed, let us suppose that we have to evaluate:

$$\mathcal{I} = \int_a^b dx\, F(x), \tag{3.15}$$

where $F(x)$ is a function that is peaked in a given point between $a$ and $b$. Whenever we know the (approximated) location of the relevant regions where the function $F(x)$ is sizable, we can define probability density $\mathcal{P}(x)$ in $[a,b]$ (i.e., $\mathcal{P}(x) > 0$ and $\int_a^b dx\, \mathcal{P}(x) = 1$), which is also sizable in these regions and small everywhere else. Then, we can rewrite the original integral as:

$$\mathcal{I} = \int_a^b dx\, \frac{F(x)}{\mathcal{P}(x)}\, \mathcal{P}(x). \tag{3.16}$$

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

Now, if we are able to generate random numbers that are distributed according to $\mathcal{P}(x)$, we can evaluate the integral $\mathcal{I}$ as:

$$\mathcal{I} \approx \frac{1}{N} \sum_i \frac{F(x_i)}{\mathcal{P}(x_i)}; \tag{3.17}$$

the corresponding errorbar can be estimated from:

$$s^2 = \frac{1}{N} \sum_i \left[ \frac{F(x_i)}{\mathcal{P}(x_i)} \right]^2 - \left[ \frac{1}{N} \sum_i \frac{F(x_i)}{\mathcal{P}(x_i)} \right]^2. \tag{3.18}$$

The crucial point is that if $\mathcal{P}(x)$ is chosen to be close enough to $F(x)$, then the statistical fluctuations are highly reduced. Indeed, $s^2$ is a non-negative number, its minimum $s^2 = 0$ being reached for $\mathcal{P}(x) \propto F(x)$. In this case, the Monte Carlo sampling has no fluctuations. It must be emphasized that this limiting case is trivial and the Monte Carlo is useless: if we are able to extract random numbers that are distributed with $\mathcal{P}(x) \propto F(x)$ then we would have performed the integral analytically. Instead, in the realistic situation, whenever $\mathcal{P}(x)$ resembles the function $F(x)$, the statistical fluctuations are dramatically decreased, thus reducing the number of samplings that is needed to reach a given accuracy.

In general, there are sophisticated algorithms that generate very long sequences of essentially uncorrelated and uniformly distributed random numbers (or, to be more precise, *pseudo-random numbers*, see appendix A). Therefore, if we want to produce numbers that are distributed according to a different probability function, some further work is needed. In the following, we discuss some simple cases in which this task can be accomplished and the difficulties that arise in general cases.

## 3.5 Sampling a Discrete Distribution Probability

Here, we discuss how it is possible to sample a discrete probability distribution $P(k)$, with $k = 1, \dots, M$ mutually exclusive events. One option is to use the so-called *acceptance-rejection* method (or more simply, just *rejection* method). This approach can be also used to sample continuous variables and is closely related to the one that we have illustrated for computing $\pi$ in section 3.3. In practice, we embed the histogram of the probability $P(k)$ into a large rectangular board and then we shoot bullets in it (assuming that these are uniformly distributed in the board), see Fig. 3.2. Let us denote by $P_{\max}$ the maximum value of the $P(k)$'s; then, we generate two uniformly distributed random numbers: a first one, $r_1$ in $[0, 1)$, to obtain the event $k_r = \text{int}(M \times r_1) + 1$ (where $\text{int}(x)$ indicates the integer part of $x$),

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

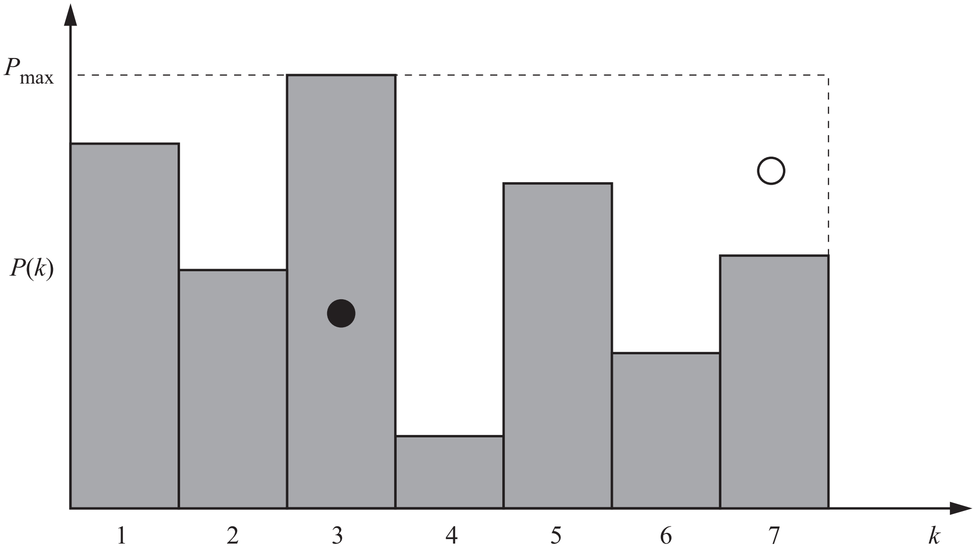

Figure 3.2 Rejection method to sample a discrete probability $P(k)$, with $k = 1, \ldots, M$ possible outcomes (here $M = 7$). Two uniformly distributed random numbers are generated: $r_1$ in $[0, 1)$ to obtain the event $k_r = \text{int}(M \times r_1) + 1$ and $r_2$ in $[0, P_{\max})$. If $r_2 \leq P(k_r)$ the trial is successful and $k_r$ is taken (the full dot selects $k_r = 3$), otherwise the trial is rejected (the empty dot indicates the rejected $k_r = 4$ event) and another couple of random numbers is generated.

and a second one, $r_2$ in $[0, P_{\max})$. If $r_2 \leq P(k_r)$ then the trial is successful and $k_r$ is taken as the output, otherwise the trial is rejected and another couple of random numbers is generated. It is clear that the probability to obtain a given event $k$ is proportional to $P(k)$, since the events are generated uniformly, but accepted only if $r_2 \leq P(k)$.

This approach is not very efficient, since a given number of trials are rejected and, therefore, do not contribute to generate any output. Nevertheless, the computational cost of a rejected trial is not huge since it just requires the generation of two random numbers. From Fig. 3.2, it is clear that the rejection probability is proportional to $\sum_k (P_{\max} - P_k) = MP_{\max} - 1$, which is the area of the section of the rectangular board above the histogram. In the trivial (and useless) case where all events are equiprobable with $P_{\max} = 1/M$ all trials are accepted. By contrast, when one event has a probability that is much larger than all the other ones, the number of rejected trials will be very large and the algorithm becomes very inefficient. In fact, a simpler approach is possible, without rejection. To visualize this approach, we must organize all probabilities in a single row, forming a sequence of boxes of length $P(k)$, see Fig. 3.3. Since $\sum_k P(k) = 1$, the total length of the row is equal to one. Then, just one uniformly distributed random number $r$ is generated

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

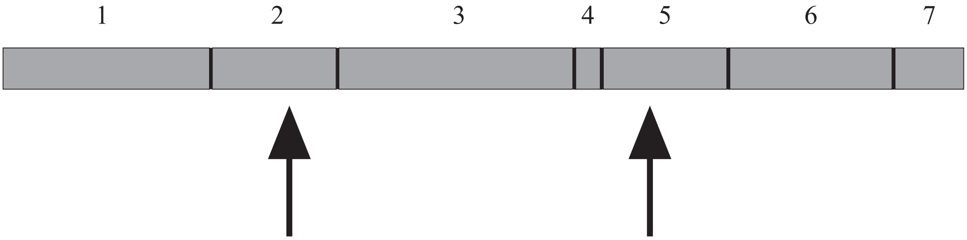

Figure 3.3 No-rejection method to sample a discrete probability $P(k)$, with $k = 1, \dots, M$ possible outcomes (here $M = 7$). Just one uniformly distributed random number $r$ is generated in $[0, 1)$, the event $k_r$ is chosen, such that $\sum_{k=1,k_r-1} P(k) \leq r < \sum_{k=1,k_r} P(k)$. The two arrows indicate two attempts to select the events: the events $k_r = 2$ and 5 have been obtained.

in $[0, 1)$, which identifies a given box in the row and determines the event $k_r$, see Fig. 3.3. More formally, the value of $k_r$ is the one that satisfies the following condition:

$$\sum_{k=1}^{k_r-1} P(k) \leq r < \sum_{k=1}^{k_r} P(k). \tag{3.19}$$

Clearly, also in this case, the probability to select a given $k$ is given by $P(k)$, thus providing the correct result.

## 3.6 Sampling a Continuous Density Probability

Let us discuss the case of continuous random variables with probability density $\mathcal{P}(x)$. For simplicity, we consider the one-dimensional case in which there is a single random variable $x$. Following the discussion of section 2.4, we apply Eq. (2.41) in the case where $x$ is uniformly distributed $[0, 1)$ (which is what random-number generators provide us) to get:

$$x = \int_{-\infty}^{y} ds \, \mathcal{P}_y(s). \tag{3.20}$$

This equation tells us that, if we want to have a random variable that is distributed according to a given probability density $\mathcal{P}_y(y)$, we must (i) perform the integral of $\mathcal{P}_y(y)$ to obtain its cumulative probability $F(y)$:

$$F(y) = \int_{-\infty}^{y} ds \, \mathcal{P}_y(s), \tag{3.21}$$

and then (ii) extract a random variable $x$ that is uniformly distributed in $[0, 1)$ and find the value $y$ such that $F(y) = x$; therefore, we must invert $F(y)$ and find

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

$y = F^{-1}(x)$. In this sense, Eq. (3.20) is nothing less than the continuous version of Eq. (3.19) that was obtained to sample a discrete distribution. For generic distributions, it is not obvious that steps (i) and (ii) can be done efficiently.

Let us consider a simple example that can be worked out analytically. Suppose that we want to generate random numbers according to the exponential distribution:

$$\mathcal{P}(y) = \begin{cases} Ae^{-Ay} & \text{for } y \geq 0, \\ 0 & \text{for } y < 0, \end{cases} \tag{3.22}$$

where $A$ is a given constant. Then, we have the following equation:

$$x = A \int_0^y ds\, e^{-As} = 1 - e^{-Ay}, \tag{3.23}$$

which can be easily solved for $y$, giving:

$$y = -\frac{1}{A} \ln(1 - x). \tag{3.24}$$

In this simple example, we were able to perform both the integration and the inversion. However, this could be not achievable in a simple way for generic probabilities. Here, we show an important case that, however, can be overcome with a simple trick. Suppose that we want to generate random numbers with a Gaussian distribution (for simplicity we take $\mu = 0$ and $\sigma = 1$):

$$\mathcal{P}(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}. \tag{3.25}$$

In this case, we would solve the following equation:

$$x = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} ds\, e^{-\frac{s^2}{2}}. \tag{3.26}$$

Unfortunately, the integral does not have a simple closed form and, therefore, this approach does not give any useful outcome. Nevertheless, there is a way to overcome this problem, which goes under the name of Box-Muller trick. Let us start from noting that:

$$\left\{ \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} dx\, e^{-\frac{x^2}{2}} \right\}^2 = \frac{1}{2\pi} \int_{-\infty}^{+\infty} dx\, e^{-\frac{x^2}{2}} \int_{-\infty}^{+\infty} dy\, e^{-\frac{y^2}{2}} = 1. \tag{3.27}$$

By introducing polar coordinates:

$$x = \rho \cos\theta, \tag{3.28}$$

$$y = \rho \sin\theta, \tag{3.29}$$

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

we have:

$$\frac{1}{2\pi}\int_{-\infty}^{+\infty} dx\, e^{-\frac{x^2}{2}} \int_{-\infty}^{+\infty} dy\, e^{-\frac{y^2}{2}} = \int_0^{2\pi} \frac{d\theta}{2\pi} \int_0^{\infty} d\rho\, \rho\, e^{-\frac{\rho^2}{2}}. \tag{3.30}$$

Then, by considering the new variable $\xi = \rho^2/2$, we finally have:

$$\frac{1}{2\pi}\int_{-\infty}^{+\infty} dx\, e^{-\frac{x^2}{2}} \int_{-\infty}^{+\infty} dy\, e^{-\frac{y^2}{2}} = \int_0^{2\pi} \frac{d\theta}{2\pi} \int_0^{\infty} d\xi\, e^{-\xi}. \tag{3.31}$$

The meaning of this equation is that two independent variables $x$ and $y$ with a Gaussian distribution are equivalent to other two independent variables $\theta$ and $\xi$, the former one being uniformly distributed in $[0, 2\pi)$ and the latter one being exponentially distributed, see Eq. (3.24) with $A = 1$. Therefore, two Gaussian variables can be easily obtained from extracting $\theta$ and $\xi$:

$$\theta = 2\pi r_1, \tag{3.32}$$

$$\xi = -\ln(1 - r_2), \tag{3.33}$$

where $r_1$ and $r_2$ are two random variables uniformly distributed in $[0, 1)$. Finally, given $\rho = \sqrt{2\xi}$ we get:

$$x = \cos(2\pi r_1) \times \sqrt{-2\ \ln(1 - r_2)}, \tag{3.34}$$

$$y = \sin(2\pi r_1) \times \sqrt{-2\ \ln(1 - r_2)}, \tag{3.35}$$

In this way, it is easily possible to get Gaussian variables out of uniformly distributed numbers.

As mentioned before, the steps that must be performed in order to obtain random numbers that are distributed according to a generic $\mathcal{P}(x)$ are not always easy to be done. This is particularly true for multi-dimensional cases. Performing a direct sampling is often unfeasible both in the discrete and continuous cases. Indeed, in the former case, the total number of events cannot be very large, since they must be kept in the computer memory; instead, in the latter one, we generically face to the serious limitations that we have discussed, especially in the multi-dimensional case. Therefore, a different strategy must be considered.

## 3.7 Markov Chains

In the following, for simplicity of notations, we will consider the case of a single random variable $x$ that assumes a discrete set of values, the generalization to continuous systems being straightforward. For example, $\{x\}$ may define the discrete Hilbert space of a many-body system on a finite lattice: in this case, the total number of possible configurations $\{x\}$ may easily overwhelm the computer memory, such that a direct sampling is not possible.

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

The idea to sample a generic probability distribution is to construct a non-deterministic, i.e., random, process for which a configuration $x_n$ evolves as a function of a discrete iteration time $n$ according to a stochastic dynamics:

$$x_{n+1} = F_n(x_1, \dots, x_n, \xi_n), \tag{3.36}$$

where $F_n$ is a function that may depend upon all the previous configurations up to $n$. The stochastic nature of the dynamics (3.36) is due to the fact that $F_n$ also depends upon a random variable $\xi_n$ that is distributed according to a probability density $\chi(\xi_n)$. At variance with the deterministic dynamics generated by the classical equations of motions (i.e., Newton's equations), in a stochastic dynamics, the concept of trajectory is not defined and the configurations $x_n$ are random variables that are dynamically generated along the stochastic process and are distributed according to a probability distribution that evolves with $n$. Here, the main point is to define a suitable function $F_n$ such that the configurations $x_n$ will be distributed (for large enough time $n$) according to the probability that we want to sample. In this way, we can overcome the fact that we are not able to perform a direct sampling of the probability distribution.

A particularly simple case is given by the so-called Markov chains, where the configuration at time $n + 1$ just depends upon the one at time $n$:

$$x_{n+1} = F(x_n, \xi_n), \tag{3.37}$$

where the function $F$ is taken to be time independent. Although $\xi_n$ and $\xi_{n+1}$ are independent random variables, $x_n \equiv x$ and $x_{n+1} \equiv x'$ are not independent; the joint probability distribution of these variables can be decomposed into the product of the marginal and the conditional probability, see section 2.2:

$$\mathcal{P}_{\text{joint},n}(x', x) = \omega(x'|x)\, \mathcal{P}_n(x). \tag{3.38}$$

Here, the conditional probability is such that $\omega(x'|x) \geq 0$ for all $x$ and $x'$ and satisfies the following normalization:

$$\sum_{x'} \omega(x'|x) = 1. \tag{3.39}$$

It represents the probability that, having the configuration $x$ at the iteration $n$, $x'$ appears at $n + 1$; its actual form depends upon the function $F(x, \xi)$ and the probability distribution $\chi(\xi)$.

We are now in the position of deriving the so-called *Master equation* associated to the Markov chain. Indeed, the marginal probability of the variable $x'$ is given by:

$$\mathcal{P}_{n+1}(x') = \sum_{x} \mathcal{P}_{\text{joint},n}(x', x), \tag{3.40}$$

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

so that, by using Eq. (3.38), we get:

$$\mathcal{P}_{n+1}(x') = \sum_{x} \omega(x'|x)\, \mathcal{P}_n(x). \tag{3.41}$$

This equation allows us to calculate the evolution of the marginal probability $\mathcal{P}_n(x)$ as a function of $n$, since the conditional probability $\omega(x'|x)$ is determined by the stochastic dynamics in Eq. (3.37) and does not depend upon $n$. More precisely, although the actual value of the random variable $x$ is not known deterministically, the probability distribution of $x$ is instead known at each iteration $n$, once an initial condition is given, i.e., $\mathcal{P}_0(x)$. The solution for $\mathcal{P}_n(x)$ is obtained iteratively by solving the Master equation, starting from the given initial condition up to the desired value of $n$. It is simple to simulate a Markov chain on a computer, by generating random numbers for $\xi$ at each iteration $n$, and this is the reason why Markov chains are particularly important for Monte Carlo calculations.

Before discussing in detail the properties of Markov chains and how they can be used to sample a given distribution function, we would like to show how this approach can be applied to compute $\pi$, similarly to what we have shown for the direct sampling before. In this case, instead of randomly shooting bullets in the square and counting the "hits" inside the circle (see Fig. 3.1), we imagine to perform a random walk in the square and lay down a bullet on each position that we visit, see Fig. 3.4. Starting from a random place inside the square, we move on at discrete times $n$: for example, the new position at time $n+1$ can be chosen randomly in a small square of side $\delta$, centered around the position at time $n$ (see Fig. 3.4):

$$x^{\text{new}} = x^{\text{old}} + \xi_x, \tag{3.42}$$

$$y^{\text{new}} = y^{\text{old}} + \xi_y, \tag{3.43}$$

where $\xi_x$ and $\xi_y$ are two independent random numbers that are uniformly distributed in $[-\delta/2, \delta/2]$. At the end of a long random walk, the ratio between the number of bullets inside the circle and the total number of them will approach $\pi/4$. In this approach, it is clear that the new position $(x^{\text{new}}, y^{\text{new}})$ is correlated to the old one $(x^{\text{old}}, y^{\text{old}})$, since the new coordinates cannot be farer than $\delta/2$ from the old ones. In particular, the correlation increases when $\delta$ becomes smaller and smaller. By contrast, in the direct sampling of Fig. 3.1 the position of each bullet was independent from the other ones. Nevertheless, also here, for a long enough random walk, we visit uniformly all the area of the square and the number of bullets inside the circle over the total number of bullets must be proportional to the ratio between the area of the circle and the one of the square, i.e., $\pi/4$.

There is a small caveat in this approach: what should we do if our step brings us outside the square? Surprisingly, the correct answer, as it will be clear when we will discuss the Metropolis algorithm, is that the trial must be rejected

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

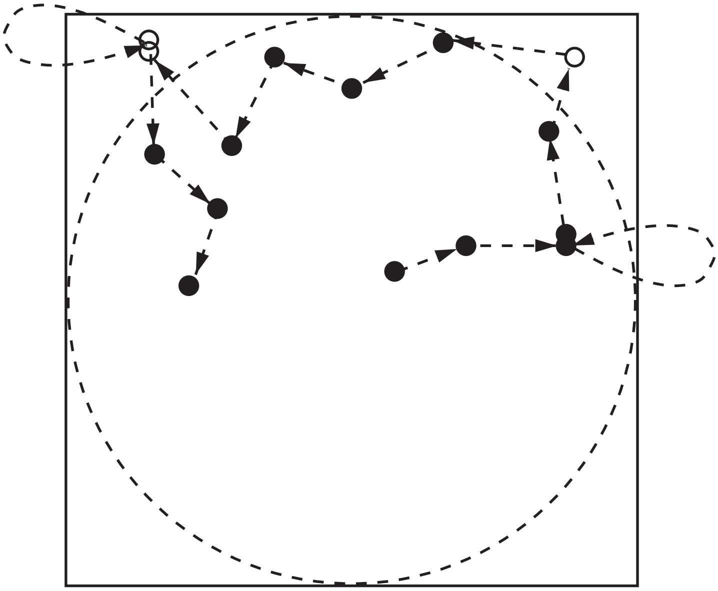

Figure 3.4 Markov chain sampling to compute $\pi$. We perform a random walk and drop off bullets behind us. The new position at time $n+1$ is chosen randomly in a small square of side $\delta$ centered around the position at time $n$. For a large number $N$ of steps the ratio between the number of bullets inside the circle and the total number of trials converges to $\pi/4$; see text. Notice that in two cases (third and eleventh steps) would have brought the bullet outside the big square; in this case, the steps is not done and a second bullet is put on top of the previous one.

(we do not move) and a bullet must be put on top of the previous one, indicating that we occupied the same position for two consecutive times. Then we try another move, until we succeed in moving inside the square.

## 3.8 Detailed Balance and Approach to Equilibrium

At this point, the natural and important question about the Markov process is to understand under which conditions the sequence of distributions $\mathcal{P}_n(x)$ converges to some limiting (i.e., equilibrium) distribution $\mathcal{P}_{\text{eq}}(x)$ or not. In the following, we will assume that $\mathcal{P}_{\text{eq}}(x) > 0$ for all the configurations $x$. Indeed, the configurations for which $\mathcal{P}_{\text{eq}}(x) = 0$ do not contribute to the final result and can be effectively discarded. For example, in classical statistical mechanics, $\mathcal{P}_{\text{eq}}(x) \propto \exp[-\beta V(x)]$ (where $\beta$ is the inverse of the temperature and $V(x)$ is the potential energy) and $\mathcal{P}_{\text{eq}}(x) \neq 0$ for all configurations with $V(x) < \infty$; those configurations with $V(x) = \infty$ do not contribute to the partition function. The general theory of Markov chains and their approach to equilibrium is a vast field of research

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

(Norris, 1997), which we do not want to discuss here. The questions that we want to address now are:

1. Does a stationary distribution $\mathcal{P}_{\rm eq}(x)$ exist?
2. Is the convergence to $\mathcal{P}_{\rm eq}(x)$ guaranteed when starting from a given *arbitrary* $\mathcal{P}_0(x)$?

The first question requires that:

$$\mathcal{P}_{\rm eq}(x') = \sum_x \omega(x'|x)\,\mathcal{P}_{\rm eq}(x). \tag{3.44}$$

In order to satisfy this stationarity requirement, it is sufficient (but not necessary) to satisfy the so-called *detailed balance* condition:

$$\omega(x'|x)\,\mathcal{P}_{\rm eq}(x) = \omega(x|x')\,\mathcal{P}_{\rm eq}(x'). \tag{3.45}$$

This relationship indicates that the number of processes undergoing a transition $x \to x'$ has to be exactly compensated, to maintain a stable stationary condition, by the same amount of reverse processes $x' \to x$. It is very simple to show that the detailed balance condition allows a stationary solution of the Master equation. Indeed, if for some $n$ we have that $\mathcal{P}_n(x) = \mathcal{P}_{\rm eq}(x)$, then:

$$\mathcal{P}_{n+1}(x') = \sum_x \omega(x'|x)\,\mathcal{P}_{\rm eq}(x) = \mathcal{P}_{\rm eq}(x') \sum_x \omega(x|x') = \mathcal{P}_{\rm eq}(x'), \tag{3.46}$$

where we used the detailed balance condition of Eq. (3.45) and the normalization condition for the conditional probability (3.39). Therefore, we have shown that the Master equation (3.41) admits stationary solutions, namely solutions that do not depend upon the (discrete) time $n$.

Now, we would like to understand under which conditions the equilibrium solution is unique and when a generic initial condition $\mathcal{P}_0(x)$ converges to it. First of all, we would like to define the concept of *periodicity* in the Markov chains. A state $x$ has a period $k$ if any return to it occurs in multiples of $k$ steps; if $k = 1$ the state is said to be *aperiodic*: in this case returns to $x$ occurs at irregular time steps. A Markov chain is aperiodic if all states are aperiodic. In Fig. 3.5, we show two examples of periodic and aperiodic Markov chains. The second concept is *reducibility*. A state $x$ is *accessible* from another one $x'$ if there is a non-zero probability to visit $x$ starting the Markov chain from $x'$. Notice that it is not required to have a finite transition probability that directly couples the two states but only that they are connected by a sequence of elementary steps. A Markov chain is *irreducible* if any state is accessible from any other one (in other words, whenever it is possible to reach any state starting from any other state). Finally, a state is said

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

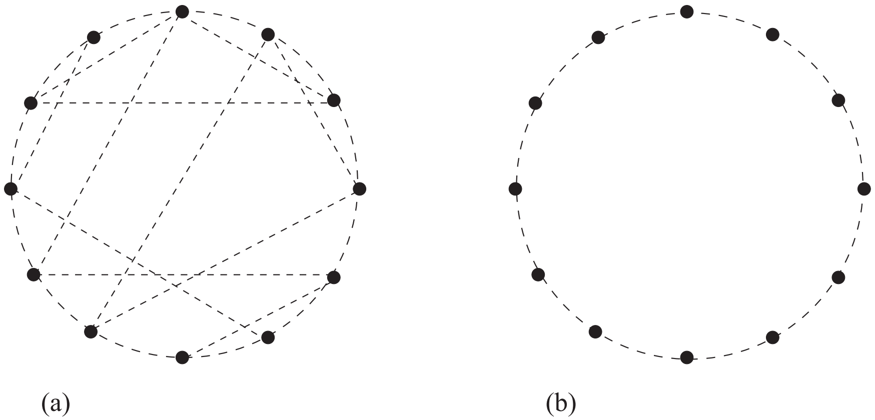

Figure 3.5 Aperiodic (a) and periodic with $k = 2$ (b) examples of Markov chains. Dashed lines indicate allowed transitions governed by the conditional probability $\omega(x'|x)$. In (b), only nearest-neighbor transitions exist; therefore, starting from any site, it is possible to come back to it only after an even number of steps (i.e., $k = 2$).

to be *positive recurrent* if the return time is finite (in other words, if it is possible to come back to it in a finite number of steps). A Markov chain is *ergodic* if it is aperiodic and all states are positive recurrent.

Let us now address the second question. First of all, the transition probability $\omega(x'|x)$ can be seen as a *non-symmetric* matrix. However, in presence of the detailed balance condition of Eq. (3.45), $\omega(x'|x)$ can be rewritten in terms of a *symmetric* matrix **H**, through a similarity transformation:

$$H_{x,x'} = H_{x',x} = \omega\left(x'|x\right) \frac{\Upsilon_0(x)}{\Upsilon_0(x')}, \tag{3.47}$$

where $H_{x,x'} \geq 0$ and $\Upsilon_0(x) = \sqrt{\mathcal{P}_{\text{eq}}(x)}$ defines a vector, whose components are strictly positive for all configurations $x$ (see at the beginning of section 3.8).

The first observation is that $\Upsilon_0(x)$ is an eigenstate of **H** with eigenvalue $\lambda_0 = 1$. Indeed, from the previous definition of Eq. (3.47), we have that:

$$\sum_{x'} H_{x,x'} \Upsilon_0(x') = \Upsilon_0(x) \sum_{x'} \omega(x'|x) = \Upsilon_0(x), \tag{3.48}$$

where we used the normalization of the conditional probability.

Now, we will prove that there are no other eigenvalues of **H** that, in absolute value, are larger than $\lambda_0$, namely $|\lambda_\alpha| \leq 1$. In order to prove this statement, we consider the square of **H** that obviously has positive eigenvalues $\lambda_\alpha^2$. This is necessary in order to exclude the existence of an eigenvalue equal to $-1$, as in the

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

case reported in Fig. 3.5(b) that is periodic with period $k = 2$. Then, we take the eigenvector $\Psi(x)$ with the largest eigenvalue $\lambda_\Psi^2$:

$$\sum_{x'} (H^2)_{x,x'} \Psi(x') = \lambda_\Psi^2 \Psi(x), \tag{3.49}$$

which implies (assuming that $\Psi(x)$ is normalized, i.e., $\sum_x \Psi^2(x) = 1$):

$$\sum_{x,x'} \Psi(x)(H^2)_{x,x'} \Psi(x') = \lambda_\Psi^2. \tag{3.50}$$

By taking the absolute values of both sides, we obtain:

$$\lambda_\Psi^2 = \left| \sum_{x,x'} \Psi(x)(H^2)_{x,x'} \Psi(x') \right| \leq \sum_{x,x'} \left| \Psi(x) \right| (H^2)_{x,x'} \left| \Psi(x') \right|. \tag{3.51}$$

For the Min-Max property of a Hermitian matrix, it follows that also $|\Psi(x)|$ is an eigenstate of $\mathbf{H}^2$ with eigenvalue equal to $\lambda_\Psi^2$. However, the matrix $\mathbf{H}^2$ is symmetric and, therefore, eigenvectors corresponding to different eigenvalues must be orthogonal. Since from Eq. (3.48) we have that $\Upsilon_0(x) = \sqrt{\mathcal{P}_{\rm eq}(x)}$ is an eigenvector of $\mathbf{H}^2$, with eigenvalue $\lambda_0^2 = 1$, then $|\Psi(x)|$ cannot be orthogonal to it. Therefore, we arrive to the conclusion that $\lambda_\Psi^2 = \lambda_0^2 = 1$.

Nevertheless, it is possible that the eigenvalue $\lambda_0^2 = 1$ is not unique, since for degenerate eigenvalues the eigenvectors are not forced to be orthogonal. However, the possibility to have a degenerate eigenvalue is ruled out by imposing the requirement that $\mathbf{H}^2$ is irreducible. Indeed, let us suppose that another eigenstate $\Upsilon'_0(x)$ of $\mathbf{H}^2$ has eigenvalue $\lambda_0^2 = 1$. Then, for any constant $\alpha$, $\Upsilon_0(x) + \alpha \Upsilon'_0(x)$ is also an eigenstate with the same eigenvalue. Moreover, from the previous discussion, also $\Phi(x) = |\Upsilon_0(x) + \alpha \Upsilon'_0(x)|$ is an eigenstate, and the constant $\alpha$ can be chosen to have $\Phi(\bar{x}) = 0$ for a particular configuration $x \equiv \bar{x}$. Then, since $\Phi(x)$ is an eigenstate of $\mathbf{H}^2$, we have that:

$$\sum_x (H^2)_{\bar{x},x} \Phi(x) = \lambda_0^2 \Phi(\bar{x}) = 0, \tag{3.52}$$

which directly implies that $\Phi(x) = 0$ for all configurations connected to $\bar{x}$ by $(H^2)_{\bar{x},x}$, since $\Phi(x)$ is non-negative and $(H^2)_{\bar{x},x}$ is strictly positive. By applying iteratively the previous condition to the new configurations connected with $\bar{x}$, we can show that all configurations that are generated have $\Phi(x) = 0$. Irreducibility implies that *all* configurations will be reached by this procedure and $\Phi(x) = 0$ is verified in the whole space of configurations. Therefore, $\Upsilon'_0(x)$ is just proportional to $\Upsilon_0(x)$ and is not a different eigenvector. This fact implies that the maximum eigenvalue $\lambda_0^2 = 1$ is non-degenerate and, in particular, $\lambda_0 = -1$ does not exist and $\lambda_0 = 1$ is unique.

We would like to mention that this result is related to the Perron-Frobenius theorem (Meyer, 2000), which applies to non-negative and ergodic matrices that, however, are not necessarily symmetric. In this case, it is possible to show that the maximum eigenvalue $\lambda_0$ is real, positive, and unique (i.e., all the other eigenvalues $\lambda_i$, which can be complex, are such that $|\lambda_i| < \lambda_0$). Moreover, all the components of the left eigenvector corresponding to $\lambda_0$ are positive.

Going back to the symmetric case, the previous results imply that any initial $\mathcal{P}_0(x)$ will converge toward the stationary distribution $\mathcal{P}_{\rm eq}(x) = \Upsilon_0^2(x)$. Indeed, by using Eq. (3.47), the Master equation (3.41) is written as:

$$\mathcal{P}_n(x) = \sum_{x'} H_{x,x'} \frac{\Upsilon_0(x)}{\Upsilon_0(x')} \mathcal{P}_{n-1}(x'); \tag{3.53}$$

then, by iterating this procedure, i.e., by expressing the marginal probability at every step in terms of the previous one, we obtain a relation between $\mathcal{P}_n(x)$ and the initial probability at step $n = 0$:

$$\mathcal{P}_n(x) = \sum_{x'} (H^n)_{x,x'} \frac{\Upsilon_0(x)}{\Upsilon_0(x')} \mathcal{P}_0(x'), \tag{3.54}$$

here the $n$-th power of the matrix $\mathbf{H}$ can be expanded in terms of its eigenvectors:

$$(H^n)_{x,x'} = \sum_\alpha \lambda_\alpha^n \Upsilon_\alpha(x') \Upsilon_\alpha(x). \tag{3.55}$$

By replacing this expansion in Eq. (3.54) we obtain:

$$\begin{aligned} \mathcal{P}_n(x) &= \Upsilon_0(x) \sum_\alpha \lambda_\alpha^n \Upsilon_\alpha(x) \left[ \sum_{x'} \frac{\Upsilon_\alpha(x')}{\Upsilon_0(x')} \mathcal{P}_0(x') \right] \\ &= \Upsilon_0^2(x) \left[ \sum_{x'} \mathcal{P}_0(x') \right] + \Upsilon_0(x) \sum_{\alpha > 0} \lambda_\alpha^n \Upsilon_\alpha(x) \left[ \sum_{x'} \frac{\Upsilon_\alpha(x')}{\Upsilon_0(x')} \mathcal{P}_0(x') \right]. \end{aligned} \tag{3.56}$$

For large $n$ all terms with $\alpha > 0$ decay exponentially, as $|\lambda_\alpha| < 1$ for $\alpha > 0$, and, therefore, only the first one survives in the above summation. Given the fact that the initial probability distribution is normalized, i.e., $\sum_{x'} \mathcal{P}_0(x') = 1$, we finally get that the probability distribution converges to the desired one:

$$\lim_{n \to \infty} \mathcal{P}_n(x) = \Upsilon_0^2(x) = \mathcal{P}_{\rm eq}(x). \tag{3.57}$$

In practical implementations, we can assume that, after a *thermalization time* $n_{\rm therm}$, the probability distribution $\mathcal{P}_n(x)$ is essentially converged to the equilibrium one $\mathcal{P}_{\rm eq}(x)$, so that the configurations $x_n$ (with $n > n_{\rm therm}$) can be used to evaluate the quantity of our interest. In most of the cases, however, subsequent configurations are not independent from each other and a finite number of steps is

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

needed to reduce the degree of correlation among them. The *correlation time* is the time that is necessary to have (essentially) independent configurations. Correlation and thermalization times coincide, being directly related to the spectrum of the transition probability (3.47). Indeed, from Eq. (3.56), which gives the approach to equilibrium of the probability distribution, it is evident that the largest eigenvalue (smaller than 1) determines the number of Markov steps that are necessary to loose memory of a given state and obtain an (essentially) independent one.

## 3.9 Metropolis Algorithm

In this section, we present a practical way of constructing a conditional probability $\omega(x'|x)$ that satisfies the detailed balance condition (3.45), such that, for large values of $n$, the configurations $x_n$ are distributed according to a given probability distribution $\mathcal{P}_{\rm eq}(x)$. Metropolis and collaborators (Metropolis et al., 1957) introduced a very simple scheme, which is also very general and can be applied to many different cases. Later, the so-called Metropolis algorithm has been extended to more general cases by W. Keith Hastings (1970) (very often, the name of "Metropolis-Hastings algorithm" is also used), As a first step, we split the transition probability $\omega(x'|x)$ into two pieces:

$$\omega(x'|x) = T(x'|x)A(x'|x), \tag{3.58}$$

where $T(x'|x)$ defines a *trial probability* that proposes the new configuration $x'$ from the present one $x$ and $A(x'|x)$ is the *acceptance probability*. In the original work by Metropolis and co-workers, the trial probability has been taken symmetric, i.e., $T(x'|x) = T(x|x')$. However, in the generalized version of the algorithm $T(x'|x)$ can be chosen with large freedom, as long as ergodicity is ensured. Then, in order to define a Markov process that satisfies the detailed balance condition, the proposed configuration $x'$ is accepted with a probability:

$$A(x'|x) = \text{Min}\left\{1, \frac{\mathcal{P}_{\rm eq}(x')T(x|x')}{\mathcal{P}_{\rm eq}(x)T(x'|x)}\right\}. \tag{3.59}$$

Without loss of generality, we can always choose $T(x|x) = 0$, namely we never propose to remain with the same configuration. Nevertheless, $\omega(x|x)$ can be finite, since the proposed move can be rejected. The actual value of $\omega(x|x)$ is fixed by the normalization condition $\sum_{x'} \omega(x'|x) = 1$.

In most cases (as in the original work by Metropolis and collaborators), it is useful to consider symmetric trial probabilities $T(x'|x) = T(x|x')$. In this case, the acceptance probability simplifies into:

$$A(x'|x) = \text{Min}\left\{1, \frac{\mathcal{P}_{\rm eq}(x')}{\mathcal{P}_{\rm eq}(x)}\right\}. \tag{3.60}$$

The proof that detailed balance is satisfied by considering the acceptance probability of Eq. (3.59) is very simple. Indeed, let us consider the case in which $x$ and $x' \neq x$ are such that $\mathcal{P}_{\mathrm{eq}}(x')T(x|x')/[\mathcal{P}_{\mathrm{eq}}(x)T(x'|x)] > 1$. In this case, we have that:

$$A(x'|x) = 1, \tag{3.61}$$

$$A(x|x') = \frac{\mathcal{P}_{\mathrm{eq}}(x)T(x'|x)}{\mathcal{P}_{\mathrm{eq}}(x')T(x|x')}; \tag{3.62}$$

then, we can directly verify that the detailed balance is satisfied:

$$T(x'|x)A(x'|x)\mathcal{P}_{\mathrm{eq}}(x) = T(x|x')A(x|x')\mathcal{P}_{\mathrm{eq}}(x'). \tag{3.63}$$

A similar proof can be obtained in the opposite case where $x$ and $x'$ are such that $\mathcal{P}_{\mathrm{eq}}(x')T(x|x')/[\mathcal{P}_{\mathrm{eq}}(x)T(x'|x)] < 1$.

Summarizing, if $x_n$ is the configuration at time $n$, the Markov chain iteration is defined in two steps:

1. Propose a move by generating a configuration $x'$ according to the transition probability $T(x'|x_n)$;
2. Accept or reject the trial move. The move is *accepted* and the new configuration $x_{n+1}$ is taken to be equal to $x'$, if a random number $\eta$, uniformly distributed in $[0, 1)$, is such that $\eta < A(x'|x_n)$; otherwise the move is *rejected* and one keeps $x_{n+1} = x_n$.

The important simplifications introduced by the Metropolis algorithm are:

- It is enough to know the equilibrium probability distribution $\mathcal{P}_{\mathrm{eq}}(x)$ up to a normalization constant: indeed, only the ratio $\mathcal{P}_{\mathrm{eq}}(x')/\mathcal{P}_{\mathrm{eq}}(x)$ is needed in calculating the acceptance rate $A(x'|x)$ in Eq. (3.59). This allows us to avoid to evaluate a computationally prohibitive normalization.
- The transition probability $T(x'|x)$ can be chosen to be very simple. For example, in a one-dimensional problem on the continuum, a new coordinate of a particle $x'$ can be taken with the rule $x' = x + \xi$, where $\xi$ is a random number uniformly distributed in $[-a, a]$, yielding $T(x'|x) = 1/(2a)$ for $x - a < x' < x + a$. Notice that in this case $T(x'|x) = T(x|x')$.
- Whenever the new configuration $x'$ is very close to the old one $x$ (e.g., for the example described in the previous point for $a$ small enough) all the moves have a high probability to be accepted, since $\mathcal{P}_{\mathrm{eq}}(x')/\mathcal{P}_{\mathrm{eq}}(x) \approx 1$, and the rejection mechanism is ineffective. However, in this case the configurations that are generated along the Markov chain are highly correlated among themselves. By contrast, proposing a new configuration that is very far from the old one can be dangerous, since $\mathcal{P}_{\mathrm{eq}}(x')/\mathcal{P}_{\mathrm{eq}}(x)$ could be very small; nevertheless, once accepted, the new

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

configuration will be very weakly correlated to the previous one. A good rule of thumb to decrease the correlation time is to tune the trial probability $T(x'|x)$ (for example, by increasing $a$ in the above example), in order to have an average acceptance rate of about 0.5, which corresponds to accepting, on average, only half of the total proposed moves. Although there is no reason that this represents the optimal choice, it usually provides a very good option.

Finally, we would like to come back to the example given in Fig. 3.4, where we have mentioned that, whenever we do a step that brings us outside the big square, we have to reject the move and put a second bullet on top of the previous one. Within the Metropolis algorithm this fact becomes clear: the trial probability $T(x'|x)$ is uniform within the small square of side $\delta$ centered around our present position; since $\mathcal{P}_{\text{eq}}(x)$ is constant inside the big square and zero outside, the trial move is always accepted if falls inside the big square and, instead, it is rejected if it goes outside. In this case, the new configuration is the same as the previous one, i.e., $x_{n+1} \equiv x_n$ so that it will enter twice in computing the average. Notice that an alternative procedure (that is equally correct) would be to consider a trial probability that does not consider moves outside the big square: in this case, when the position is close to the border, instead of having a square of side $\delta$ we must consider a small rectangle. However, in this case, we must pay attention because, in general, the trial probability will not be always symmetric and, therefore, the full acceptance probability of Eq. (3.59) must be considered.

## 3.10 How to Estimate Errorbars

Here, we would like to discuss in some detail how to determine *non-linear* functions of averages and estimate their errorbars in Monte Carlo simulations. In section 2.6, we have seen how to estimate a simple average, i.e., $\mu_x = \langle x \rangle$, and its errorbar, from a set of measurements $\{x_i\}$, with $i = 1, \dots, N$, see Eqs. (2.56) and (2.58). The same method applies to any *linear* combinations of different averages; however, in some cases, we need *non-linear* functions of the averages, such as the fluctuation of a given quantity $\langle x^2 \rangle - \langle x \rangle^2$ or a combination of different moments such as $1 - \langle x^4 \rangle / 3 \langle x^2 \rangle^2$ (i.e., the so-called Binder cumulant, which is used to locate a phase transition in classical statistical mechanics, being equal to zero in a symmetric phase and non-zero in a symmetry-broken phase).

Here, we consider non-linear functions of averages of one or more variables, $f(\mu_x, \mu_y, \dots)$. For the first example that we mentioned above, we have:

$$f\left(\mu_x, \mu_y\right) = \mu_y - \mu_x^2, \tag{3.64}$$

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

where $y = x^2$ and, therefore, $\mu_y = \langle x^2 \rangle$. Instead, for the second example:

$$f(\mu_y, \mu_z) = 1 - \frac{\mu_z}{3\mu_y^2}, \tag{3.65}$$

where $y = x^2$ and $z = x^4$, leading to $\mu_y = \langle x^2 \rangle$ and $\mu_z = \langle x^4 \rangle$, respectively.

The most natural way to estimate $f(\mu_x, \mu_y, \dots)$ from a given set of data point is to take $f(\overline{x}, \overline{y}, \dots)$, where $\overline{x} = 1/N \sum_i x_i$ and $\overline{y} = 1/N \sum_i y_i$. Indeed, this is the correct procedure; however, for non-linear functions, this kind of estimate has a bias, which is order $1/N$. Certainly, for large values of $N$, the bias is much smaller than the statistical error, which is order $1/\sqrt{N}$, and can be neglected in the calculation. Most importantly, the calculation of the errorbar on $f(\overline{x}, \overline{y}, \dots)$ requires some specification that we are going to discuss in the following. The traditional way to evaluate the incertitude is to consider the error propagation, while much more straightforward (and efficient) ways are based upon the bootstrap or jackknife procedures (Young, 2012). Here, we start by explaining the error propagation (which is very rarely used in practice) and then discuss the bootstrap and jackknife approaches which are commonly used in most cases.

### 3.10.1 Error Propagation

In the following, for simplicity, we consider a function that only depends upon two expectation values, i.e., $f(\mu_x, \mu_y)$. The generalization to the most general case is straightforward. In order to find the bias and the errorbar of $f(\overline{x}, \overline{y})$, we can expand this quantity around $f(\mu_x, \mu_y)$:

$$\begin{aligned} f(\overline{x}, \overline{y}) \approx f\left(\mu_x, \mu_y\right) &+ \left(\partial_{\mu_x} f\right) \Delta_x + \left(\partial_{\mu_y} f\right) \Delta_y \\ &+ \frac{1}{2}\left(\partial^2_{\mu_x,\mu_x} f\right) \Delta_x^2 + \left(\partial^2_{\mu_x,\mu_y} f\right) \Delta_x \Delta_y + \frac{1}{2}\left(\partial^2_{\mu_y,\mu_y} f\right) \Delta_y^2, \end{aligned} \tag{3.66}$$

where $\Delta_x = (\overline{x} - \mu_x)$ and $\Delta_y = (\overline{y} - \mu_y)$. The leading contribution to the bias comes from the second-order terms, since the first-order terms in $\Delta_x$ and $\Delta_y$ average to zero when the procedure is repeated many times, i.e., $\langle \Delta_x \rangle = \langle \Delta_y \rangle = 0$. Instead, the second-order terms have, in general, finite expectation values:

$$\left\langle \Delta_x^2 \right\rangle = \langle \overline{x}^2 \rangle - \langle \overline{x} \rangle^2 = \sigma_{\overline{x}}^2 = \frac{\sigma_x^2}{N}, \tag{3.67}$$

$$\left\langle \Delta_y^2 \right\rangle = \langle \overline{y}^2 \rangle - \langle \overline{y} \rangle^2 = \sigma_{\overline{y}}^2 = \frac{\sigma_y^2}{N}, \tag{3.68}$$

$$\langle \Delta_x \Delta_y \rangle = \langle \overline{xy} \rangle - \langle \overline{x} \rangle \langle \overline{y} \rangle = \sigma_{\overline{xy}}^2 = \frac{\sigma_{xy}^2}{N}, \tag{3.69}$$

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

where we used Eq. (2.58) and $\sigma_x^2 = \langle x^2 \rangle - \langle x \rangle^2$, $\sigma_y^2 = \langle y^2 \rangle - \langle y \rangle^2$, and $\sigma_{xy}^2 = \langle xy \rangle - \langle x \rangle \langle y \rangle$. Therefore, we have that:

$$\langle f(\bar{x}, \bar{y}) \rangle - f(\mu_x, \mu_y) \approx \frac{1}{2N} \left[ \left( \partial^2_{\mu_x, \mu_x} f \right) \sigma_x^2 + 2 \left( \partial^2_{\mu_x, \mu_y} f \right) \sigma_{xy}^2 + \left( \partial^2_{\mu_y, \mu_y} f \right) \sigma_y^2 \right], \tag{3.70}$$

which demonstrates that the difference between the exact value $f(\mu_x, \mu_y)$ and its estimation given by $f(\bar{x}, \bar{y})$ is $O(1/N)$. Notice that, whenever the function is linear in the expectation values, the second derivatives vanish and, therefore, there is no bias. Usually, we do not care about this bias, since it is much smaller than the statistical error, which is proportional to $1/\sqrt{N}$. In fact, the leading contribution to the errorbar associated to $f(\bar{x}, \bar{y})$ can be easily computed by considering the expansion (3.66). Then, the variance is given by:

$$\begin{aligned} \sigma_f^2 &= \left\langle f^2(\bar{x}, \bar{y}) \right\rangle - \langle f(\bar{x}, \bar{y}) \rangle^2 \\ &= (\partial_{\mu_x} f)^2 \left\langle \Delta_x^2 \right\rangle + 2(\partial_{\mu_x} f)(\partial_{\mu_y} f) \langle \Delta_x \Delta_y \rangle + \left( \partial_{\mu_y} f \right)^2 \left\langle \Delta_y^2 \right\rangle \\ &= \frac{1}{N} \left[ \left( \partial_{\mu_x} f \right)^2 \sigma_x^2 + 2(\partial_{\mu_x} f) \left( \partial_{\mu_y} f \right) \sigma_{xy}^2 + \left( \partial_{\mu_y} f \right)^2 \sigma_y^2 \right], \end{aligned} \tag{3.71}$$

where all second-order terms in Eq. (3.66) cancel when considering the difference between $\langle f^2(\bar{x}, \bar{y}) \rangle$ and $\langle f(\bar{x}, \bar{y}) \rangle^2$. As usual, $\sigma_f^2$ can be computed by substituting $\sigma_x^2$, $\sigma_{xy}^2$, and $\sigma_y^2$ with their estimations obtained from $s_x$, $s_{xy}$, and $s_y$, see Eq. (2.63). The main drawback of this approach is that it requires the exact calculation of all partial derivatives, besides keeping track of all the variances and covariances. In the following, we present two simple techniques that do not require any analytical calculation.

### 3.10.2 The Bootstrap Method

Within the bootstrap approach, we generate $N_{\text{boot}}$ data sets each containing *exactly* $N$ points just by selecting randomly the points from the original data set, which is denoted by $\{x_i\}$, with $i = 1, \ldots, N$. Along this procedure, the probability that a data point is selected is $1/N$ and, therefore, on average it will appear once in each data set. However, in the new sample, each point of the original data set can appear more than once (or it may not appear at all).

Let us denote by $n_{i,\alpha}$ the number of times that $x_i$ appears in the bootstrap $\alpha$, with $\alpha = 1, \ldots, N_{\text{boot}}$. Since each bootstrap data set contains $N$ data points, we have the following constraint:

$$\sum_{i=1}^{N} n_{i,\alpha} = N. \tag{3.72}$$

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

Whenever the number of data sets $N_{\text{boot}}$ is large enough, it reproduces the correct averages; in particular, we denote:

$$[n_i]_{\text{boot}} = \frac{1}{N_{\text{boot}}} \sum_{\alpha=1}^{N_{\text{boot}}} n_{i,\alpha}, \tag{3.73}$$

$$\left[n_i^2\right]_{\text{boot}} = \frac{1}{N_{\text{boot}}} \sum_{\alpha=1}^{N_{\text{boot}}} n_{i,\alpha}^2. \tag{3.74}$$

Since the probability that $x_i$ occurs $n_{i,\alpha}$ times in the bootstrap $\alpha$ is given by the binomial probability:

$$P(n_{i,\alpha}) = \frac{N!}{n_{i,\alpha}!\ (N - n_{i,\alpha})!} p^{n_{i,\alpha}} (1-p)^{N-n_{i,\alpha}}, \tag{3.75}$$

where $p = 1/N$, the mean and variance are given by:

$$[n_i]_{\text{boot}} = Np = 1, \tag{3.76}$$

$$\left[n_i^2\right]_{\text{boot}} - [n_i]_{\text{boot}}^2 = Np(1-p) = 1 - \frac{1}{N}. \tag{3.77}$$

Notice that, because of the presence of the constraint of Eq. (3.72), the values of $n_{i,\alpha}$ and $n_{j,\alpha}$ for $i \neq j$ in the same bootstrap data set are not independent, but they have a (small) correlation (that goes to zero as $N \to \infty$). Indeed, by squaring Eq. (3.72) and averaging over $N_{\text{boot}}$, we obtain:

$$\frac{1}{N^2} \sum_{i,j} \left[n_i n_j\right]_{\text{boot}} = 1, \tag{3.78}$$

which can be rewritten by using the fact that $[n_i]_{\text{boot}} = 1$ as:

$$\frac{1}{N^2} \sum_{i,j} \left( \left[n_i n_j\right]_{\text{boot}} - [n_i]_{\text{boot}} \left[n_j\right]_{\text{boot}} \right) = 0. \tag{3.79}$$

By splitting the terms with $i = j$ from the others (that give all equal contributions) and using Eq. (3.77), we obtain:

$$\frac{1}{N}\left(1 - \frac{1}{N}\right) + \left(\frac{N-1}{N}\right) \left( \left[n_i n_j\right]_{\text{boot}} - [n_i]_{\text{boot}} \left[n_j\right]_{\text{boot}} \right) = 0, \tag{3.80}$$

which finally leads to:

$$\left[n_i n_j\right]_{\text{boot}} - [n_i]_{\text{boot}} \left[n_j\right]_{\text{boot}} = -\frac{1}{N}. \tag{3.81}$$

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

We are now in the position to be able to compute the averages of different quantities. In particular, let us start with the simple case of $\mu = \langle x \rangle$. The average for a given bootstrap data set is given by:

$$x_\alpha^B = \frac{1}{N} \sum_{i=1}^{N} n_{i,\alpha} x_i. \tag{3.82}$$

The final bootstrap estimate of $\mu$ is then given by:

$$\overline{x^B} = \frac{1}{N_\text{boot}} \sum_{\alpha=1}^{N_\text{boot}} x_\alpha^B = \frac{1}{N} \sum_{i=1}^{N} [n_i]_\text{boot}\, x_i = \frac{1}{N} \sum_{i=1}^{N} x_i = \overline{x}, \tag{3.83}$$

where we have used Eq. (3.76). Therefore, the average over the bootstrap data sets gives exactly the average of the original data set $\{x_i\}$. Then, to compute the variance, we notice that:

$$\overline{(x^B)^2} = \frac{1}{N_\text{boot}} \sum_{\alpha=1}^{N_\text{boot}} \left(x_\alpha^B\right)^2 = \frac{1}{N^2} \sum_{i,j} \left[n_i n_j\right]_\text{boot} x_i x_j. \tag{3.84}$$

By using Eqs. (3.77) and (3.81), we get :

$$s_{x^B}^2 = \overline{(x^B)^2} - \left(\overline{x^B}\right)^2 = \frac{1}{N^2}\left(1 - \frac{1}{N}\right) \sum_{i=1}^{N} x_i^2 - \frac{1}{N^3} \sum_{i \neq j} x_i\, x_j = \frac{s^2}{N}, \tag{3.85}$$

where $s^2$ is the variance defined in Eq. (2.61). Therefore, the expectation values are:

$$\left\langle \overline{x^B} \right\rangle = \langle \overline{x} \rangle = \mu, \tag{3.86}$$

$$\left\langle s_{x^B}^2 \right\rangle = \left(\frac{N-1}{N^2}\right) \sigma^2 = \left(\frac{N-1}{N}\right) \sigma_{\overline{x}}^2, \tag{3.87}$$

where we used Eqs. (2.58) and (2.63). In summary, the bootstrap estimate of the variance $\sigma_{\overline{x}}^2$ is given by $N/(N-1)s_{x^B}^2$. This procedure does not require the calculation of the partial derivatives of Eq. (3.71); therefore, it is much easier to implement than the straightforward error propagation.

Similarly, we can easily compute the bootstrap estimate of $f(\mu_x, \mu_y)$:

$$f_\alpha^B = f\left(x_\alpha^B, y_\alpha^B\right). \tag{3.88}$$

The final estimate is given by averaging the bootstrap data set:

$$\overline{f^B} = \frac{1}{N_\text{boot}} \sum_{\alpha=1}^{N_\text{boot}} f_\alpha^B, \tag{3.89}$$

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

while the errorbar is obtained from:

$$s^2_{f^B} = \overline{(f^B)^2} - \left(\overline{f^B}\right)^2 . \tag{3.90}$$

Indeed, it can be shown that by expanding $f^B_\alpha$ around $f(\mu_x, \mu_y)$, similarly to what has been done in Eq. (3.66):

$$f^B_\alpha = f\left(x^B_\alpha, y^B_\alpha\right) \approx f(\mu_x, \mu_y) + (\partial_{\mu_x} f)\Delta^B_{\alpha,x} + (\partial_{\mu_y} f)\Delta^B_{\alpha,y}, \tag{3.91}$$

where $\Delta^B_{\alpha,x} = (x^B_\alpha - \mu_x)$ and $\Delta^B_{\alpha,y} = (y^B_\alpha - \mu_y)$, and then, following the steps of Eqs. (3.83) and (3.85), we obtain that the bootstrap estimate of $\sigma^2_{\overline{f}}$ is given by $N/(N-1)s^2_{f^B}$.

The drawback of the bootstrap method is that about 37% of the points are not selected in a single bootstrap. Indeed, the probability that a data point is not taken in a given bootstrap $\alpha$ is $(1-1/N)^N$, which for large $N$ approaches $e^{-1} \approx 0.37$. Therefore, much of the information of the original data set is not used. In the following, we describe an alternative approach that does not suffer from this problem.

### 3.10.3 The Jackknife Method

Within the jackknife approach, we define the $i$-th estimate to be the average over all the data in the original data set *except* the point $i$. As before, we start the discussion for the simple case of $\mu = \langle x \rangle$. Here, we have:

$$x^J_i = \frac{1}{N-1} \sum_{j=1\ (j \neq i)}^{N} x_j = \frac{N}{N-1}\overline{x} - \frac{1}{N-1} x_i. \tag{3.92}$$

The final jackknife estimate of $\mu$ is given by:

$$\overline{x^J} = \frac{1}{N} \sum_{i=1}^{N} x^J_i = \frac{N}{N-1}\overline{x} - \frac{1}{N-1}\overline{x} = \overline{x}. \tag{3.93}$$

In order to compute the errorbar associated to it, we notice that:

$$\overline{(x^J)^2} = \frac{1}{N} \sum_{i=1}^{N} \left(x^J_i\right)^2 = \overline{x}^2 + \frac{1}{(N-1)^2} \left(\overline{x^2} - \overline{x}^2\right). \tag{3.94}$$

The jackknife estimate of the variance is obtained from:

$$s^2_{x^J} = \overline{(x^J)^2} - \left(\overline{x^J}\right)^2 = \frac{1}{(N-1)^2} \left(\overline{x^2} - \overline{x}^2\right) = \frac{s^2}{(N-1)^2}. \tag{3.95}$$

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

The expectation value of Eq. (3.93) is obviously the mean $\mu$, while the expectation value of Eq. (3.95) is the variance of the mean divided by $(N-1)$:

$$\langle \overline{x^J} \rangle = \langle \bar{x} \rangle = \mu, \tag{3.96}$$

$$\langle s^2_{x^J} \rangle = \frac{\sigma^2}{N(N-1)} = \frac{\sigma^2_{\bar{x}}}{N-1}, \tag{3.97}$$

where we used again Eqs. (2.58) and (2.63). In summary, the jackknife estimate of the variance $\sigma^2_{\bar{x}}$ is given by $(N-1)s^2_{x^J}$. Notice that in the jackknife approach there is a $(N-1)$ factor that multiplies $s^2_{x^J}$, this is due to the fact that the new samples are very correlated, since they would be all equal except that each one neglects just one point.

Similarly to what has been discussed for the bootstrap approach, the jackknife technique can be used to estimate any function of expectation values. We have just to define:

$$f^J_i = f\left(x^J_i, y^J_i\right). \tag{3.98}$$

The final estimate is given by averaging the jackknife data sets:

$$\overline{f^J} = \frac{1}{N}\sum_{i=1}^{N} f^J_i, \tag{3.99}$$

while the errorbar is obtained from:

$$s^2_{f^J} = \overline{(f^J)^2} - \left(\overline{f^J}\right)^2. \tag{3.100}$$

Again, by expanding $f^J_i$ around $f(\mu_x, \mu_y)$:

$$f^J_i = f\left(x^J_i, y^J_i\right) \approx f(\mu_x, \mu_y) + (\partial_{\mu_x} f)\Delta^J_{i,x} + (\partial_{\mu_y} f)\Delta^J_{i,y}, \tag{3.101}$$

where $\Delta^J_{i,x} = (x^J_i - \mu_x)$ and $\Delta^J_{i,y} = (y^J_i - \mu_y)$, and then following the procedure of Eqs. (3.93) and (3.95), we obtain that the jackknife estimate of the variance $\sigma^2_{\bar{f}}$ is $(N-1)s^2_{f^J}$.

## 3.11 Errorbars in Correlated Samplings

Let us now discuss the case where the original data set consists of *correlated* points, i.e., the set of $\{x_i\}$ with $i = 1, \ldots, N$ are not independent. This situation appears whenever the data set is not generated by a direct sampling but instead by a Markov process. Indeed, in this case, the subsequent points will possess some degree of correlation, since, in general, it is very hard to accept a new configuration which is

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

completely decorrelated from the previous one. Nevertheless, also for a correlated data set, the average gives an unbiased estimation of the exact mean:

$$\langle \bar{x} \rangle = \frac{1}{N} \sum_{i=1}^{N} \langle x_i \rangle = \langle x \rangle, \tag{3.102}$$

since the average is a linear function of the data set $\{x_i\}$. By contrast, the quantity $s^2$ defined in Eq. (2.61) is no longer an unbiased estimator of the exact variance:

$$\langle s^2 \rangle = \frac{1}{N} \sum_{i} \langle x_i^2 \rangle - \frac{1}{N^2} \sum_{i,j} \langle x_i x_j \rangle \neq \left( \frac{N-1}{N} \right) \sigma_x^2, \tag{3.103}$$

which is due to the fact that $\langle x_i x_j \rangle \neq \langle x_i \rangle \langle x_j \rangle$. In general, the estimation of the variance using $s^2$ leads to underestimating the errorbars.

To overcome this problem, we can perform the so-called *binning technique* (also called *block analysis*). We divide up the data set $\{x_i\}$ derived from a long Markov chain into several ($N_{\text{bin}}$) segments (i.e., bins), each of length $L_{\text{bin}} = N/N_{\text{bin}}$. On each bin $j$, with $j = 1, \ldots, N_{\text{bin}}$, we define the partial average:

$$x^j = \frac{1}{L_{\text{bin}}} \sum_{i=(j-1)L_{\text{bin}}+1}^{jL_{\text{bin}}} x_i. \tag{3.104}$$

Clearly, the average over the bins is equal to the original average:

$$\overline{x^j} = \frac{1}{N_{\text{bin}}} \sum_{j=1}^{N_{\text{bin}}} x^j = \bar{x}. \tag{3.105}$$

However, the probability distribution of the “new” (binned) variables $x^j$ is different from the one of the $x_i$’s. Given the definition of Eq. (3.104), the variance of the $x^j$’s is generally smaller than the one of the $x_i$’s. This fact can be easily understood in the case where the original variables are already independent and $N_{\text{bin}}$ is large: in this case, the central limit theorem, discussed in section 2.7, holds and implies that the variance of the binned variables is $1/L_{\text{bin}}$ smaller than the one of the original variables. In the general case, by increasing the bin length $L_{\text{bin}}$, the new variables $x^j$ will be more and more uncorrelated among each other, eventually becoming independent random variables. Indeed, after the equilibration part of the Markov process, that we assume already performed at the step $i = 1$, the average correlation function:

$$C(n-m) = \langle x_n x_m \rangle - \langle x_n \rangle \langle x_m \rangle, \tag{3.106}$$

depends only on the discrete time difference $n-m$ (since stationarity implies time-homogeneity) and approaches zero exponentially as $C(n-m) \propto e^{-|n-m|/\tau}$, where $\tau$ (in units of the discrete time-step) is the correlation time in the Markov chain.

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press

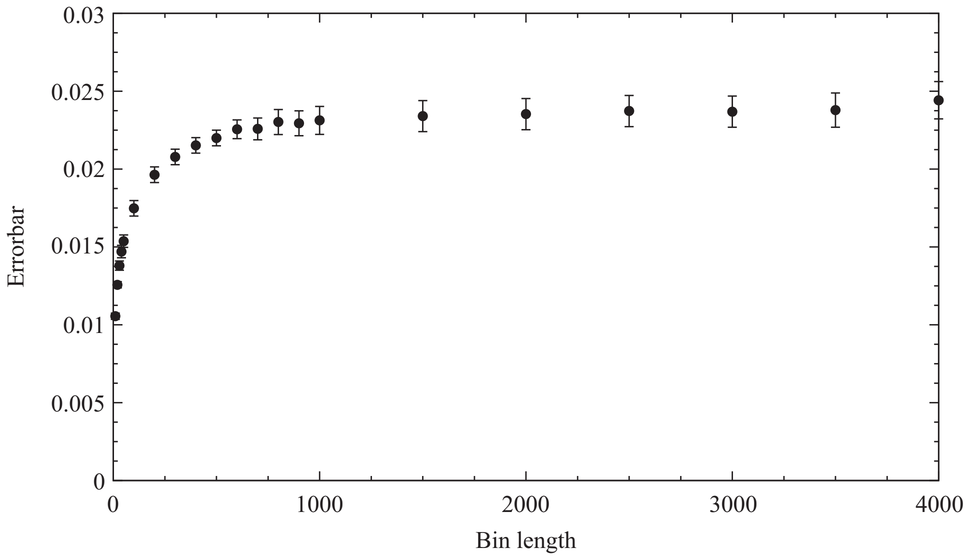

Figure 3.6 The errorbar $s_{\bar{x}}$ estimated with the binning technique, as a function of the bin length $L_{\text{bin}}$ for a typical simulation based upon a Markov chain process.

Therefore, if we take $L_{\text{bin}}$ to be sufficiently larger than $\tau$, then the different bin averages $x^j$ can be reasonably considered to be independent random variables and the variance can be easily estimated:

$$s^2_{\text{bin}} = \frac{1}{N_{\text{bin}}} \sum_{j=1}^{N_{\text{bin}}} \left(x^j - \bar{x}\right)^2 . \tag{3.107}$$

Then, the variance of the average value (3.105) is given by:

$$s^2_{\bar{x}} = \frac{s^2_{\text{bin}}}{N_{\text{bin}}}. \tag{3.108}$$

The errorbar on the average value is given by the square root of $s^2_{\bar{x}}$. Notice that, in the case where the original variables are already uncorrelated, the reduced (i.e., $1/L_{\text{bin}}$) variance of the binned variables is compensated with the smaller number (i.e., $N_{\text{bin}}$) of them, leading to the same variance of the mean values before and after the binning procedure. In Fig. 3.6, we report a typical case of correlated measures, where the errorbar $s_{\bar{x}}$ depends upon the length of the bin and shows a plateaux for sufficiently large values of $L_{\text{bin}}$.

https://doi.org/10.1017/9781316417041.004 Published online by Cambridge University Press