Computational Methods - Lessons 9-10-11 (Luxembourg, 11-18-25.11.2020)

Monte Carlo methods

Matteo Barborini*,[a], Valentin Vassilev Galindo*,[b], Prof. Alexandre Tkatchenko*

Theoretical Chemical Physics group

Faculté des Sciences, des Technologies et de Médecine - Département Physique et sciences des matériaux

*Campus Limpertsberg, Université du Luxembourg - 162 A, avenue de la Faïencerie, L-1511 Luxembourg

[a]matteo.barborini@uni.lu, [b]valentin.vassilev@uni.lu

1. Monte Carlo Methods

The term Monte Carlo refers to different numerical techniques used to compute complex integrals that are based on stochastic samplings. In this lesson we will discuss the basic concepts regarding Monte Carlo integration techniques[1] and the stochastic procedures that are commonly used. We will also recall some of the basic statistical concepts that are required to understand the procedures.

1.1. Simple numerical Integration with Monte Carlo.

The main application of the Monte Carlo method, which underlines its great simplicity and power, is the numerical computation of[2] integrals in multi-dimensional spaces.

The common numerical techniques used to estimate integrals are essentially based on the uniform subdivision of the integration volume $\mathcal{V}$ into hypercubes of volume $\Delta\mathcal{V}$, and reduce the calculation to the sum $\sum_{i=1}^{N} f(\mathbf{r}_i)\,\Delta\mathcal{V}$ of a finite number N of terms. The accuracy of the integral estimation through these quadrature methods increases as the volume of the hypercubes decreases, and this corresponds to an increase in the total number of terms in the sum and thus of the computational demand. In the case of one-dimensional integrals in which the integration domain is divided into n identical segments Δx, the computational expenditure increases linearly as these decrease. As the dimensions d of the space increases, assuming a hypercubic integration domain $\mathcal{V}$ of which each edge is divided into n equal intervals, the number of terms in the sum increases as the power of the dimensions $N = n^d$. As soon as d is significantly greater than 1, for example for integrals involving systems of M particles' coordinates ($d = 3M$), the quadrature methods become computationally more and more expensive and difficult to manage. A good alternative to these approaches is that of randomly sampling the integration domain: this is the basic idea behind the Monte Carlo methods.

For now, for the sake of simplicity, let us assume that we have a one dimensional function $f(x)$ that depends on a variable x. We assume that the integration domain is a segment of length $\mathcal{V}$, then we can randomly extract values of the variable x according to a uniform probability density in $\mathcal{V}$, defined as $p(x) = \frac{1}{\mathcal{V}}$, and compute for each values of x the values of the function $f(x)$. It is important to notice that $p(x)$ is a correct probability density since it satisfies the conditions:

$$p(x) > 0 \quad \forall\, x \in \mathcal{V} \qquad \text{and} \qquad \int_{\mathcal{V}} p(x)\,dx = 1.$$

The Monte Carlo estimation of the integral will then be equal to the expectation value of the functions $f(x)$ computed on the values of the random variable x extracted with probability $p(x)$ in the integration volume $\mathcal{V}$:

$$\int_{\mathcal{V}} f(x)\,dx = \mathcal{V}\int_{\mathcal{V}} f(x)\,\frac{1}{\mathcal{V}}dx = \mathcal{V}\int_{\mathcal{V}} f(x)\,p(x)dx = \mathcal{V}\,\langle f(x)\rangle_{p(x)} = \mathcal{V}\mathsf{E}\left[f(x)\right] \tag{1}$$

where the expectation value of the function $f(x)$ for a continuous random variable x is defined as the integral

$$\mathsf{E}\left[f(x)\right] = \int_{\mathcal{V}} f(x)\,p(x)dx. \tag{2}$$

Associated to this expectation value we define the variance of the function $f(x)$ in the interval $\mathcal{V}$ as

$$\mathrm{Var}\left[f(x)\right] = \left\langle \left(f(x) - \langle f(x)\rangle_{p(x)}\right)^2 \right\rangle_{p(x)} = \int_{\mathcal{V}} \left(f(x) - \mathsf{E}\left[f(x)\right]\right)^2 p(x)dx \tag{3}$$

which is a measure of how much the the values of $f(x)$ differ from its expectation value $\mathsf{E}\left[f(x)\right]$.

Now, since x is a continuous variable, the population is infinite in the finite interval, yet we can approximate the integral by extracting a sub-set of $\mathcal{N}$ elements of the population, a **sample**, $x_1, x_2, \ldots, x_{\mathcal{N}}$, for which we have the set of values of the function $f(x_1), f(x_2), \ldots, f(x_{\mathcal{N}})$. We define the average value of the function $f(x)$ on the sample as:

$$\bar{f} = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} f(x_i). \tag{4}$$

so that the approximate value of the integral will be

$$\int_{\mathcal{V}} f\,(x)\,dx \approx \mathcal{V}\bar{f}. \tag{5}$$

We can associate to this mean value an error associated with its estimation, the **sample standard error** which is given by

$$\sigma = \sqrt{\frac{\mathrm{Var}\,[f(x)]}{\mathcal{N}}}. \tag{6}$$

Yet,as for the expectation value of the population, also the variance of the population is unknown, so we can obtain an estimation of the variance of the population from the sample as

$$s^2 = \frac{1}{\mathcal{N}-1} \sum_{i=1}^{\mathcal{N}} \left(f(x_i) - \bar{f}\right)^2. \tag{7}$$

The difference between the population variance (eq. 3) and the sample variance (eq. 7) is in the **Bessel factor** of $\frac{\mathcal{N}}{\mathcal{N}-1}$ that takes into account the fact that the variance is estimated on a finite sample (as is the mean value), introducing a bias related to the linear dependency between s^2 and $\bar{f}$. We can finally conclude that the Monte Carlo estimation of the integral on a finite sample $\mathcal{N}$ of values of $f(x)$ will written as

$$\int_{\mathcal{V}} f\,(x)\,dx \approx \mathcal{V}\bar{f} \pm \mathcal{V}\frac{s}{\sqrt{N}} \tag{8}$$

to which we associate a standard error that depends on the sample variance and decreases as the inverse of the square root of the number of samples.
This result comes from the **central limit theorem** which establishes that given a set of $\mathcal{N}$ points x_i not correlated, and distributed according to the normalized probability density $p\,(x)$, it is possible to define a new random variable

$$F = \frac{f\,(\mathbf{r}_1) + f\,(\mathbf{r}_2) + \ldots + f\,(\mathbf{r}_N)}{N}, \tag{9}$$

with the mean value defined in eq. 4 and the variance defined in eq 7. The central limit theorem states that for $\mathcal{N} \to \infty$ the random variable F is normally distributed, with an average value equal to $E[f]$ (eq. 2) and a variance $\mathrm{Var}[f]$ defined in eq. 3.

This procedure can be easily extended in the general case in which the function $f(\mathbf{r})$ depends on multiple variables. In this case we can obtain $\mathcal{N}$ values of the function $f(\mathbf{r})$ by randomly choosing $\mathbf{r}$ vectors in the integration space $\mathcal{V}$ (each variable is chosen randomly in its integration region with a uniform distribution) and again we will have that

$$\int_{\mathcal{V}} f\,(\mathbf{r})\,d\mathbf{r} \approx \mathcal{V}\bar{f} \pm \mathcal{V}\frac{s}{\sqrt{N}} \tag{10}$$

with $\bar{f} = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} f(\mathbf{r}_i)$ and $s^2 = \frac{1}{\mathcal{N}-1} \sum_{i=1}^{\mathcal{N}} \left(f(\mathbf{r}_i) - \bar{f}\right)^2$. An example of how to extract numbers in a uniform distribution in the interval $[0, 1]$ can be found in the python script `les09.1.py`. In the python script `les09.2.py` you can find an example on how to extract random number with a uniform distribution in a general interval.

1.2. Area of a circle. The most classical example to prove the Monte Carlo approach is the computation of the area of the circle (or the calculus of π). Let us define a function

$$f(x,y) = \begin{cases} 1 & x^2+y^2 \leq r^2 \\ 0 & x^2+y^2 > r^2 \end{cases} \tag{11}$$

in the integration intervals $x \in [-1:1]$ and $y \in [-1:1]$, that is $f(x,y)$ is non-zero inside the area of the circle radius r and zero elsewhere. We then write the integral as

$$\int_{\mathcal{V}} f(x,y)\, dxdy = 4r^2 \int_{\mathcal{V}} f(x,y)\, \frac{1}{4r^2} dxdy \tag{12}$$

where the integration volume is the square of area $\mathcal{V} = 4r^2$. We now sample $\mathcal{N}$ points with uniform probability $p(x,y) = \frac{1}{4r^2}$, *ie* x and y are extracted independently in their integration intervals of length $2r$, so that the Monte Carlo estimation will be

$$\int_{\mathcal{V}} f(x,y)\, dxdy \approx 4r^2 \bar{f} \pm \frac{4r^2}{\sqrt{\mathcal{N}}} \sqrt{\frac{1}{\mathcal{N}-1} \sum_{i=1}^{\mathcal{N}} \left(f(x_i,y_i) - \bar{f}\right)^2} \tag{13}$$

where $\bar{f} = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} f(x,y) \approx \frac{\pi}{4}$, and it corresponds on counting the number of points within the circle and dividing it by the total number of points $\mathcal{N}$. A script in python can be found in the file `les09.3.py`.

1.3. Importance Sampling. Sampling the space uniformly is not always convenient: when the sample variance per point[1] is very large, it may become necessary, in order to obtain an acceptable error, to sample an unacceptably large number of points. To overcome this problem one can applying the so called **importance sampling**, that is based on the idea to rewrite the function $f(\mathbf{r})$ as the product of two functions $h(\mathbf{r})p(\mathbf{r})$ where $h(\mathbf{r})$ is an 'almost constant' function, or in general a function with lower variance than $f(\mathbf{r})$, and $p(\mathbf{r})$ is a distribution function of the random variables **r** that can be directly sampled, and satisfies the conditions:

$$p(\mathbf{r}) > 0 \quad \text{e} \quad \int_{\mathcal{V}} p(\mathbf{r})\, d\mathbf{r} = 1.$$

In this case the integral becomes

$$\int_{\mathcal{V}} f(\mathbf{r})\, d\mathbf{r} = \int_{\mathcal{V}} h(\mathbf{r})\, p(\mathbf{r})\, d\mathbf{r} \quad \text{with} \quad h(\mathbf{r}) = \frac{f(\mathbf{r})}{p(\mathbf{r})}.$$

and the associated Monte Carlo estimation will be

$$\int_{\mathcal{V}} f(\mathbf{r})\, d\mathbf{r} \approx \bar{h} \pm \frac{s_h}{\sqrt{\mathcal{N}}}. \tag{14}$$

where $\bar{h}$ is the average $\bar{h} = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} h(\mathbf{r}_i)$ and s_h^2 is the variance of the function $h(\mathbf{r}_i)$ defined as $s_h^2 = \frac{1}{\mathcal{N}-1} \sum_{i=1}^{\mathcal{N}} \left(h(\mathbf{r}_i) - \bar{h}\right)^2$.

It is understood that the strategy is more effective the more it is possible to determine $h(\mathbf{r})$ such that it is nearly a constant in the integration region, *ie* so that its variance s_h^2 is very small, leading to a fast convergence of the integral. Mathematically, the best expression of $p(\mathbf{r})$ (and thus of $h(x)$), the one that minimizes the variance, is

$$p(\mathbf{r}) = \frac{|f(\mathbf{r})|}{\int_{\mathcal{V}} |f(\mathbf{r})|\, d\mathbf{r}}; \tag{15}$$

however, this result leads to the obvious paradox that the exact choice of $p(\mathbf{r})$ depends on the prior knowledge of the integral that one is trying to calculate numerically; therefore the best choice is precluded and at most we can hope to assume, thanks to the physical intuition on the problem under examination, a $p(\mathbf{r})$ not too different from the ideal one.

It is also important to clarify that the volume $\mathcal{V}$, in the case of importance sampling, is the integral of the distribution $p(\mathbf{r})$; therefore, in order to solve the integral, it is essential to ensure that $p(\mathbf{r})$ is normalizable, that is, that its integral is finite.

From what has been said so far, we understand the importance that also the variance associated with the function $h(\vec{r})$ is finite, *ie* that its integral exists and is finite

$$I = \int_{\mathcal{V}} \left(h(\mathbf{r}) - \bar{h}\right)^2 p(\mathbf{r})\, d\mathbf{r} < \infty. \tag{16}$$

[1] the variance associated with the distribution of the values that the $f(\mathbf{r})$ function assumes during the $\mathcal{N}$ samples.

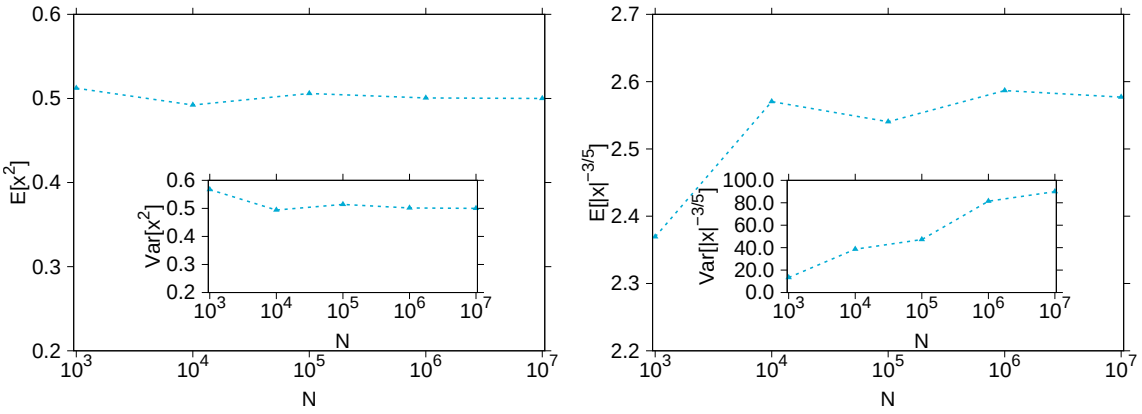

Figure 1: The left graph shows the integral of the function $h(x) = x^2$, which, together with its variance per point, converges to the exact value for $N \to \infty$. The graph on the right instead shows how the variance per point of the estimate of the integral of $h(x) = |x|^{-3/5}$ for various samples, never converges to a finite value but fluctuates, preventing a correct MC estimate.

1.4. Integrals with $p(\mathbf{r})$ as a Gaussian distribution. To better understand the importance of this requirement, two simple examples of Monte Carlo integration can be made.
As a first example we can consider solving the integral

$$I = A \int_{-\infty}^{+\infty} x^2 e^{-x^2} dx, \tag{17}$$

where $h(x) = x^2$ and $p(x) = Ae^{-x^2}$, where A is the normalization constant of the distribution. A script in python can be found in the file `les10.1.py` As shown in Figure 1 the estimate of I through Monte Carlo integration converges as the number of samples $\mathcal{N}$ increases. In addition to the estimate of I through the average of the values of $h(x_i)$ that were sampled, also the variance per point converges as the sample increases; thus the integral estimation is correct.
On the other hand, if we consider the function $h(x) = |x|^{-3/5}$, with $p(x) = Ae^{-x^2}$, we find that, despite the fact that the integral

$$I = A \int_{-\infty}^{+\infty} |x|^{-3/5} e^{-x^2} dx, \tag{18}$$

is finite, the variance, in which an integral of the type $\int_{-\infty}^{+\infty} |x|^{-6/5} e^{-x^2} dx$ appears, is not finite, as shown in the right panel of Figure 1. A script in python can be found in the file `les10.2.py`.

1.4.1. Direct sampling of a Gaussian distribution: Box-Muller transform. In order to extract random numbers according to a Gaussian distribution we can use the Box-Muller transform. Let ν and η be two random numbers extracted in the uniform interval $[0, 1]$. We can define two random numbers, g_1 and g_2, extracted according to a Gaussian distribution with zero mean value and unitary variance, as:

$$g_1 = \sqrt{-2\ln(\nu)}\cos(2\pi\eta) \qquad g_2 = \sqrt{-2\ln(\nu)}\sin(2\pi\eta)\,. \tag{19}$$

To obtain numbers extracted with a general Gaussian with mean value μ and variance σ^2 we can simply translate and rescale the transformed random numbers according to the relations

$$g_1' = \mu + \sigma g_1 \qquad g_2' = \mu + \sigma g_2. \tag{20}$$

A script in python to generate random numbers distributed according to a Gaussian distribution can be found in the file `boxMuller.py`. A simpler version is shown in the script `les09.4.py`.

1.5. Sampling a general distribution: the Metropolis-Hastings algorithm. The advantage of stochastic integration is that it can be applied to whatever complicated functions, also with explicit correlation between the integration variables, with the only condition that also the variance associated to the function $h(\mathbf{r})$ is finite.
Yet, for complicated distributions, is it not possible to automatically generate numbers according to $p(\mathbf{r})$.
For this reason, in order to sample the configurations the most efficient approach is to apply the general purpose sampling method (combining a Markov chain with an accept/reject procedure) which was first introduced my Metropolis *et al.*[3] and later generalized by Hastings[4].
The Metropolis-Hastings algorithm is able to generate $\mathcal{N}$ consecutive configurations through the following steps that describe a random walk:

1. Initialise a *walker* in a configuration $\mathbf{r}$;
2. Propose a trial move from $\mathbf{r}$ to a new position $\mathbf{r}'$ with transition probability $\mathcal{T}(\mathbf{r}' \leftarrow \mathbf{r})$, so that the probability that a *walker* initially in $\mathbf{r}$ afterwards transitions in an infinitesimal volume $d\mathbf{r}'$ of the configurational space is equal to $d\mathbf{r}'\mathcal{T}(\mathbf{r}' \leftarrow \mathbf{r})$. The simplest transition probability that is usually used is the one that corresponds to extracting with uniform probability a new move in an interval $[-\Delta, \Delta]$, *ie* $\mathcal{T}(\mathbf{r}' \leftarrow \mathbf{r}) = \mathcal{T}(\mathbf{r} \leftarrow \mathbf{r}') = \frac{1}{(2\Delta)^d}$ (where d is the dimension of the system). This corresponds on extracting a vector $\mathbf{z}$ of d random numbers in the interval $[0, 1]$ so that the new move is given by:
$$\mathbf{r}' = \mathbf{r} + (2\mathbf{z} - 1)\Delta. \tag{21}$$
3. Accepted the trial move with probability
$$\mathcal{A}(\mathbf{r}' \leftarrow \mathbf{r}) = \min\left\{\frac{p(\mathbf{r}')\mathcal{T}(\mathbf{r} \leftarrow \mathbf{r}')}{p(\mathbf{r})\mathcal{T}(\mathbf{r}' \leftarrow \mathbf{r})}, 1\right\}. \tag{22}$$
This corresponds to always accepting the move if $\mathcal{A}(\mathbf{r}' \leftarrow \mathbf{r}) \geq 1$, else, we extract a random number $\eta \in [0, 1]$: if $\eta \leq \mathcal{A}(\mathbf{r}' \leftarrow \mathbf{r})$ the move is accepted, otherwise it is refused.
4. If the trial move is accepted $\mathbf{r}'$ becomes the new position of the *walker* and a new transition is proposed from there, else a new transition is proposed from the previous position $\mathbf{r}$.

This procedure is repeated until $\mathcal{N}$ configurations (some of them repeated) are sampled.
An important property of this *random walk* procedure is that it satisfies the *detailed balance principle* after an equilibration period. To prove this and to show its implications, let's assume to have a distribution of independent walkers according to a general function $\mathcal{P}(\mathbf{r})$, so that the number of walkers in an infinitesimal volume of space is $d\mathbf{r}\mathcal{P}(\mathbf{r})$. Once the random walk of each walker is progressed for a sufficient amount of time we should reach an equilibrium in which the number of walkers transitioning between two volumes of space $d\mathbf{r}'$ and $d\mathbf{r}$ in one time step should be the same.
This equilibrium condition is expressed as:
$$d\mathbf{r}'\mathcal{A}(\mathbf{r}' \leftarrow \mathbf{r})\mathcal{T}(\mathbf{r}' \leftarrow \mathbf{r})\mathcal{P}(\mathbf{r})d\mathbf{r} = d\mathbf{r}\mathcal{A}(\mathbf{r} \leftarrow \mathbf{r}')\mathcal{T}(\mathbf{r} \leftarrow \mathbf{r}')\mathcal{P}(\mathbf{r}')d\mathbf{r}'$$
so that:
$$\frac{\mathcal{P}(\mathbf{r}')}{\mathcal{P}(\mathbf{r})} = \frac{\mathcal{A}(\mathbf{r}' \leftarrow \mathbf{r})\mathcal{T}(\mathbf{r}' \leftarrow \mathbf{r})}{\mathcal{A}(\mathbf{r} \leftarrow \mathbf{r}')\mathcal{T}(\mathbf{r} \leftarrow \mathbf{r}')}.$$
By substituting the acceptance probability defined in eq. 22 we have that
$$\frac{\mathcal{P}(\mathbf{r}')}{\mathcal{P}(\mathbf{r})} = \frac{p(\mathbf{r}')}{p(\mathbf{r})}$$
and this guarantees that the random walk, after an equilibration (or thermalization) period, is actually sampling the correct probability density $p(\mathbf{r})$ and so the **detailed balance principle** is satisfied.
One important comment that has to be made, is that the random walk does not need the probability density to be normalized since in the definition of $\mathcal{A}(\mathbf{r}' \leftarrow \mathbf{r})$ only the ratio between probability densities appears, cancelling out the normalization factor.
The main drawback of *random walks* is that the configurations that are generated, are correlated since the new configurations have *memory* of the space from which they have transitioned.

1.6. Correlation in Monte Carlo sampling. In Monte Carlo the stochastic sampling of the random variable $\mathbf{r}$ according to $p(\mathbf{r} \in \mathcal{V})$ cannot be done directly (because of the complexity of the probability distribution), but is done through a *random walk* defined through the Metropolis-Hastings algorithm.
During a random walk each new sample $\mathbf{r}'$ is extracted according to a chosen transition probability $T(\mathbf{r}' \leftarrow \mathbf{r})$ which has a memory of the position $\mathbf{r}$ from which the move is proposed. This 'memory' introduces an auto-correlation within the sampling process that diminishes as the distance between $\mathbf{r}$ and $\mathbf{r}'$ increases, *ie* as the random walk proceeds.
The auto-correlation in the random variable $\mathbf{r}$ causes the underestimation of the standard errors associated to the variable's functions $f(\mathbf{r})$.
Two consecutive values of the random variable's functions will be maximally correlated and as time passes, *ie* the process evolves, the auto-correlation disappears (Figure 2). The time in which the auto-correlation vanishes is called auto-correlation time τ_{corr}, and is defined as:

$$\tau_{corr} = 1 + 2\int_0^\infty \mathcal{R}(\tau)d\tau \tag{23}$$

where $\mathcal{R}(\tau)$ is the auto-correlation function. For a real function $f(\mathbf{r})$ sampled on a discreet time series of $\mathcal{N}$ elements, the normalized (because divided by the population variance estimated through the finite sample) the auto-correlation function is given by:

$$\mathcal{R}_f(\tau) = \frac{1}{s_f^2(\mathcal{N}-\tau)}\sum_{i=1}^{\mathcal{N}-\tau}\left(f(\mathbf{r}_i)-\bar{f}\right)\left(f(\mathbf{r}_{i+\tau})-\bar{f}\right) \tag{24}$$

where it is important that $\tau << \mathcal{N}$ in order to obtain significant statistics.
Once the correlation function is computed in the 'significant' interval, the auto-correlation time is given by

$$\tau_{corr} = 1 + 2\sum_{\tau=1}^{T}\mathcal{R}_f(\tau) \quad T << \mathcal{N}, \tag{25}$$

through which it is possible to obtain an estimation of the uncorrelated variance of a function $f(\mathbf{r})$ as

$$\bar{s}_f^2 \leftarrow s_f^2\tau_{corr}. \tag{26}$$

An example on how the autocorrelation is computed can be found in the python script `meanAndStdErr.py`.
In the following sections we will look into the problem of correlation when studying the 1D quantum Harmonic oscillator.

1.7. 'Reblocking' method for correcting correlation effects. A more straightforward approach for uncorrelating the variance is through the binning analysis, based on the reblocking transformation technique. Assume to have $\mathcal{N}$ correlated values of the function $f(\mathbf{r})$ from the sample of $\mathbf{r}$, *ie* $\{f(\mathbf{r}_1), f(\mathbf{r}_2), \ldots, f(\mathbf{r}_\mathcal{N})\}$: By dividing this series in $\mathcal{N}_\mathcal{B}$ consecutive blocks (or bins) of length $\mathcal{B} = \frac{\mathcal{N}}{\mathcal{N}_\mathcal{B}}$ we can define for each of them the average of the function as:

$$\mathcal{F}_b = \frac{1}{\mathcal{B}}\sum_{i=1}^{\mathcal{B}} f(\mathbf{r}_{(b-1)\mathcal{N}_\mathcal{B}+i}) \quad b \in [1:\mathcal{N}_\mathcal{B}] \tag{27}$$

thus obtaining a series of new variables $\{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_{\mathcal{N}_\mathcal{B}}\}$ which are now uncorrelated if the block size is chosen so that $\mathcal{B} > \tau_{corr}$.
The mean value of these variables will be exactly the same of the original series

$$\bar{f} = \frac{1}{\mathcal{N}_\mathcal{B}}\sum_{b=1}^{\mathcal{N}_\mathcal{B}}\mathcal{F}_b, \tag{28}$$

since by substitution we have that

$$\frac{1}{\mathcal{N}_\mathcal{B}}\sum_{b=1}^{\mathcal{N}_\mathcal{B}}\mathcal{F}_b = \frac{1}{\mathcal{B}\mathcal{N}_\mathcal{B}}\sum_{b=1}^{\mathcal{N}_\mathcal{B}}\sum_{i=1}^{\mathcal{B}} f(\mathbf{r}_{(b-1)\mathcal{N}_\mathcal{B}+i}) = \frac{1}{\mathcal{N}}\sum_{i=1}^{\mathcal{N}} f(\mathbf{r}_i).$$

The variance on the other hand will be computed on the reduced series as

$$s^2_{\mathcal{F}_b} = \frac{1}{\mathcal{N}_\mathcal{B} - 1} \sum_{b=1}^{\mathcal{N}_\mathcal{B}} \left(\mathcal{F}_b - \overline{f}\right)^2 . \tag{29}$$

A way to estimate the uncorrelated value of the variance over a finite sample is to study its variation as a function of the bin length: In fact, if $\mathcal{N} >> \mathcal{N}_\mathcal{B}$, the variance should reach a plateau value for $\mathcal{N}_\mathcal{B} \approx \tau_{corr}$. A function to apply the reblocking procedure to a time-series can be found in the python script `meanAndStdErr.py`, we will show its effect on decorrelating the variables when studying the 1D harmonic oscillator.

1.8. Resampling methods: the jackknife. Once the stochastic variables are uncorrelated through reblocking another problem has to be addressed especially when dealing with non-linear functions of random variables, such as ratios or products. First, the computation of these quantities on finite samples might lead to large errors due to the intrinsic bias; Second, in order to compute the variances of the new functions one has to explicitly write the propagation of their errors, often computing also the cross correlations between the variables.
Another way to estimate the statistical errors and to correct the bias of complex functions of the observables, without relying on assumptions regarding their distributions, is to apply resampling methods such as the bootstrap and in particular the simple jackknife.
To understand the basic idea of the jackknife resampling connecting the jackknife quantities to the values previously defined. Let us consider the newly defined series $\{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_{\mathcal{N}_\mathcal{B}}\}$ of uncorrelated measures. We define the $\mathcal{N}_\mathcal{B}$ jackknife resampling averages as the mean values of all the elements minus one

$$\mathcal{F}_b^{jk} = \frac{1}{\mathcal{N}_\mathcal{B} - 1} \sum_{i \neq b}^{\mathcal{N}_\mathcal{B}} \mathcal{F}_i = \frac{\mathcal{N}_\mathcal{B}\overline{f} - \mathcal{F}_b}{\mathcal{N}_\mathcal{B} - 1} \tag{30}$$

giving the series $\left\{\mathcal{F}_1^{jk}, \mathcal{F}_2^{jk}, \ldots, \mathcal{F}_{\mathcal{N}_\mathcal{B}}^{jk}\right\}$. We now define the jackknife mean value of the local energies as

$$\overline{\mathcal{F}}^{jk} = \frac{1}{\mathcal{N}_\mathcal{B}} \sum_{b=1}^{\mathcal{N}_\mathcal{B}} \mathcal{F}_b^{jk} \tag{31}$$

that for this linear function is equal to the expectation value of the original series, in fact it can be easily shown by substitution that

$$\overline{\mathcal{F}}^{jk} = \frac{1}{\mathcal{N}_\mathcal{B}} \sum_{b=1}^{\mathcal{N}_\mathcal{B}} \frac{\mathcal{N}_\mathcal{B}\overline{f} - \mathcal{F}_b}{\mathcal{N}_\mathcal{B} - 1} = \frac{\mathcal{N}_\mathcal{B}\overline{f}}{\mathcal{N}_\mathcal{B} - 1} - \frac{1}{(\mathcal{N}_\mathcal{B} - 1)\mathcal{N}_\mathcal{B}} \sum_{b=1}^{\mathcal{N}_\mathcal{B}} \mathcal{F}_b = \frac{\mathcal{N}_\mathcal{B}\overline{f}}{\mathcal{N}_\mathcal{B} - 1} - \frac{\overline{f}}{\mathcal{N}_\mathcal{B} - 1} = \overline{f}.$$

For the variance (and the covariance) we have that since each element of the series is constructed by all the elements of the original sample minus one, the resampled elements are now overcorrelated. This overcorrelated jackknife standard deviation

$$s^2_{\mathcal{F}_b^{jk}} = \frac{1}{\mathcal{N}_\mathcal{B} - 1} \sum_{b=1}^{\mathcal{N}_\mathcal{B}} \left(\mathcal{F}_b^{jk} - \overline{\mathcal{F}}^{jk}\right)^2 \tag{32}$$

will be connected to the standard deviation of the initial series by a simple factor, in fact we can show by substitution that

$$s^2_{\mathcal{F}_b^{jk}} = \frac{1}{\mathcal{N}_\mathcal{B} - 1} \sum_{b=1}^{\mathcal{N}_\mathcal{B}} \left(\mathcal{F}_b^{jk} - \overline{f}\right)^2 = \frac{1}{\mathcal{N}_\mathcal{B} - 1} \sum_{b=1}^{\mathcal{N}_\mathcal{B}} \left(\frac{\mathcal{N}_\mathcal{B}\overline{f} - \mathcal{F}_b}{\mathcal{N}_\mathcal{B} - 1} - \overline{f}\right)^2 =$$

$$= \frac{1}{\mathcal{N}_\mathcal{B} - 1} \sum_{b=1}^{\mathcal{N}_\mathcal{B}} \left(\frac{\overline{f} - \mathcal{F}_b}{\mathcal{N}_\mathcal{B} - 1}\right)^2 = \frac{s^2_{\mathcal{F}_b}}{(\mathcal{N}_\mathcal{B} - 1)^2}$$

from which we are left with the equation

$$s^2_{\mathcal{F}_b} = s^2_{\mathcal{F}_b^{jk}} (\mathcal{N}_\mathcal{B} - 1)^2, \tag{33}$$

that leaves us with

$$s^2_{\mathcal{F}_b} = (\mathcal{N}_\mathcal{B} - 1) \sum_{b=1}^{\mathcal{N}_\mathcal{B}} \left(\mathcal{F}_b^{jk} - \overline{\mathcal{F}}^{jk} \right)^2 . \tag{34}$$

Thus, for a linear function of an uncorrelated random variable the jackknife resampling gives us, as expected, exactly the mean value and standard deviation as the standard procedure.
On the other hand resampling is important when evaluating non-linear functions of the expectation values of different random variables on a finite population sample. An example of this can be found in the evaluation of the atomic forces in QMC or in the evaluation of the generalized forces that act on the out-of-equilibrium parameters during the optimization procedure.

2. Variational Monte Carlo

Variational Monte Carlo[1] is the simplest of the quantum Monte Carlo methods that will be considered, and is based on the application of Monte Carlo integration to the calculation of the energy functional defined in the variational method. As already mentioned, compared to traditional quadrature techniques, Monte Carlo methods allow a more efficient estimation of multidimensional integrals, that appear for example in the evaluation of the mean values of physical observables of many-particle systems.
In the structure of matter we are interested in studying quantum systems of M bodies, described by the component vector $\bar{\mathbf{r}} = (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_M)$ of $3M$ independent coordinates, described by Hamiltonians of the type

$$\hat{H} = -\frac{\hbar^2}{2m} \sum_{i=1}^{M} \vec{\nabla}_i^2 + V(\bar{\mathbf{r}})$$

where $V(\bar{\mathbf{r}})$ is a potential energy that generally contains non-separable interaction terms, making the Schrödinger stationary equation analytically unsolvable.
In the event that a perturbative treatment of the problem is not sufficient or not even possible, one can resort to the variational method[5], which allows to make an estimate, for example, of the eigenvalue of the ground state energy, provided that a trial wave function $\psi_T(\bar{\mathbf{r}})$ is known, which is a good approximation of the exact fundamental eigenstate. In this case, in fact, we can define the energy functional associated with $\psi_T(\bar{\mathbf{r}})$ as

$$E\left[\psi_T(\bar{\mathbf{r}})\right] = \frac{\langle \psi_T | \hat{H} | \psi_T \rangle}{\langle \psi_T | \psi_T \rangle} = \frac{\int_V \psi_T^*(\bar{\mathbf{r}}) \hat{H} \psi_T(\bar{\mathbf{r}}) d\bar{\mathbf{r}}}{\int_V |\psi_T(\bar{\mathbf{r}})|^2 d\bar{\mathbf{r}}} \tag{35}$$

provided that some conditions are met. First of all, $\psi_T(\bar{\mathbf{r}})$ must be continuous and differentiable wherever the potential $V(\bar{\mathbf{r}})$ is finite; it must then be normalizable, and furthermore, it must be such that the integral $\int_V \psi_T^*(\bar{\mathbf{r}}) \hat{H} \psi_T(\bar{\mathbf{r}}) d\bar{\mathbf{r}}$ is finite. However, these conditions are not sufficient to perform a numerical estimate of the functional through Monte Carlo. In fact, for this purpose, as discussed in the previous sections, also the variance must be correctly determined and the trial wave function must be such that $\int_V \psi_T^*(\bar{\mathbf{r}}) \hat{H}^2 \psi_T(\bar{\mathbf{r}}) d\bar{\mathbf{r}}$ is also finite.
The energy functional (eq. 35) has the property of being greater than or equal to the energy associated with the exact ground state $\phi_0(\bar{\mathbf{r}})$. This is easy to prove since, being the set of the exact eigenstates of the Hamiltonian $\{\phi_i(\bar{\mathbf{r}}), i = 1, 2...\}$ a complete orthonormal basis in the Hilbert space of the M particles, one can always write a trial wave function as their linear combination $\psi_T(\bar{\mathbf{r}}) = \sum_i a_i \phi_i(\bar{\mathbf{r}})$, obtaining the expression of the functional

$$E\left[\psi_T\right] = \frac{\sum_i |a_i|^2 E_i}{\sum_i |a_i|^2}. \tag{36}$$

By subtracting from this equation the exact ground state's eigenvalue we get

$$E\left[\psi_T\right] - E_0 = \frac{\sum_i |a_i|^2 (E_i - E_0)}{\sum_i |a_i|^2} \tag{37}$$

where the right term is always positive so that

$$E\left[\psi_T\right] - E_0 \geq 0 \qquad E\left[\psi_T\right] \geq E_0. \tag{38}$$

Of course, if the trial wave function was exactly the ground state (or any eigenstate of the system), this relationship would turn into the equality $E[\phi_i] = E_i$, but the knowledge of the exact ground state is almost never possible for systems of M bodies having non-separable Hamiltonians, and in general we only have an approximate form of the eigenstate, expressed, perhaps, as a function of some parameters.
From these considerations we understand the importance of an optimal choice of $\psi_T(\bar{\mathbf{r}})$, since a good estimation of the properties of the system depends on it (in the example, the ground state's energy). However, this relates to the physical intuition and to the set of a priori properties of the quantum state in question, and is true for any variational approach, whether the integrals are performed analytically or with numerical techniques.
In applying Monte Carlo integration to the variational method, the Energy functional 35 is rewritten by dividing and multiplying the integrand in the numerator by the trial function $\psi_T(\bar{\mathbf{r}})$ as

$$E[\psi_T] = \int_V \frac{|\psi_T(\bar{\mathbf{r}})|^2}{\int_V |\psi_T(\bar{\mathbf{r}})|^2 d\bar{\mathbf{r}}} \frac{1}{\psi_T(\bar{\mathbf{r}})} \hat{H}\psi_T(\bar{\mathbf{r}}) d\bar{\mathbf{r}} = \int_V E_{loc}(\bar{\mathbf{r}}) p(\bar{\mathbf{r}}) d\bar{\mathbf{r}} \tag{39}$$

where

$$E_{loc}(\bar{\mathbf{r}}) = \frac{1}{\psi_T(\bar{\mathbf{r}})} \hat{H}\psi_T(\bar{\mathbf{r}}) \tag{40}$$

is the *local energy* of a configuration $\bar{\mathbf{r}}$, and

$$p(\bar{\mathbf{r}}) = \frac{|\psi_T(\bar{\mathbf{r}})|^2}{\int_V |\psi_T(\bar{\mathbf{r}})|^2 d(\bar{\mathbf{r}})} \tag{41}$$

is the probability density associated with that configuration $\bar{\mathbf{r}}$ (the probability of finding the system in that particular configuration).
Once the functional has been rewritten in terms of the local energy $E_{loc}(\bar{\mathbf{r}})$, one proceeds to calculate the energy associated with the trial function through a random walk performed with the Metropolis-Hastings algorithm, extracting with probability $p(\bar{\mathbf{r}})$ the configurations of the sample, and averaging the corresponding values of local energy that are obtained

$$E[\psi_T] \approx \langle E_{loc} \rangle_N \pm \sqrt{\frac{\langle E_{loc}^2 \rangle_N - \langle E_{loc} \rangle_N^2}{N}}. \tag{42}$$

Particular attention should be paid to the variance of the local energy distribution $\langle E_{loc}^2 \rangle_N - \langle E_{loc} \rangle_N^2$, which can be used as a measure of the quality of the trial wave function. This derives from the fact that, as the trial wave function $\psi_T(\bar{\mathbf{r}})$ approaches an eigenstate of the system, the local energy fluctuations reduce until if becomes independent of the position $\bar{\mathbf{r}}$ becoming a constant equal to the eigenvalue[1].

3. 1D quantum Harmonic oscillator

In order to explore the usage of the Monte Carlo methods described above, we have chosen to consider a simple quantum system, whose solutions, analytically known, can be compared with those obtained numerically. A simple and very important system in all physics is the harmonic oscillator[5], which in the one-dimensional case is described by the Schrödinger equation

$$\hat{H}\phi_n(x) = E_n\phi_n(x) \qquad \hat{H} = -\frac{\hbar}{2m}\frac{d^2}{dx^2} + \frac{1}{2}m\omega^2 x^2$$

of which the eigenvalues and eigenstates are known to be defined as

$$\phi_n(\xi) = \frac{1}{\sqrt{2^n n!}} \left(\frac{m\omega}{\pi\hbar}\right)^{1/4} e^{-\xi^2/2} H_n(\xi) \qquad E_n = \hbar\omega\left(n + \frac{1}{2}\right) \qquad n = 0, 1, 2, 3, \ldots$$

being $\xi = \sqrt{\frac{m\omega}{\hbar}} x$ the rescaled spatial variable and being the functions $H_n(\xi)$ the Hermite polynomials defined as

$$H_n(\xi) = (-1)^n e^{-\xi^2} \left(\frac{d}{d\xi}\right)^n e^{\xi^2}.$$

3.1. Variational Monte Carlo for the ground state. To apply the variational Monte Carlo method[1] to the ground state of the harmonic oscillator, one must first choose a trial function $\psi_T(x;\Delta\omega)$, dependent on one or more variational parameters. For purely illustrative purposes, it is convenient to assume a functional form that has in itself the expression of the exact eigenfunction, and depends on a single parameter, such as

$$\psi_T(x;\Delta\omega) = e^{-\frac{m\omega'}{2\hbar}x^2} \qquad \omega' = \omega + \Delta\omega$$

that for $\omega' = \omega$ (that is $\Delta\omega = 0$), reduces to the exact state

$$\phi_0(x) = \left(\frac{m\omega}{\hbar\pi}\right)^{1/4} e^{-\frac{m\omega}{2\hbar}x^2},$$

to which it is associated the eigenvalue $E_0 = \frac{1}{2}\hbar\omega$: clearly, for $\Delta\omega \neq 0$ $\psi_T(x;\Delta\omega)$ will not be a solution of Schrödinger's equation.
As previously discussed, the fact that we have chosen the Metropolis algorithm to sample points in space allows us to assume a non-normalized trial function. In this simple case, however, the average value of $\hat{H}$ and the normalization of $\psi_T(x;\Delta\omega)$ are known, and this allows us to obtain the analytical expression, as a function of the parameter $\Delta\omega$, of the Energy functional

$$E[\psi_T] = \frac{\int_V \psi_T^* \hat{H}\psi_T dV}{\int_V |\psi_T|^2 dV} = \frac{1}{2}\hbar\omega' + \frac{\hbar}{2\omega'}\left(\omega^2 - \omega'^2\right) \tag{43}$$

and of its associated variance

$$\sigma_E^2 = \frac{\int_V \psi_T^* \hat{H}^2\psi_T dV}{\int_V |\psi_T|^2 dV} - \left(\frac{\int_V \psi_T^* \hat{H}\psi_T dV}{\int_V |\psi_T|^2 dV}\right)^2 = \frac{\hbar^2\left(\omega^2 - \omega'^2\right)^2}{8\omega'^2}. \tag{44}$$

These formulas provide the exact reference which we will compare with the Monte Carlo numerical results.
To perform the simulation, the first step as we have seen, is to rewrite the energy functional, separating the local energy $E_{loc}(x)$ from the probability distribution $|\psi_T(x;\Delta\omega)|^2$:

$$E[\psi_T] = \int_V |\psi_T(x;\Delta\omega)|^2 \frac{1}{\psi_T(x;\Delta\omega)}\hat{H}\psi_T(x;\Delta\omega)dx = \int_V E_{loc}(x)\,|\psi_T(x;\Delta\omega)|^2\,dx, \tag{45}$$

computing the expression of the local energy as a function of the configuration:

$$E_{loc}(x) = \frac{1}{\psi_T(x;\Delta\omega)}\hat{H}\psi_T(x;\Delta\omega), \tag{46}$$

which clearly will depend on the parameter $\Delta\omega$

$$E_{loc}(x) = \underbrace{\frac{1}{2}\hbar\omega' - \frac{1}{2}m\omega'^2x^2}_{T_{loc}(x)} + \underbrace{\frac{1}{2}m\omega^2x^2}_{V_{loc}(x)} = \frac{1}{2}\hbar\omega' + \frac{1}{2}m\left(\omega^2 - \omega'^2\right)x^2, \tag{47}$$

and will be equal to the sum of two terms, representing respectively the local kinetic energy $T_{loc}(x)$ and the local potential energy $V_{loc}(x)$. These two functions are useful for studying the correlation and equilibration of the system, since, unlike the local energy, they have non-null variance also for $\Delta\omega = 0$.
In the simulation it is usually convenient to set to one all the constants ($\hbar = m = \omega = 1$) that appear in the Hamiltonian, so that the difference $\Delta\omega$ is measured in units of ω.
The results thus obtained for estimating the functional and the variance per point as a function of the parameter are shown in figure 2. In the parameter space there is always an absolute minimum of the energy functional, and this represents the best possible estimate of the system's eigenstate for the chosen trial function. To this minimum will also correspond the minimum of the variance, which will have the important property of canceling itself out if the trial function contains the expression of an exact eigenfunction of the system. Since for complex systems the spectrum of the eigenvalues is not known and it is not possible to evaluate how close the functional estimate is to the exact result, it is appropriate to use the variance as a measure of the quality of the $\psi_T(\bar{\mathbf{r}})$. This result can be obtained by directly sampling the Gaussian distribution related to the trial wave function, this can be found in the python script file `les11.1.py`. The sampling obtained through the Metropolis-Hastings algorithm can be found in the python script `les11.2.py`.

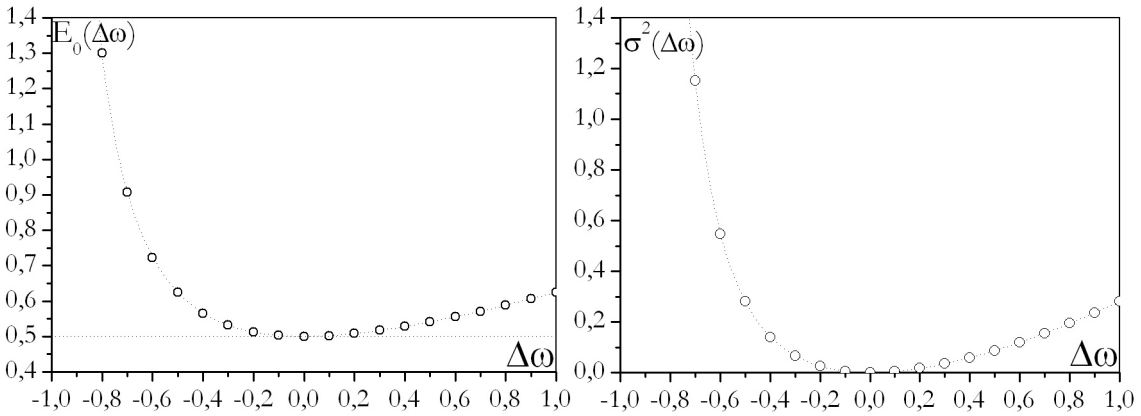

Figure 2: The two pictures show repectively the VMC estimations of the energy functional and its variance as a function of the trial wave function's parameter $\Delta\omega$. The dotted lines represent the analytical expressions computed in the test. The minimum value of the energy functional corresponds to $\Delta\omega = 0$ when the energy corresponds to the exact eigenvalue of the ground state E_0 and the trial wave function corresponds to the exact ground state eigenfunction $\phi_0(x)$. For $\Delta\omega = 0$ we can see that the variance is null since the local energy becomes a constant function of the configurations.

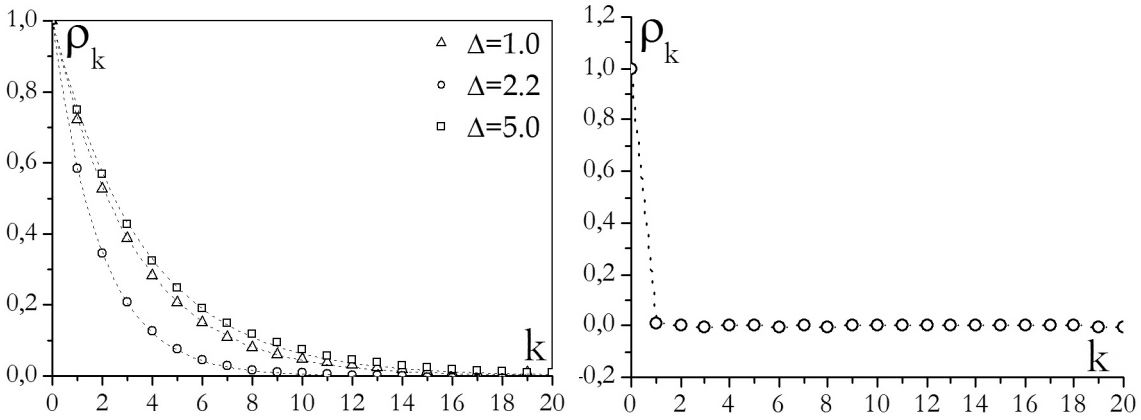

Figure 3: On the left we can see the correlation function for different values of the transition step Δ; on the right we can see the correlation function for the decorrelated values obtained through reblocking.

One of the problems already highlighted in the description of the Metropolis algorithm is that of the optimization of the simulation through the choice of the initial position x_0 of the sampling, and of the maximum size of the step Δ, in order to respectively reduce the equilibration time and the correlation of the data in the random path.

By setting $x_0 = 0$ as initial position and by thermalizing the sampling, it is possible to study the autocorrelation time as a function of the Δ step in the transition probability through the the correlation function described in eq. 24. This can be found in the first panel of figure 3. In the right panel we can see how after reblocking the correlation time drops to 1. The correlation and reblocking procedure is studied in the python script `les11.3.py`.

References

(1) Kalos, M. H.; Whitlock, P. A. *Monte Carlo Methods*; John Wiley & Sons, Ltd, 2009; Chapter 3, pp 35–76.

(2) Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3rd ed.; Cambridge University Press: USA, 2007.

(3) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.* **1953**, *21*, 1087–1092.

(4) Hastings, W. K. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **1970**, *57*, 97–109.
(5) Sakurai, J. J.; Napolitano, J. *Modern Quantum Mechanics*, 2nd ed.; Cambridge University Press, 2017.