# Advanced Methods in Computational Sciences: Monte Carlo Simulations

A Lecture Series by

Prof. Dr. habil. Martin O. Steinhauser

Japan, October 2024

# LECTURE 1

# THE MONTE CARLO METHOD - AN INTRODUCTION

## Summary.

This lecture series provides an introduction into the basic principles of the Monte Carlo Method.

## Learning targets

✓ Understand the basic principles of the Monte Carlo simulation technique.

## 1.1 What is the Monte Carlo Method?

First of all, Monte Carlo is not *one single* method, but it refers to a whole *class of numerical approximative methods* which can be characterized by their reliance on repeated random sampling and averaging to obtain results. One of their major advantages compared to other numerical approaches is their systematic improvement with the number of samples $N$, as the error $\Delta\epsilon$ decreases as follows:

$$\Delta\epsilon \propto \frac{1}{\sqrt{N}}. \tag{1.1}$$

A simple but good example of this is the computation of $\pi$ which we will consider a little later in this text.

Monte Carlo is a whole class of methods which are based on sampling of random numbers.

## 1.2 Typical Applications of the Monte Carlo method

Monte Carlo methods have a broad spectrum of applications, including the following:

- **Physics:** They are used in many areas in physics, some applications include statistical physics (e.g. Monte Carlo molecular modeling) or in Quantum Chromodynamics. In the Large Hadron Collider (LHC) at CERN, Monte Carlo methods were used to simulate signals of Higgs particles, and they are used for designing detectors (and to help understand as well as predict their behavior).

- **Economics** Monte Carlo (MC) models have also found their place in economics where they have been used for e.g. financial derivatives.

- **Design:** Monte Carlo methods have also penetrated areas that might seem surprising at first sight. For example, they help in solving coupled integro-differential equations of radiation fields and energy transport, which is essential for global illumination in photorealistic images of 3D models.

Let us now consider the example of the calculation of $\pi$ that we can directly implement and which will familiarize us with this method.

### 1.2.1 Computation of $\pi$

A very easy and illustrative example of the method of random sampling and averaging is the computation of the number $\pi$.

The basic idea for doing this is rather simple: We consider the unit area $x \in [0,1]$ and $y \in [0,1]$ (dotted lines in Fig. 1.1) and compare the area within the quarter circle, $P(x,y)$, to the area of the unit square (see Fig. 1.1). This will give $\pi/4$.

This result is mathematically exact and can be expressed as an integral in the following way:

$$\pi = \int_0^1 \sqrt{1-x^2}\, dx. \qquad (1.2)$$

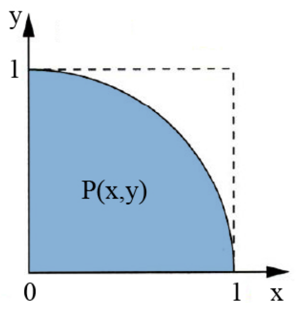


**Figure 1.1.** Illustration of the areas considered in the computation of $\pi$.

Another way to compute $\pi$ goes as follows: We consider $N$ random points in the unit square that are characterized by their $x$ and $y$ coordinates $x_i$ and $y_i$. Then, the number of points $N_c$ lying within the quarter circle (i.e. those points fulfilling the relation $x^2 + y^2 \leq 1$) is divided by the total number $N$ of points and this fraction gives us an approximate value of $\pi$:

$$\pi(N) = 4\frac{N_c(N)}{N}. \tag{1.3}$$

Of course, the more points we consider, the better the approximation will become. In fact, the error of the approximation is $\Delta\epsilon = \pi(N) - \pi \propto \frac{1}{\sqrt{N}}$, and thus really does decrease with the number $N$ of points. This is easy to understand. Imagine you chose just two points, one of which is lying inside the quarter circle, the other one outside. This will give you $\pi(2) = 2$, a rather crude approximation to 3.14159265... but if you pick 10 points, they will approximate the real value of $\pi$ much better. For $N = 10,000$ for instance you may find that $N_c = 7.854$, giving $N(10,000) = 3.1416$. It is, however, not obvious from the start that the error should decrease $\propto \frac{1}{\sqrt{N}}$, which is why we will come back to this a little later in the text.

## 1.3 Computation of Integrals

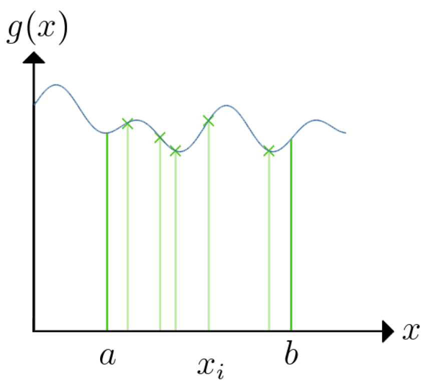
Another well-known application of Monte Carlo is the computation of integrals, *particularly higher dimensional integrals* (we shall see later on that Monte Carlo is in fact the most efficient way to compute higher dimensional integrals). Let us consider the integral of a function $g(x)$ in an interval given by $[a, b]$.

Monte Carlo is the most efficient method to calculate high-dimensional integrals.

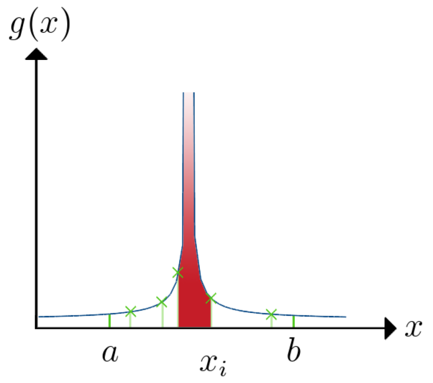
We may approximate this integral by choosing $N$ points $x_i$ on the $x$-axis with their corresponding values $g(x_i)$, summing and averaging over these sampled results and multiply-ing the resulting espression with the length of the intervsal:

$$\int_a^b g(x)\,dx \approx (b-a)\left[\frac{1}{N}\sum_{i=1}^{N} g(x_i)\right]. \tag{1.4}$$

We now need to say a word or two about the nature of these randomly chosen points $x_i$ on the $x$-axis. If we choose them completely at random, the process is called **simple sampling**, which works very well if $g(x)$ is a smooth function as in Fig. 1.2 (a).



**(a)** Simple sampling for a smooth function.



**(b)** Simple sampling when applied to a strongly fluctuating function, here with a singularity.

**Figure 1.2.** Illustration of simple sampling.

But what if we cannot make the assumption that $g(x)$ is smooth? Let us consider a less "cooperative" function, for instance one featuring a singularity at a certain value as in Fig. 1.2 (b). Due to the limited number of points that we usually choose around the singularity (in simple sampling), we are not able to really capture the function's behavior. The approximation of the integral by a sum would be a very rough approximation and most of the sampled points would be "wasted", because they are in regions where the function doesn't change much.

Thus, we need more precision which goes beyond the possibilities of simple sampling. This additional precision is provided by a second function, a *distribution function* $p(x)$ which makes our condition for successful sampling less strict: we demand that only $\frac{g(x)}{p(x)}$ needs to be smooth. The sampling points are now distributed according to to $p(x)$ and we obtain for the integral in Eq. (1.4):

$$\int_a^b g(x)\,dx = \int_a^b \frac{g(x)}{p(x)} p(x)\,bx \approx (b-a)\left[\frac{1}{N}\sum_{i=1}^{N}\frac{g(x_i)}{p(x_i)}\right]. \tag{1.5}$$

We have changed our way of sampling by using the distribution function $p(x)$, thus reducing the requirement on $g(x)$. This manifests itself in the sum in Eq. (1.5) above. One could state that $p(x)$ helps us to pick our sampling points according to their importance, i.e. we select more points close to a singularity or close to those regions of a function where its derivative is large – consequently, this kind of sampling is called **importance sampling**.

### 1.3.1 Integration Errors

The introduced approximations of Eq. (1.5) and (1.4) are not analytical solutions. Therefore, we need to know what error one has to expect when using such an approximation.

#### 1.3.1.1 Error in Conventional Approximation Methods

Let us first consider the error in conventional methods for approximating the values of a function. We are going to use the in the following on the *trapezium rule.* Consider the Taylor Series expansion integrated from $x_0$ to $x_0 + \Delta x$:

$$\begin{aligned}\int_{x_0}^{x_0+\Delta x} f(x)\,dx &= f(x_0)\Delta x + \frac{1}{2}f'(x_0)\Delta x^2 + \frac{1}{2}f''(x_0)\Delta x^3 + \cdots \qquad (1.6)\\ &= \left[\frac{1}{2}f(x_0) + \frac{1}{2}\left(f(x_0) + f'(x_0)\Delta x + \frac{1}{2}f''(x_0)\Delta x + \cdots\right) + \cdots\right]\Delta x\\ &= \frac{1}{2}\left(f(x_0) + f(x_0+\Delta x)\right)\Delta x + O(\Delta x^3).\end{aligned}$$

This approximation, represented by $\frac{1}{2}\left[f(x_0) + f(x_0+\Delta x)\right]\Delta x$, is called the *trapezium rule* based on its geometric interpretation, see Fig. 1.3. We can now see that the error is $\propto (\Delta x)^3$. Suppose, we take $x$ to be one third times its previous value, then the error will decrease by a factor of 27. At the same time the size of the domain is also divided by this factor. The net factor for the error is thus only 9 (i.e. $(\Delta x)^2$) and not 27 (i.e. $(\Delta x)^3$) as originally conjectured. We now subdivide the interval $[x_0, x_1]$ into $N$ subintervals of size $\Delta x = \frac{x_1 - x_0}{N}$. The Compound trapezium rule approximation to

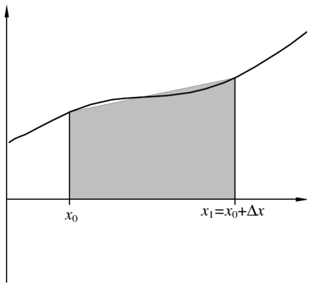


**Figure 1.3.** Graphical representation of the trapezium rule.

the integral of Eq. (1.6) is therefore

$$\begin{aligned}\int_{x_0}^{x_1} f(x)\,dx &\approx \frac{\Delta x}{2}\sum_{j=0}^{N-1} f(x_0+j\Delta x)+f(x_0+(j+1)\Delta x) \\ &= \frac{\Delta x}{2}\left[f(x_0)+2f(x_0+\Delta x)+2f(x_0+2\Delta x)+\right. \\ &\quad \left.+2\,f(x_0+(N-1)\Delta x)+f(x_1)\right].\end{aligned} \tag{1.7}$$

We thus see that while the error for each step is $O((\Delta x)^3)$, the *cumulative* error is $N$ times this or $O((\Delta x)^2) \propto O(N^{-2})$.

A generalization to 2 dimensions, where a function $f : X \to Y$ takes an argument $x$ from the *domain* $X$ and assigns it a value $y = f(x)$ in the *range* $Y$ of the function $f$, is rather straightforward. Instead of each interval contributing an error of $(\Delta x)^3$ we now have a two dimensional domain and instead of the error being $\Delta\epsilon \propto N(\Delta x)^3 \propto (\Delta x)^2$ (since $N \propto \frac{1}{\Delta x}$), we now have for each segment an error $\Delta\epsilon \propto (\Delta x)^4$. The cumulative error is then $\Delta\epsilon \propto N(\Delta x)^4 \propto (\Delta x)^2$.

We can now generalize to $d$ dimensions (i.e. the dimension of the domain is $d$-dimensional). The "intervals" are then $d$-dimensional, and the error for each segment is $(\Delta x)^{d+2}$, with the sum over all "subintervals" giving us $N(\Delta x)^{d+2}$, but $N \propto \frac{1}{(\Delta x)^d}$, so that the error is $\Delta\epsilon \propto N(\Delta x)^{d+2} \propto (\Delta x)^2$. The error is thus *independent* of the dimension.

We can conclude that the error of conventional methods is of order $O((\Delta x)^2)$, and given that the time $T \propto N \propto \frac{1}{(\Delta x)^d}$, we see that $\Delta x \propto T^{-1/d}$ and the error is $\Delta\epsilon \propto (\Delta x)^2 \propto T^{-2/d}$.

### 1.3.1.2 Error of the Monte Carlo Method

Let us consider the simple case of a one dimensional function of one variable, $g : [a, b] \to \mathbb{R}$. If we pick $N$ equidistant points in the interval $[a, b]$ we have a distance of $h = \frac{b-a}{N}$ between each of these points. The estimate for the integral in this case is

$$I = \int_a^b g(x)\,dx \approx \frac{b-a}{N}\sum_{i=1}^{N} g(x_i) = (b-a)\langle g\rangle = Q, \tag{1.8}$$

where $\langle g\rangle$ stands for the sample mean of the integrand. The variance var$(g)$ of the function can be estimated as

$$\mathrm{var}(g) = \sigma^2 = \frac{1}{N-1}\sum_{i=1} N\left(g(x_i) - \langle g\rangle\right)^2, \tag{1.9}$$

where the denominator is $N - -1$ as we want to obtain the unbiased estimate of the variance. Using the *central limit theorem*, the variance of the estimate of the integral can be estimated as

$$\mathrm{var}(Q) = (b-a)^2 \frac{\mathrm{var}(g)}{N} = (b-a)^2 \frac{\sigma^2}{N}, \tag{1.10}$$

which for large $N$ decreases like $\frac{1}{N}$. Thus, the error estimate is

$$\Delta Q \approx \sqrt{\mathrm{var}(Q)} = (b-a)\frac{\sigma^2}{\sqrt{N}}. \tag{1.11}$$

We can now generalize this to *multidimensional* integrals as follows:

$$I = \int_{a_1}^{b_1} dx_1 \int_{a_2}^{b_2} dx_2 \cdots \int_{a_n}^{b_n} dx_n f(x_1, x_2, ..., x_n). \tag{1.12}$$

Hence, the integral of Eq. (1.4) becomes a *hypercube* with volume $V$ as integration volume, where $V$ is given by:

$$V = \{x : a_1 \leq x_1 \leq b_1, ..., a_n \leq x_n \leq b_n\}. \tag{1.13}$$

We now use the volume $V$ instead of the interval $[a, b]$ in Eq. (1.10):

$$\mathrm{var}(Q) = V^2 \frac{\mathrm{var}(g)}{N} = V^2 \frac{\sigma^2}{N}, \tag{1.14}$$

and the error estimate of Eq. (1.11) is:

$$\Delta Q \approx \sqrt{\mathrm{var}(Q)} = V \frac{\sigma}{\sqrt{N}}. \tag{1.15}$$

Thus, the proportionality of the error to $1/\sqrt{N}$ still remains.

#### 1.3.1.3 Critical Dimension

We have seen in the previous section that in conventional methods, the error of computing integrals goes like $T^{-\frac{2}{d}}$ and thus depends on the dimension, while the approximation error in Monte Carlo methods is independent of the dimension. There is a crucial point – a *critical dimension* $d_c$ – at which Monte Carlo methods become more efficient than standard approximations for integrals, namely when

$$T^{-\frac{2}{d_c}} = \frac{1}{\sqrt{T}} \Rightarrow d_c = 4. \tag{1.16}$$

We can thus conclude that for $d > 4$, Monte Carlo becomes more efficient and is therefore used in areas where higher dimensional integrals with $d > 4$ are commonplace.

## 1.4 Higher Dimensional Integrals

Let us now consider an example of higher dimensional integration: Consider $N$ hard spheres of radius $R$ in a 3D box of volume $V$. The spheres are characterized by their position vectors $\vec{x}_i = (x_i, y_i, z_i)$ pointing to their centers with $1 \leq i \leq N$. We denote the distance between two spheres $i$ and $j$ as

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}, \tag{1.17}$$

where we have to take into account that the spheres are hard, i.e. cannot overlap.

Thus the minimal distance between two neighboring spheres is the distance when the spheres are in contact, which is equal to $2R$. This translates to the following condition on $r_{ij}$:

$$r_{ij} > 2R. \tag{1.18}$$

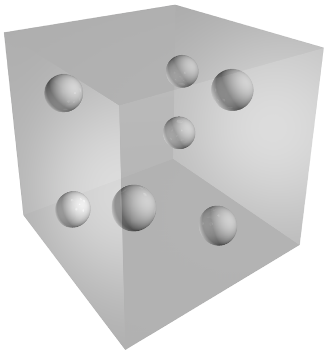


**Figure 1.4.** Illustration of a cube filled with spheres of radius $R$.

So far this does not seem to have too much to do with integration, we are simply placing some spheres in a volume V and measuring some distance $r_{ij}$. Let us assume that we are interested in the average distance between the centers of spheres in the box then we need to consider the following integral to reach an analytical result:

$$\langle r_{ij} \rangle = \frac{1}{Z} \int \frac{2}{N(N-1)} \sum_{i<j} r_{ij} d^3r_1 \cdots d^3r_N, \tag{1.19}$$

where

$$Z = \int d^3r_1 \cdots d^3r_N. \tag{1.20}$$

Let us stay with Eq. (1.19) for a moment and have a look at the different factors. The first factor, $1/Z$, is a normalization factor, called *partition function*. The following factor in zhe integral is of combinatory origin, in fact it stems from random drawing from a finite population without replacement. We start out with $N$ choices (the whole population) and pick one sphere at random. There are only $(N-1)$ choices left for the second one (as it cannot be the first one again). This gives a total number of $N \cdot (N-1)$ possible combinations for picking two spheres. As they are indistinguishable, there is an additional factor of $1/2$ and we thus divide by this whole combinatory expression. Note that we needed to correct with a factor of 1/2 if the sum was over $i \neq j$ (since, in that case, we would have counted each volume twice), which however is avoided by simply summing over $i < j$.

The Monte Carlo approach to this formula is relatively simple:

1. Choose a particle position (i.e. the center of the new sphere).
2. Make sure that the new sphere does not overlap with any pre-existing spheres (see condition on $r_{ij}$). If it does overlap, reject the position and try again.
3. Once all the spheres have been placed, calculate the distances $r_{ij}$.

We then use these distances $r_{ij}$ to compute the average. This procedure will converge and give a value that approximates the integral under the right conditions.

It is important to note that our result will of course depend on the number of spheres. Imagine we had taken a ridiculously small number of spheres (let us say only two!) in a rather large volume and we carried out the distance measurement. There would not be much to average over and on top of that, due to the relative liberty of positioning the two spheres in the volume, the distance (and with it the average distance) would fluctuate wildly depending on the setup. If we repeat this $N$ times and average over it, the result will thus not converge as nicely as it would with a relatively large number $N$ of spheres where the space left would be small. Placing many spheres in the volume improves convergence but slows down the algorithm as with more spheres a higher rate of rejected positions is involved. For large $N$ it is even possible that the last few spheres cannot be placed at all!

## 1.5 Canonical Monte Carlo

This section is supposed to quickly refresh your memory of what an ensemble is. It is a short mini-review of some of the things you should already know from our treatment of thermodynamics as one theoretical basis of the Molecular Dynamics simulation method.

### 1.5.1 Ensembles

An ensemble is a set of a large number of identical systems, called a *Gibbs ensemble*. In the context of physics, one usually considers the phase space (which for $N$ particles is $6N$ dimensional) of a given system. The selected points are then regarded as a collection of representative points in phase space. An important and ever-recurring topic is the probability measure (on which the statistical properties depend). Let us say that we have two regions $R_1$ and $R_2$

with $R_1$ having a larger measure than $R_2$. Consequently, if we pick a system at random from our ensemble, it is more probable that it is in a microstate pertaining to $R_1$ than $R_2$. The choice of this probability measure depends on the specific details of the system as well as on the assumptions made about the ensemble. The normalizing factor of the measure is called the *partition function of the ensemble*. An ensemble is said to be stationary if the associated measure is time independent.

The *ensemble average* of an observable (i.e. of a real-valued function $f$) defined on phase space with the probability measure $d\mu$ (restricting to $\mu$-integrable variables) is defined as

$$\langle f \rangle = \int_A f \, d\mu. \tag{1.21}$$

The *time average* is defined in a different way. We start out with a representative starting point $x(0)$ in phase space. Then, the time average of a function $f$ evolving in phase space is given by

$$\bar{f}_t = \lim_{T\to\infty} \frac{1}{T} \int_0^T f(x(t)) \, dt. \tag{1.22}$$

The most important ensembles are the following:

- **Microcanonical ensemble:** An isolated system with constant number of particles $N$, constant volume $V$ and constant inner energy $E$.
- **Canonical ensemble:** A closed system in a heat reservoir with constant $N$, constant temperature $T$ and constant $V$.
- **Grand canonical ensemble:** An open system where the chemical potential $\mu$ is constant along with $V$ and $T$.

These two averages are connected by the *ergodic hypothesis*. This hypothesis states that for long periods of time, the time one particle spends in the microstates of some region $\mathbf{\Omega}$ of phase space with the same energy is directly proportional to the volume $V(\mathbf{\Omega})$ of that region. One can thus conclude that all the accessible microstates are equiprobable (over a long period of time). In statistical analysis, one often assumes that the time average $\bar{Q}_t$ of some quantity $Q$ and the ensemble average $\langle Q \rangle$ are the same.

### 1.5.2 Monte Carlo in the Canonical Ensemble

Let the energy of a configuration $X$ be given by $E(X)$, then the probability (at thermal equilibrium) for a system to be in state $X$ is given by the *Boltzmann distribution*:

$$p_{eq}(X) = \frac{1}{Z_C} e^{-\frac{E(X)}{k_B T}}, \tag{1.23}$$

with the partition function $Z_C$ (the normalization function of the measure)

$$Z_C = \sum_X e^{-\frac{E(X)}{k_B T}}. \tag{1.24}$$

If we sum over *all* probabilities of a system to be in state $X$, we obtain 1. This normalization expresses the certainty, that the system *must* be found in at least one state when we look *everywhere* in phase space, i.e.:

$$\sum_X p_{eq}(X) = 1. \tag{1.25}$$

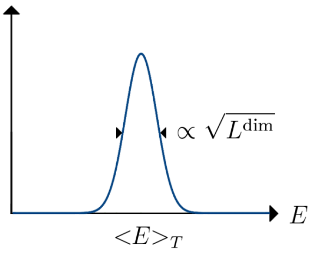


**Figure 1.5.** The energy distribution.

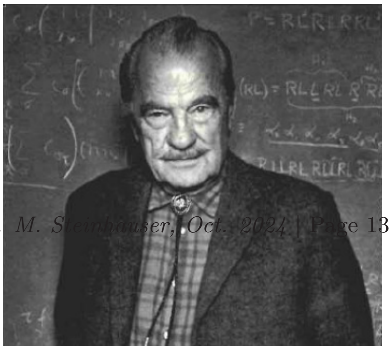
Let us now consider an ensemble average for a quantity $Q$ which for discrete $X$ is given by:

$$\langle Q \rangle = \sum_X Q(X) p_{eq}(X). \tag{1.26}$$

Let's assume that we want to calculate the ensemble average for a given property, e.g. the energy. Unfortunately, the distribution of energy around the time average $\bar{E}_t$ gets sharper with increasing system size (the peak width increases with the system size as $\sqrt{L^{\dim}}$ while the system increases with $L^{\dim}$. such that the relative width decreases as $1/\sqrt{L^{\dim}}$. Consequently, it is inefficient to pick equally distributed configurations over the energy, which is similar to the situation with singularities of functions as considered in Sec. 1.2.1.

## 1.6 The Metropolis Algorithm

This algorithm was first formulated by *N. C. Metropolis*, *A. W. Rosenbluth*, *M. N. Rosenbluth*, *A. H. Teller* and *E. Teller* in 1953. It is a *special case of a Markov chain*, where the $i$-th element $X_i$ only depends on the previous



one, $X_{i-1}$. More precisely, the idea is that we carry out the previously established method of importance sampling through a Markov chain, going from $X_1$ to $X_2$ to $X_3$ and so on, where the probability for a given configuration $X_i$ is $p_{eq}(X_i)$.

In the following we describe the Metropolis algorithm for the case of a system where the configurations $X_i$ are described by a position vector $\vec{r}$. The algorithm is basically a repetition of three essential steps.

**The Metropolis Algorithm**

1. A new position $\vec{y} = \vec{r} + \kappa\vec{q}$ is chosen, where $\vec{q}$ is a random vector with components between $[-1, 1]$ and $\kappa$ is a fixed number.
2. Calculate the energy difference $\Delta E = E(\vec{y}) - E(\vec{r})$ between the original position $\vec{r}$ and the new position $\vec{y}$.
3. If $\Delta E \leq 0$, then accept the new position $\vec{y}$ as a new one, however, if $\Delta E > 0$, accept $\vec{y}$ only as a new position with probability $p_{eq}$, the Boltzmann distribution.

In practice, for step 3, one draws a random number $\xi$ from the intervall $[0, 1]$ and compares it with probability $p_{eq}$. The new position is then accepted if $\xi < p_{eq}$.

Small values of $\kappa$ lead to large acceptance rates, however, in this case, there is the disadvantage of strong autocorrelations $\tau$ between successive states of the system:

$$\tau = \sum_{i=0}^{\infty} \Gamma(i) = \sum_{i=0}^{\infty} \langle (A_0 - \langle A \rangle)(A_i - \langle A \rangle) \rangle, \tag{1.27}$$

where $A$ denotes the different states of the system. Large values of $\kappa$ reduce the autocorrelation of states (i.e. the memory that states have of their predecessor states), but have the disadvantage of having a very low acceptance rate of new positions.

**Listing 1** The Metropolis algorithm

```
                                                        ▷ Step 1
1. Start with a system in an arbitrarily chosen state $X$ and calculate the
energy $E_X$ of the system.
                                                        ▷ Step 2
2. Generate a new state $Y$ by a small *ergodic* perturbation to $X$ and calculate
the energy $E_Y$ of the system.
                                                        ▷ Step 3
3.
if $(E_Y - E_X) \leq 0$ then
    accept the new state $Y$,
else
    if $(E_Y - E_X) > 0$ then
        accept the new state with probability $exp[-\beta(E_Y - E_X)]$;
     return to Step 2 and repeat until equilibrium is reached.
    end if
end if
```

In a more formal way, the Metropolis algorithm can be written as pseudocode as displayed in Algorithm 1.

### 1.6.1 Properties of Markov Chains

As the Metropolis algorithm is a special case of a Markov chain, we want to look at their properties in this section. We start our considerations in a given configuration $X$ and propose a new configuration $Y$ with a **transition probability** $T(X \to Y)$. Again, we want the sum over all possible new configurations to be unity (normalization condition of Eq. (1.25). Furthermore, the probability associated with the transition from a configuration $X$ to a configuration $Y$ needs to be the same as the transition probability of the inverse process $T(Y \to X)$ (reversibility). Last but not least, we recall from thermodynamics the ergodic hypothesis, see Sec. 1.5.1, which states that thermodynamical systems are generally chaotic (assumption of *molecular chaos*) so that all phase space volume elements corresponding to a given energy are equiprobable. More precisely, for almost all measurable quantities $O$ the time averaged value $\bar{O}_t$ is equal to the ensemble average $\langle O \rangle$, i.e. $\bar{O}_t = \langle O \rangle$. In the context of Markov chains this means that a state $i$ is ergodic if it is *aperiodic* and *positive recurrent*.[1] Even though the hitting time is finite, the state $i$ does not need to have a finite expectation value $M_i = \mathrm{Exp}[T_i]$. If however, the state is *positively recurrent*, $M_i$

[1] Positive recurrent means the following: If we start in a state $i$ and define $T_i$ to be the first return time to our state $i$ ("hitting time"), $T_i = \inf\{n \geq 1 : X_n = i | X_0 = i\}$, then the state $i$ is recurrent if and only if $P(T_i \to \infty) = 0$ (i.e. the hitting time is always finite).

is finite. If *all* states in a Markov chain are ergodic, then the chain is called ergodic. Consequently, one must be able to reach any configuration $X$ after a finite number of steps.

A Markov Chain *transition probability* $T(X \to Y)$ has the following properties:

- **Ergodicity:** One must be able to reach any configuration $X$ after a finite number of steps.
- **Normalization:** $\sum_Y T(X \to Y) = 1$.
- **Reversibility:** $T(X \to Y) = T(Y \to X)$.

Now that we have established the properties of Markov chains, we want to understand the second step of the Monte Carlo approach which is making sure that the new configuration is in accordance with the desired properties of the considered system. In the context of our example of hard spheres with radius $R$ in Sec. 1.4, this referred to the second step formulated on Page 10 which made sure that no spheres may overlap. We recall that the first step in our consideration is *proposing* a new configuration $Y$. However, not every new configuration $Y$ is *accepted*. Instead, we propose a new configuration and denote an acceptance probability which will make sure that the model behaves the way we want. We denote the probability of acceptance of a new configuration $Y$ , starting from a configuration $X$, by $A(X \to Y)$. In practice, we might be interested a bit more in the overall probability of a configuration actually making it through these two steps; this probability is the product of the transition probability $T(X \to Y)$ and the acceptance probability $A(X \to Y)$, and is called the **probability of a Markov chain**:

$$W(X \to Y) = T(X \to Y) \cdot A(X \to Y). \tag{1.28}$$

If we are interested in the evolution of the probability $p(X, t)$ to find $X$ in time, we can derive this with a logical "Gedankenexperiment": There are two processes changing the probability $p(X, t)$:

- A configuration $X$ is produced by coming from $Y$ (this will contribute positively).
- A configuration $X$ is destroyed by going to some other configuration (this will decrease the probability).

The first of these two processes is proportional to the probability for a system to be in state $Y, p(Y)$, while the second one needs to be proportional to the probability for a system to be in $X, p(X)$. When we combine all of this we obtain a so-called **master equation**:

$$\frac{dp(X,t)}{dt} = \sum_Y p(Y)W(Y \to X) - \sum_Y p(X)W(X \to Y). \tag{1.29}$$

At this point we can conclude the properties of $W(X \to Y)$: Similarly to the transition probability $T(X \to X)$, the first property is ergodicity, but it manifests itself in a somewhat different way. Given that any configuration $X$ must be reachable after a finite number of steps, the transition probability $T(X \to Y)$ cannot be zero because otherwise a state $Z$ with $T(X \to Z) = 0$ would not be reachable, and the acceptance rate is non-zero as well, so we have $\forall X, Y,\ W(X \to Y) > 0$.

Next, again, we sum over all possible configurations to obtain unity, giving us the second property (normalization). The third property, homogeneity, is new. It tells us that if we sum over all initial configurations, the product of the probability for a system to be in $Y$ multiplied by its Markov probability for the transition to $X$ is given by the probability for a system to be in $X$. In other words, the probability for a system to be in $X$ is simply a result of systems coming from other configurations over to $X$.

A Markov Chain *probability* $W(X \to Y)$ has the following properties:

- **Ergodicity:** $\forall X, Y,\ W(X \to Y) > 0$.
- **Normalization:** $\sum_Y W(X \to Y) = 1$.
- **Homogeneity:** $\sum_Y p(Y)W(Y \to X) = p(X)$.

**Think!**

We note that the attentive reader might object that $W$ cannot possibly ever reach $W(X \to Y) = 1$, because $W(X \to Y) = A(X \to Y) \cdot T(X \to Y)$, where both, $A(X \to Y)$ and $T(X \to Y)$ are probabilities with $A \leq 1$, $T \leq 1$. The answer to this objection is that we are looking at *conditional* probabilities here, as $A$ is the probability of acceptance *for a given new configuration* $Y$, *provided that* $T$ is given. In mathematical notation, this is expressed as $P(A|T)$.

Please note that the event of choosing a new configuration with transition probability $T(Y \to Y)$ is independent of the probability $A(X \to Y)$ of accepting it. Therefore, we have

$$\sum_Y P(T) \quad = \quad 1, \tag{1.30a}$$

$$\sum_Y P(A) \quad = \quad 1, \tag{1.30b}$$

and the conditional probability $P(A|T)$, "$A$ provided that $T$ is given", i.e. the probability to accept a new configuration produced by $T$, is

$$P(A|T) = \frac{P(A \cup T)}{P(T)} = \frac{W(X \to Y)}{T(X \to Y)}, \tag{1.31}$$

where we can now easily identify our original $A(X \to Y)$ as the conditional probability $P(A|T)$. Furthermore, for two independent events $A$ and $B$

$$\sum_i A_i = 1, \tag{1.32a}$$

$$\sum_i B_i = 1, \tag{1.32b}$$

we have

$$\begin{aligned} \sum_{ij} A_i B_i \quad &= \quad (A_1 + A_2 + \cdots)(B_1 + B_2 + \cdots) = \qquad (1.33) \\ &\overset{\sum B_i = 1}{=} \quad A_1 B_1 + A_1(1 - B_1) + A_2 B_2 + A_2(1 - B - 2) + \cdots = \\ &= \quad A_1 + A_2 + \cdots = 1, \end{aligned}$$

so for two independent events we get 1 again, and thus the expression is normalizable. When we apply this to the example at hand, we find

$$\begin{aligned} P(T(X \to Y))P(A(X \to Z)) = 1 \quad &\overset{Y=Z}{=} \quad \sum_Y T(X \to Y)A(X \to Y) \qquad (1.34) \\ &= \quad \sum_Y W(X \to Y). \end{aligned}$$

### 1.6.2 Detailed Balance

Let us now consider a stationary state, i.e. a state where the statistical measure is time independent:

$$\frac{dp(X,t)}{dt} = 0 \tag{1.35}$$

Note that all Markov processes reach a steady state. However, we want to model the *thermal equilibrium*, so we use the Boltmann distribution

$$p_{st}(X) = p_{eq}(X), \tag{1.36}$$

to obtain

$$\sum_Y p_{eq}(Y)W(Y \to X) = \sum_Y p_{eq}(X)W(x \to Y). \tag{1.37}$$

Hence, we find the **detailed balance condition** (which is a sufficient condition) as:

$$p_{eq}(Y)W(Y \to X) = p_{eq}(X)W(X \to Y). \tag{1.38}$$

If the condition of Eq. (1.38) is met, the steady state of the Markov process is the thermal equilibrium. We have achieved this by using the Boltmann distribution for $p(X)$.

### 1.6.3 Glauber Dynamics

The condition of detailed balance is a very broad condition permitting many approaches for Markov processes. Another such approach was proposed by *Roy J. Glauber*, who later obtained the Nobel prize for his contributions to optics. His proposal for the acceptance probability was

$$A_G(X \to Y) = \frac{e^{-\frac{\Delta E}{k_B T}}}{1 + e^{-\frac{\delta E}{k_B T}}}. \tag{1.39}$$

**Figure 1.7.** Roy Glauber at the meeting of the Nobel Laureates in Lindau, 2012.

This also fulfills the detailed balance condition:

$$p_{eq}(X) W(X \to Y) = p_{eq}(Y) W(Y \to X). \tag{1.40}$$

Let us prove this. Our claim is that the acceptance probability of Eq. (1.39) as proposed by *Roy Glauber* fulfills the detailed balance condition of Eq. (1.40).

**Proof:** In a first step, let us reduce the condition of Eq. (1.40) to a simpler expression. We write out

$$W(X \to Y) = A(X \to Y) T(X \to Y) \tag{1.41}$$

to obtain

$$p_{eq}(X) A(X \to Y) T(X \to Y) = p_{eq}(Y) A(Y \to X) T(Y \to X), \tag{1.42}$$

where we can use the reversibility, i.e. $T(X \to Y) = T(Y \to X)$ to find

$$p_{eq}(X) A(X \to Y) = p_{eq}(Y) A(Y \to X). \tag{1.43}$$

It is thus sufficient to show that Eq. (1.40) fulfills Eq. (1.43). We can adapt the condition even further by recalling that

$$\frac{p_{eq}(Y)}{p_{eq}(X)} = e^{-\frac{\Delta E}{k_B T}}, \tag{1.44}$$

where the partition function $Z$ drops out. We can thus rewrite Eq. (1.43) by dividing both sides by $p_{eq}(X)$ to obtain

$$A(X \to Y) = \frac{p_{eq}(Y)}{p_{eq}(X)} A(Y \to X) = e^{-\frac{\Delta E}{k_B T}} A(Y \to X). \tag{1.45}$$

Hence, it is sufficient to show that (1.40) fulfills (1.45). To finally prove this we can simply write

$$\frac{A_G(X \to Y)}{A_G(Y \to X)} = \frac{e^{-\frac{\Delta E}{k_B T}}}{1 + e^{-\frac{\Delta E}{k_B T}}} \left( \frac{e^{\frac{\Delta E}{k_B T}}}{1 + e^{\frac{\Delta E}{kBT}}} \right)^{-1} = e^{-\frac{\Delta E}{k_B T}} \cdot \frac{1 + e^{\frac{\Delta E}{k_B T}}}{e^{\frac{\Delta E}{k_B T}} + 1} = e^{-\frac{\Delta E}{k_B T}} \tag{1.46}$$

Hence, $A_G$ fulfills the detailed balance condition. ■

The Glauber dynamics has some advantages over the Metropolis algorithm once you go to low temperatures because of the different formulation of the acceptance rate.

## 1.7 Final Remarks

The Monte Carlo techniques that are described in this leture can be used to compute the equilibrium properties of classical many-body systems. In this context, the word 'classical' means that the core motion of the constituent particles obeys the laws of classical mechanics. This is an excellent approximation for a wide range of materials. Only when we consider the translational or rotational motion of light atoms or molecules ($He$, $H_2$, $D_2$) or vibrational motion with a frequency such that $h > k_B T$, should we worry about quantum effects.

Before the advent of electronic computers, i.e. in the 1930s and beginning 1940s of last century, there was only one way to predict the outcome of an experiment, namely by making use of a theory that provided an approximate description of the system under consideration. The reason why an approximate theory was almost always used is that there are very few model systems for which the equilibrium properties can be computed exactly (examples are the ideal gas, the harmonic crystal and a number of lattice models, such as the two-dimensional Ising model for ferro-magnets), and even fewer model systems for which the transport properties can be computed exactly. As a result, most properties of real materials were predicted on the basis of approximate theories (examples are the van der Waals equation for dense gases, the Debye-HÃ¼ckel theory for electrolytes, or the Boltzmann equation to describe the transport properties of dilute gases). Given sufficient information about the intermolecular interactions, these theories will provide us with an estimate of the properties of interest.

Unfortunately, our knowledge of the intermolecular interactions of all but the simplest molecules is quite limited. This leads to a problem if we wish to test the validity of a particular theory by comparing directly to experiment. If we find that theory and experiment disagree, it may mean that our theory is wrong, or that we have an incorrect estimate of the intermolecular interactions, or both Clearly, it would be very nice when we could obtain essentially exact results for a given model system, without having to rely on approximate theories. Computer simulations allow us to do precisely that. On the one hand, we can now compare the calculated properties of a model system with those of an experimental system: if the two disagree, our model is inadequate, i.e. we have to improve on our estimate of the intermolecular interactions. On the other hand, we can compare the result of a simulation of a given model system with the predictions of an approximate analytical theory applied to the same model. If we now find that theory and simulation disagree, we know that the theory is flawed. So, in this case, the computer simulation plays the role of the 'experiment' that is designed to test the theory. This method to test theories before we apply them to the real world, is called a 'computer experiment'. Computer experiments have become standard practice, to the extent that they now provide the first (and often the only) test of a new theoretical result.

In fact, the early history of computer simulation illustrates this role of computer simulation. In some areas of physics there appeared to be little need for simulation because very good analytical theories were available (e.g. to predict the properties of dilute gases or of nearly harmonic crystalline solids). However, in other areas, few if any 'exact' theoretical results were known, and progress was much hindered by the fact that there were no unambiguous tests to assess the quality of approximate theories. A case in point was the theory of dense liquids. Before the advent of computer simulations, the only way to model liquids was by mechanical simulation of large assemblies of macroscopic spheres (e.g. ball bearings). But such mechanical models are rather crude, as they ignore the effect of thermal motion. Moreover, the analysis of the structures generated by mechanical simulation was very laborious and, in the end, had to be performed by computer anyway. This may explain why, when in 1953 electronic computers were, for the first time, made available for non-military research, numerical simulation of dense liquids was one of the first problems that was tackled. In fact, the first simulation of a 'liquid' was carried out by *Metropolis* and co-workers at Los Alamos, using (or, more properly, *introducing*) the Monte Carlo method. Almost simultaneously, *Fermi*, *Pasta* and *Ulam* performed a well-known numerical study of the dynamics of an anharmonic, one-dimensional crystal. However, the first proper Molecular Dynamics simulations were reported in 1956 by *Alder* and *Wainwright* at Lawrence Livermore Laboratories, who studied the dynamics of an assembly of hard spheres. The first MD simulation of a model for a 'real' material was reported in 1959 (and published in 1960) by the group of Vineyard at Brookhaven, who simulated radiation damage in crystalline *Cu*. MD simulation of a real liquid (argon) was reported in 1964 by *Rahman* at Argonne National Laboratories. After that, computers were increasingly becoming available to scientists outside the US government labs, and the practice of simulation started spreading to other continents. Much of the methodology of MD (and MC) simulations has been developed since then, although it is fair to say that the basic algorithms for MC and MD have hardly changed since the early beginnings of non-military scientific computing in the 1950s.