Computational Physics

KATHOLIEKE UNIVERSITEIT LEUVEN

Advanced Monte Carlo Methods

notes by E. Carlon – Academic year 2012/2013

Contents

1 Calculation of integrals by Monte Carlo

The term "Monte Carlo" first appeared in an article of 1949 by Metropolis and Ulam[1]. These authors suggested a method to solve a class of problems in Physics and Mathematics using a statistical approach. Actually similar ideas, under the name of statistical sampling, were used earlier, much before the invention of computers.

A classical example is the so-called problem of the Buffon's needle, named after Georges-Louis Leclerc, Comte de Buffon (1707-1788), a french naturalist and mathematician. Buffon considered a needle of length l thrown at random on a plane in which are drawn some parallel long lines separated by a distance t (see Fig. 1). Defining x the position of the center of the needle ($0 \leq x \leq l$) and θ the angle formed by the needle with the axis perpendicular to the lines ($-\pi/2 \leq \theta \leq \pi/2$) one can easily find the probability that the needle crosses the lines from

$$P = \frac{2}{t\pi} \int_0^{l/2} dx \int_{-\arccos(2x/l)}^{\arccos(2x/l)} d\theta = \frac{2l}{t\pi} \int_0^1 dy \ \arccos(y) = \frac{2l}{t\pi} \tag{1}$$

Imagine now to throw the needle N times, and let N^* the number of times it crosses the lines. We could estimate the probability as $P \approx N^*/N$. During the 19th and 20th centuries several people performed the needle experiment to obtain an estimate of the number π as

$$\pi \approx \frac{2l}{t} \frac{N}{N^*} \tag{2}$$

which follows from Eq. (1). Obviously the result becomes more and more accurate as N increases[2].

This calculation is an example of the use of statistical sampling to estimate the double integral appearing in Eq. (1). Usually, up to three dimensions, integrals can be calculated more efficiently using quadrature methods. For high dimensional integrals, however, the Monte Carlo method becomes much more efficient than quadratures. To

[1] N. Metropolis and S. Ulam *The Monte Carlo method*, JSTOR **44**, 335 (1949)

[2] An interactive Java applet for the calculation of π by sampling the ratio of needles crossing the lines over the total number of needles can be found at `http://mste.illinois.edu/reese/buffon/bufjava.html`

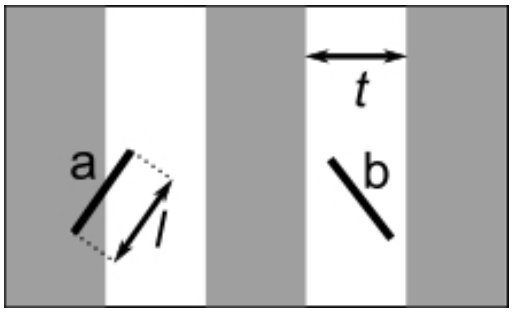

Figure 1: The Buffon's needle: the needle a crosses the line, while the b does not.

illustrate how Monte Carlo integration works we restrict ourselves in this section to some simple one-dimensional examples.

Consider a function of one variable defined in the interval $[0, 1]$. We can estimate its integral by generating N random numbers x_1, x_2 $\ldots x_N$ uniformly in $[0, 1]$ and write[3]:

$$I = \int_0^1 f(x)\ dx \approx \frac{1}{N} \sum_{n=1}^{N} f(x_n) \tag{4}$$

If $f(x)$ is a slowly varying function the approximant converges very rapidly to the exact value (if $f(x)$ is a constant it is sufficient to sample it in one point). In some cases we could have functions which are negligibly small in (large) fraction of the integration domain. In that case the uniform random number distribution is not very efficient: most of the time random numbers are generated in a part of the integration domain where the function is small. Despite a considerable computational effort we may end up with an estimate of the integrand with large statistical error. It is then more convenient to use a random number generator which samples more often the part of the integration domain where the function $f(x)$ has relatively large value. This procedure is called *importance sampling*.

A biased random number generator in the interval $[0, 1]$ is characterized by a density function $W(x)$, which has the following properties $W(x) \geq 0$ in $[0, 1]$ and

$$\int_0^1 W(x)\ dx = 1 \tag{5}$$

$W(x)\ dx$ is equal to the fraction of random numbers generated in the interval $[x, x+dx]$. If N random numbers are generated then the distance between two of these at position x is given by $1/NW(x)$. As a consequence

$$I = \int_0^1 f(x)\ dx \approx \frac{1}{N} \sum_{n=1}^{N} \frac{f(x_n)}{W(x_n)} \tag{6}$$

where x_n are generated according to the distribution $W(x)$.

[3]More generally if the domain of integration is $[a, b]$, the integral is approximated as

$$I = \int_a^b f(x)\ dx \approx \frac{b-a}{N} \sum_{n=1}^{N} f(x_n) \tag{3}$$

where x_n is a random number selected uniformly from $[a, b]$.

Example 1
Let us consider the following function $f(x) = e^{-x^2} g(x)$ in $[0, +\infty)$, where $g(x)$ is slowly varying. For the Monte Carlo computation of this integral it is convenient to use a Gaussian random number generator, characterized by a density $W(x) = \sqrt{2/\pi} e^{-x^2}$. One has:

$$I = \int_0^{+\infty} f(x)\, dx \approx \frac{1}{N} \sum_{n=1}^{N} \frac{f(x_n)}{W(x_n)} = \frac{1}{N} \sqrt{\frac{\pi}{2}} \sum_{n=1}^{N} g(x_n) \tag{7}$$

Note that the domain of integration is unbounded here.
The normalization factor should not be forgotten in the calculation! In this example this is $\sqrt{2/\pi}$.

As seen in the previous example it is convenient to choose $W(x)$ so that $f(x)/W(x)$ becomes a slow varying function. This statement can be made more precise from the calculation of the variance:

$$\sigma_I^2 = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left(\frac{f(x_i)}{W(x_i)} - \langle f/W \rangle \right) \left(\frac{f(x_j)}{W(x_j)} - \langle f/W \rangle \right) \tag{8}$$

Here $\langle f/W \rangle$ denotes the exact value of the integral, obtained from sampling with an infinite number of points. Using the fact that different samples ($i \neq j$) are independent one obtains in the limit $N \to \infty$

$$\sigma_I^2 = \frac{1}{N} \left[\langle (f/W)^2 \rangle - \langle f/W \rangle^2 \right] \tag{9}$$

Thus the error scales as the squared root of the number of sampled points ($\sigma_I \sim 1/\sqrt{N}$), but can also be minimized by choosing W such that f/W is a slowly varying function so that the term between brackets in Eq. (9) becomes small. In the optimal case one should try to reduce f/W to a constant, so that the term between brackets in Eq. (9) vanishes. However in the most general case it is impossible to do so. Note this would mean $W(x) = \alpha f(x)$. As $W(x)$ is normalized (Eq. (5)), we have $I = 1/\alpha$, which means that we know already the value of the integral.

The choice of $W(x)$ is obvious for functions which are modulated by gaussians, exponentials ...

2 Monte Carlo methods for systems in equilibrium

Consider a system in thermal equilibrium which is in contact with a reservoir at a temperature T. The system is characterized by some discrete[4] number of microscopic configurations, labeled by μ. Let E_μ be the energy of the configuration μ. According to one of the basic principles of statistical mechanics, for a system in thermodynamic equilibrium, the probability of finding it the state μ is proportional to $e^{-\beta E_\mu}$, where $\beta = 1/k_B T$ and k_B the Boltzmann constant.

Let A_μ now be an observable of the system. This could be the energy itself ($A_\mu = E_\mu$) or any other physical quantities which characterizes any given microstate μ. For a system in equilibrium at a temperature T the average value of the observable is given by:

$$\langle A \rangle = \frac{1}{Z} \sum_\mu A_\mu e^{-\beta E_\mu} \tag{10}$$

where we have introduced the partition function:

$$Z = \sum_\mu e^{-\beta E_\mu} \tag{11}$$

Can we use the Monte Carlo method to get an approximate, but accurate, estimate of $\langle A \rangle$? One possibility is to generate N configurations of our system at random. Let us denote this subset with $\{\mu\}$. In analogy to what seen in the previous Section we can approximate the thermal average with

$$\langle A \rangle \approx \frac{\sum_{\{\mu\}} A_\mu e^{-\beta E_\mu}}{\sum_{\{\mu\}} e^{-\beta E_\mu}} \tag{12}$$

where the sum is restricted to a subset $\{\mu\}$ of sampled configurations.

How can we select the configuration in $\{\mu\}$? One possibility is to perform an *random* sampling. As an example consider an Ising[5] square $L \times L$ lattice, containig thus L^2 spins. Random sampling consists in generating configurations by choosing L^2 spins either up or down with equal probability. Each time a new configuration is generated, this is uncorrelated with the previous ones. However, this is a very inefficient strategy for the calculation of $\langle A \rangle$. The reason is the following. Even a "small" system, as an Ising square lattice with $L = 20$, has a large number of possible configurations. Each spin can take two values and there are L^2 spins, hence $L = 20$ has a configurational space of dimension $2^{400} \approx 10^{120}$. Considering that with modern computers we could sample 10^{10} configurations within a few hours of CPU time, this remains a negligible fraction of the configuration space. An additional problem is the case in which the sum in Eq. (10) is dominated by very few states, as in the Ising model at low temperatures in the ferromagnetic phase where the large majority of the spins are pointing to the same direction. Random sampling most likely generates configurations where the total magnetization is small, as the spins are chosen independently from each other.

[4]For the sake of simplicity we consider the discrete case, but all what we discuss can be easily generalized to a continuum system, as for instance an interacting gas or a liquid.

[5]The Ising model in briefly introduced in Appendix A

Following what we have learnt in the previous section we should generate configurations of the system non-uniformly. If a given state μ is generated with a probability W_μ then Eq. (10) takes the form

$$\langle A \rangle \approx \frac{\sum_{\{\mu\}} A_\mu e^{-\beta E_\mu} W_\mu^{-1}}{\sum_{\{\mu\}} e^{-\beta E_\mu} W_\mu^{-1}} \tag{13}$$

which is analogous to the Eq. (6). The most natural choice is $W_\mu = e^{-\beta E_\mu}/Z$, which yields:

$$\langle A \rangle \approx \frac{\sum_{\{\mu\}} A_\mu}{\sum_{\{\mu\}} 1} = \frac{1}{N} \sum_{\{\mu\}} A_\mu \tag{14}$$

where N is the number of terms in the sum of Eq. (14).

How is it possible to generate states according to the Boltzmann factor $\exp(-\beta E_\mu)$? This is explained in the following sections. First we have to introduce the concept of Markov chain.

2.1 Markov chains

A Markov chain is a discrete stochastic process which starting from a configuration μ at time $t = 0$ generates a new configuration at a time $t + \Delta t$

$$\mu \to \gamma \to \nu \dots \tag{15}$$

Each transition depends only on the current state of the system and not on the previous history (Markov property). The evolution is stochastic i.e. is governed by transition probabilities $P(\mu \to \nu) \geq 0$. The following normalization condition holds

$$\sum_\nu P(\mu \to \nu) = 1 \tag{16}$$

The system at a given time is characterized by the probabilities p_μ of being in the state μ. We can write the probability of finding the system at a later time in a state ν as:

$$p_\nu(t + \Delta t) = \sum_\mu p_\mu(t) P(\mu \to \nu) \tag{17}$$

If the system has N states, we can arrange the p_μ's so to form a vector of dimension N and rewrite Eq. (17) in the form of a vector-matrix product

$$p(t + \Delta t) = P\ p(t) \tag{18}$$

where the matrix P has elements $P_{\mu\nu} = P(\nu \to \mu)$. Equation (18) can be rewritten also in the form of a differential equation in the limit $\Delta t \to 0$, as shown in Appendix B. In that form it is usually referred to as the Master equation.

Note that the condition (16) implies that the sum of each column of the matrix is equal to 1. We will call p a stochastic vector if 1) it has non-negative elements and

2) $\sum_\mu p_\mu = 1$. A square matrix with non-negative elements and such that the sum of all columns is equal to 1 is called stochastic matrix. Note that in general a stochastic matrix is not symmetric, i.e. $P(\mu \to \nu) \neq P(\nu \to \mu)$

The following proposition will be proven during the exercise session

Proposition
If p is a stochastic vector and P a stochastic matrix, then $P\ p$ is a stochastic vector. P has at least an eigenvalue $\lambda = 1$. All other eigenvalues will be such that $|\lambda_k| < 1$.

Example 2
Consider a system composed of 3 states μ, ν and γ. The stochastic dynamics is defined by the probabilities $P(\mu \to \nu) = a$, $P(\nu \to \gamma) = P(\gamma \to \nu) = b$ $(a, b < 1)$. All other probabilities for processes $\mu \to \gamma$, $\gamma \to \mu$ and $\nu \to \mu$ are zero. The stochastic matrix associated is

$$P = \begin{pmatrix} 1-a & 0 & 0 \\ a & 1-b & b \\ 0 & b & 1-b \end{pmatrix} \tag{19}$$

The eigenvalues are $\lambda_1 = 1$, $\lambda_2 = 1 - a$ and $\lambda_3 = 1 - 2b$. Note that in agreement with the previous proposition an eigenvalue is equal to 1, while the two others $|\lambda_1|$, $|\lambda_2| < 1$.
Particularly interesting is the eigenvector associated to $\lambda_1 = 1$, which is given by $\omega = (0, 1/2, 1/2)$. This is a stochastic vector.

We are interested in the eigenvector ω associated to the eigenvalue 1 of the stochastic matrix since this eigenvector describes a stationary state of the system, i.e. the probability of finding the system in any of the possible configurations does not evolve with time as from Eq. (18) follows

$$\omega(t + \Delta t) = P\omega(t) = \omega(t) \tag{20}$$

Example 3
In the Example 2 we found that $\omega = (0, 1/2, 1/2)$ is stationary state. This means that given an initial state, the system will evolve towards a stationary state in which it spends half of the time in ν and half in γ. In this example $\omega_\mu = 0$, i.e. we will have probability zero of finding the system in the state μ. Can you understand why?

2.2 Ergodicity

An important property which we wish our system to have is that of *ergodicity*. Ergodicity means that any state of the system should be reachable from any other state. Note that in fact many of the transition rates from and to a given state can be zero, but given two arbitrary states μ and ν there should exist at least a path of transition with non-zero rates connecting them

$$\mu \to \gamma \to \delta \to \rho \to \ldots \to \nu \tag{21}$$

Example 4
The stochastic process given in the Example 2 is non-ergodic. Take γ and μ. There is no way that the defined dynamics connects them: as $P(\gamma \to \mu) = P(\nu \to \mu) = 0$ it is not possible to reach the state μ from any other state of the system.

2.3 Detailed balance

We consider now the "inverse" problem. We would like to generate a stochastic dynamics which has as stationary state a given vector ω. Particularly interesting, on the view of what discussed before is the case of convergence to equilibrium $\omega_\mu = \exp(-\beta E_\mu)/Z$. A given stochastic vector ω_μ is stationary if it is eigenvector of the generator of dynamics P with eigenvalue equal to 1 ($P\omega = \omega$), which means

$$\omega_\nu = \sum_\mu \omega_\mu P(\mu \to \nu) \tag{22}$$

Using Eq. (16) we can rewrite this condition as

$$\sum_\mu \omega_\nu P(\nu \to \mu) = \sum_\mu \omega_\mu P(\mu \to \nu) \tag{23}$$

We refer to Eq. (23) as to the *global balance* condition. The interpretation is that ω is stationary if the total probability flow from a configuration ν to any configuration of the system ($\sum_\mu \omega_\nu P(\nu \to \mu)$) is equal to the flow from any other configuration to ν ($\sum_\mu \omega_\mu P(\mu \to \nu)$).

Note that if the probabilities satisfy the so-called *detailed balance* condition

$$\omega_\nu P(\nu \to \mu) = \omega_\mu P(\mu \to \nu) \tag{24}$$

then Eq. (23) necessarily holds. The opposite is not true: Eq. (23) may be valid, but Eq. (24) be violated (see example in the exercice session). Detailed balance is a condition which is relatively simple to implement in practice, as it can be seen in the following

Example 5
Consider a system with three states. We wish to construct a stochastic matrix which has $\omega = \frac{1}{6}(1\ 4\ 1)$ as stationary state. We can do this by making use of the detailed balance condition (Eq. (24)) and write:

$$\frac{P(1\to 2)}{P(2\to 1)} = \frac{\omega_2}{\omega_1} = 4, \qquad \frac{P(3\to 2)}{P(2\to 3)} = \frac{\omega_2}{\omega_3} = 4, \qquad \frac{P(1\to 3)}{P(3\to 1)} = \frac{\omega_3}{\omega_1} = 1 \tag{25}$$

The detailed balance condition does not impose any constraint on the probabilities of remaining in the same state $P(\mu\to\mu)$. We need anyhow to have a stochastic matrix, so the sums of all columns must add up to one. One possible solution is

$$P = \frac{1}{5}\begin{pmatrix} 0 & 1 & 1 \\ 4 & 3 & 4 \\ 1 & 1 & 0 \end{pmatrix} \tag{26}$$

You can verify explicitely that $P\omega = \omega$, so ω is indeed eigenvector of P with eigenvalue equal to 1.
Can you find other examples of matrices having the above ω as stationary state?

If we wish that our dynamics to the Boltzmann distribution $\omega_\mu = \exp(-\beta E_\mu)/Z$ the detailed balance condition becomes

$$\frac{P(\nu\to\mu)}{P(\mu\to\nu)} = e^{-\beta(E_\mu - E_\nu)} \tag{27}$$

As this condition only fixes the ratios of the $P(\nu\to\mu)$, there is quite some freedom on the specific choice of transition probabilities satisfying detailed balance.

2.4 The Metropolis algorithm

Ergodicity and detailed balance are the two important conditions that our stochastic Markov chain has to fulfill in order to converge to thermodynamic equilibrium. The rates must also fulfill the normalization condition given in Eq. (16). This normalization includes the rate $P(\mu\to\mu)$ which is associated to the absence of transition. When setting $\nu = \mu$ in Eq. (24) we find $1 = 1$, i.e. the detailed balance condition is always satisfied for any choice of $P(\mu\to\mu)$. Hence we can set it to any values that we wish.

As the purpose is to have quick convergence to equilibrium, the optimal choice of $P(\mu\to\nu)$ to have as large as possible, but still consistent with Eqs. (24) and (16). In practice we can split the rates as follows:

$$P(\mu\to\nu) = g(\mu\to\nu)A(\mu\to\nu) \tag{28}$$

where $g(\mu \to \nu)$ is called selection probability and $A(\mu \to \nu)$ the acceptance ratio. The selection probability tells us which states can be generated by the algorithm from a given initial state. The acceptance ratio is the fraction of times that the actual transition takes place.

Example 6
The simplest example of selection probability is that for an Ising model with single spin flip update. In this case a new state is generated by selecting a spin at random out of a total of M spins in the lattice. This means that in this case the selection probability is $g(\mu \to \nu) = 1/M$ if the two states differ by a single spin, while $g(\mu \to \nu) = 0$ otherwise.

The Metropolis algorithm consists in the choice of the following acceptance ratio:

$$A(\mu \to \nu) = \begin{cases} e^{-\beta(E_\nu - E_\mu)} & \text{if} E_\nu > E_\mu \\ 1 & \text{otherwise} \end{cases} \tag{29}$$

Hence, if an update selected according to the rule defined by $g(\mu \to \nu)$ leads to a lowering of the energy, it is always accepted. Otherwise it is accepted with a probability proportional to the Boltzmann factor associated to the energy difference between the final and initial state.

For any symmetric choice of selection probability ($g(\mu \to \nu) = g(\nu \to \mu)$), it is easy to show that the Metropolis acceptance ratio (Eq. (29)) satisfies detailed balance (Eq. (24)). The proof of this claim is left to the reader.

2.5 Equilibration

The ergodicity and detailed balance conditions imply that the dynamics will converge to equilibrium state. Given that the initial state is usually chosen at random it may take a while before the equilibrium distribution is reached. Figure 2 shows different snapshots of configurations of the Ising model at different simulation times. The top left figure is the random initial configuration. As the temperature is below the critical one the spins tend to aling (ferromagnetic phase). During the simulations, larger and larger domains with aligned spins are formed in the course of time, until these domains reach a characteristic size which does not change in the course of the simulation.

To determine τ_{eq} the equilibration time in a simulation, one can compute some physical quantities (like the total magnetization in an Ising lattice) and plot them as a function of time. When these start to fluctuate around a constant value thermal equilibrium is reached. Figure 3 shows a plot of the magnetization per spin of an Ising system versus the simulation time. As it can be seen the magnetization reaches an average value after about $t = 3000$ Monte Carlo steps per lattice site[6].

[6]Time in a simulation is usually measured in terms attempted flips per lattice site. This is done to have a time unit which allows comparison of two simulations of lattices of different sizes.

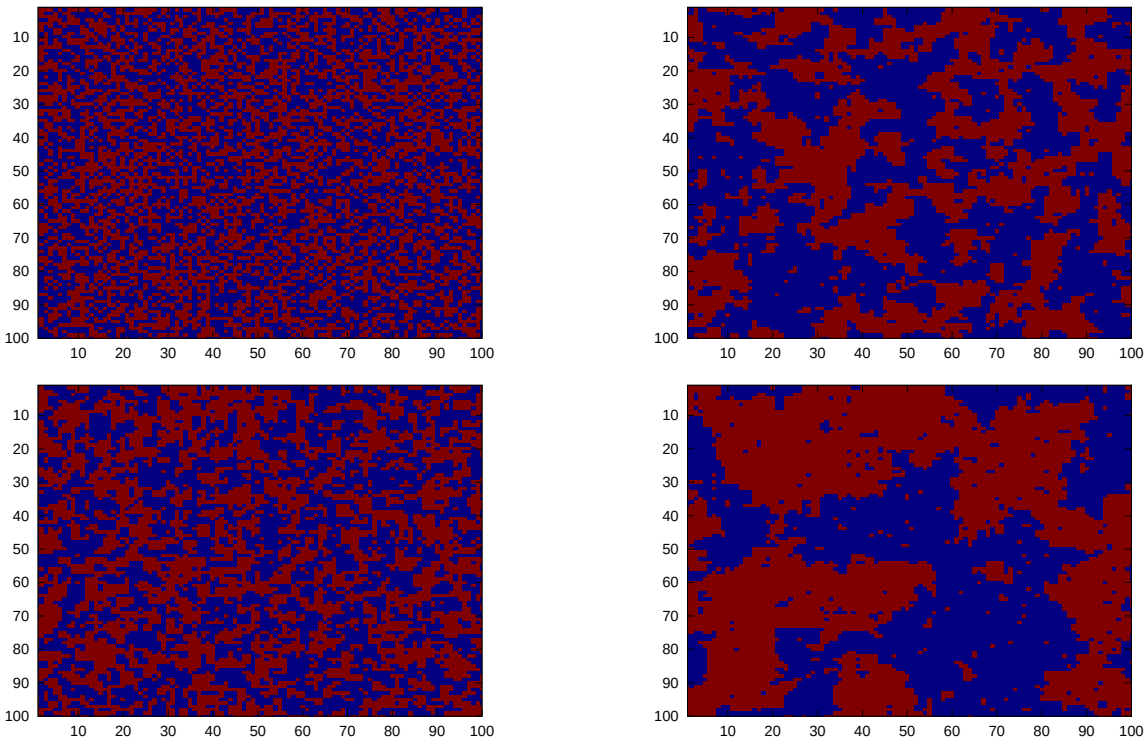

Figure 2: Configurations of a 100×100 Ising lattice at $T = 0.88T_c$ obtained from the Metropolis algoritm at different time steps. Top left: initial configuration, Bottom left: 10^4 Monte Carlo steps, Top right: 10^5 MC steps, Bottom right: 10^6 MC steps. If we measure the time in *sweeps*, e.g. MC steps per spin the times shown correspond to 1, 10 and 100 sweeps.

2.6 Correlation time

A basic requirement for an efficient simulation is that our Markov chain goes through a sufficiently large series of independent states so to make the averages in Eq. (14) sufficiently robusts. In the Ising model with the Metropolis algorithm discussed above at most one spin is flipped at each time step. The Ising magnetization for a lattice with N spins

$$m = \frac{1}{N} \sum_i s_i \tag{30}$$

changes by an amount $\mathcal{O}(1/N)$ at each Monte Carlo step. The consequences is that the new configuration is strongly correlated with the original one and therefore many steps are needed before this correlation is lost.

To quantify this we can measure the correlation time. Let us suppose that we have let the simulation run for a sufficient time so to have reached equilibrium. All the physical quantities fluctuate around their average equilibrium values. Given, for instance, $m(t)$ the magnetization of the Ising model at time t, an interesting quantity

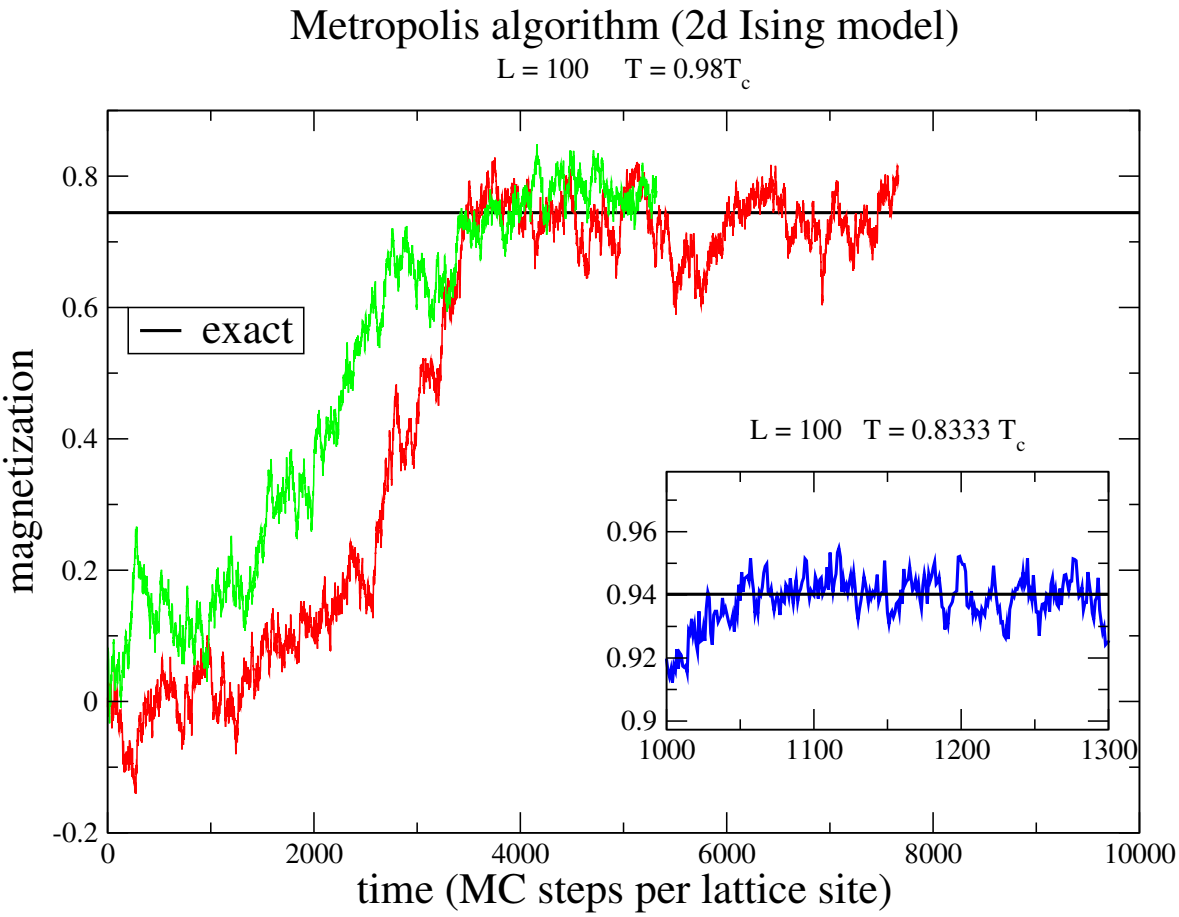

Figure 3: Magnetization of a 100×100 Ising model in the ferromagnetic phase as a function of the simulation time. In the main frame two random different initial configurations were simulated at $T = 0.98T_c$. After about 3000 Monte Carlo sweeps the two curves converge to the same average value. The solid line is the exact value of the magnetization of the 2d Ising model. Inset: A similar run for a lower temperature $T = 0.8333T_c$. Note that the equilibrium magnetization is reached faster ($\tau_{\rm eq} \approx 1000$) compared to the $T = 0.98T_c$ case. Fluctuations around the average magnetization are also much smaller.

is the autocorrelation function[7]:

$$\chi(t) = \int dt' \; [m(t') - \langle m \rangle][m(t'+t) - \langle m \rangle] \tag{31}$$

where $\langle m \rangle$ is the average magnetization value, i.e. average over all values measured during the Monte Carlo run. Consider a very small value of t in Eq. (31); in this case $m(t'+t) \approx m(t')$ and the integrand will have a positive value, hence $\chi(t) > 0$ for t sufficiently small. For sufficiently long times $\chi(t)$ is expected to decay to zero. This is because $m(t'+t)$ is expected to become uncorrelated to $m(t')$, so if $m(t') > \langle m \rangle$, we expect equally likely a fluctuation of m above or below its average values at times $t'+t$. When integrated in dt' the positive and negative contributions will cancel and the integral in Eq. (31) will vanish. Indeed it does so with a leading asymptotic behavior of the type:

$$\chi(t) \sim e^{-t/\tau} \tag{32}$$

This exponential behavior can be justified on the basis of the dynamics described by the Markov chain as in Eq. (17). Consider a vector $p(0)$ describing the initial configuration at time $t = 0$. We can write it as a linear combination of eigenvectors of P as $p = \sum_k c_k v^{(k)}$. The state vector at a later time $t = n\Delta t$ is obtained by a product with P^n as

$$p(t) = P^n p(0) = \sum_k c_k \lambda_k^n v^{(k)} \tag{33}$$

where λ_k are the associated eigenvalues. As we know the eigenvalue associated to the stationary state $v^{(1)}$ is $\lambda_1 = 1$.

Given an observable m (as the magnetization in the Ising model), its average over a state given by the vector p can be written as

$$m = \sum_\mu m_\mu p_\mu = M \cdot p \tag{34}$$

where m_μ is the value of the observable in the state μ, which we formally wrote as a scalar product with M. Now using (33) we find

$$m(t) = M \cdot v^{(1)} + \sum_{k>1} c_k \lambda_k^n M \cdot v^{(k)} = \langle m \rangle + \sum_{k>1} c_k \lambda_k^n M_k \tag{35}$$

where we have used the fact that the magnetization in the stationary state equals $\langle m \rangle$ and we defined $M_k = M \cdot v^{(k)}$. Recalling that $n = t/\Delta t$ the previous relation identifies the relaxation times

$$\tau_k = -\frac{1}{\log |\lambda_k|} \tag{36}$$

from which follows

$$m(t) - \langle m \rangle = \sum_{k>1} c_k M_k e^{-t/\tau_k} \tag{37}$$

[7]Although we discuss here the Ising model and the properties of the magnetization, Eq. (31) is more general and can be defined for any physical quantity

Using this expression we find for the autocorrelation function

$$\chi(t) = Ae^{-t/\tau_1} + Be^{-t/\tau_2} + \dots \tag{38}$$

which, in its leading behavior, reduces to an exponential decay as anticipated in Eq. (32). The form given in Eq. (38) may be a more appropriate fitting in typical simulation results.

Example 7
To complete the calculation of the Example 2 we add the two other eigenvectors $v^{(2)} = (-b, b-a, a)$ and $v^{(3)} = (0, 1, -1)$. The stationary state was $v^{(1)} = (0, 1/2, 1/2)$. Note that the sum of the elements of the vectors $v^{(2)}$ and $v^{(3)}$ is zero. This is a general property of any stochastic matrix, can you prove it?
Suppose now that the system is initially in $p = (1/3, 1/3, 1/3)$. The linear combination

$$p = \sum_{k=1}^{3} c_k v^{(k)} = v^{(1)} - \frac{1}{3b} v^{(2)} + \left(\frac{1}{6} - \frac{a}{3b}\right) v^{(3)} \tag{39}$$

The fact that $c_1 = 1$ is a general property of any stochastic matrix. Can you prove it?

In principle, one should use in the determination of averages as in Eq. (14) only data which are taken at time intervals larger than the autocorrelation time τ. In practice this is not always possible as it would reduce sensibly the number of data points. One good practice is to use data taken at time intervals $\Delta\mathcal{T}$; the correlation implies that the error estimates of the data averages depend on the correlation time τ. For instance for uncorrelated data, of say magnetizations, the standard deviation on the mean of n experimental points, is given by [8]

$$\sigma = \sqrt{\frac{1}{n-1}\left(\bar{m^2} - \bar{m}^2\right)} \tag{40}$$

It can be shown[9] that for measurments taken at intervals $\Delta\mathcal{T}$ one has

$$\sigma = \sqrt{\frac{1 + 2\tau/\Delta\mathcal{T}}{n-1}\left(\bar{m^2} - \bar{m}^2\right)} \tag{41}$$

which reduces to Eq. (40) in the limit $\Delta\mathcal{T} \gg \tau$.

[8] For more details see: M. Newman and G. T. Barkema, *Monte Carlo Methods in Statistical Physics*, Oxford University Press (1999).
[9] Müller-Krumbhaar and Binder, *J. Stat. Phys.* **8**, 1 (1973)

2.7 Critical slowing down: the dynamical exponent

In the vicinity of the critical point in a system in equilibrium various quantities are characterized by power law singularities which are governed by some critical exponents. For instance in the Ising model the magnetization vanishes, by approaching the critical temperature from below as:

$$m \sim \left(\frac{T_c - T}{T_c} \right)^{\beta} \tag{42}$$

The specific heat is also diverging as

$$c \sim \left| \frac{T - T_c}{T_c} \right|^{-\alpha} \tag{43}$$

Two spins at a distance $\vec{r}$ are correlated for temperatures above and below T_c as follows:

$$\langle s_{\vec{\delta}} s_{\vec{\delta}+\vec{r}} \rangle - \langle s_{\vec{\delta}} \rangle \langle s_{\vec{\delta}+\vec{r}} \rangle \sim e^{-|\vec{r}|/\xi} \tag{44}$$

where ξ is the correlation length. By approaching T_c this quantity diverges as:

$$\xi \sim \left| \frac{T - T_c}{T_c} \right|^{-\nu} \tag{45}$$

The values of α, β ν and γ (for the magnetic susceptibility) are *universal*, i.e. they depend only on the dimensionality and not, for instance, on the type of lattice considered (e.g. triangular, squared ...). In the 2d Ising model one has $\alpha = 0$ (which corresponds to a logarithmic singularity), $\beta = 1/8$, $\nu = 1$ and $\gamma = 7/4$.

The correlation length ξ is the typical size of cluster of spins pointing towards the same direction (see Fig. 4). As these regions grow in size in the vicinity of T_c, the Metropolis Monte Carlo algorithm suffers from a so-called critical slowing down, i.e. it becomes increasingly difficult to flip a spin at random as this is likely to be coupled to neighboring spins pointing to the same direction. Hence also the correlation time diverges when approaching the critical point as follows:

$$\tau \sim \left| \frac{T - T_c}{T_c} \right|^{-z\nu} \tag{46}$$

a relation which defines a new exponent z called the *dynamical* exponent. This exponent is algorithm dependent. It tells us how fast/slow the correlation time grows in the vicinity of the critical point. An algorithm with optimal performance near T_c would have a small value of z.

In order to calculate z from simulation one usually sets the temperature to the critical point $T = T_c$ (note T_c is known exactly for the 2d Ising model). We note that in a finite system of size L, the correlation length cannot diverge but is bounded by the system size i.e. $\xi \sim L$. Using this result and combining Eq. (45) with Eq. (46) we find

$$\tau \sim \xi^z \sim L^z \tag{47}$$

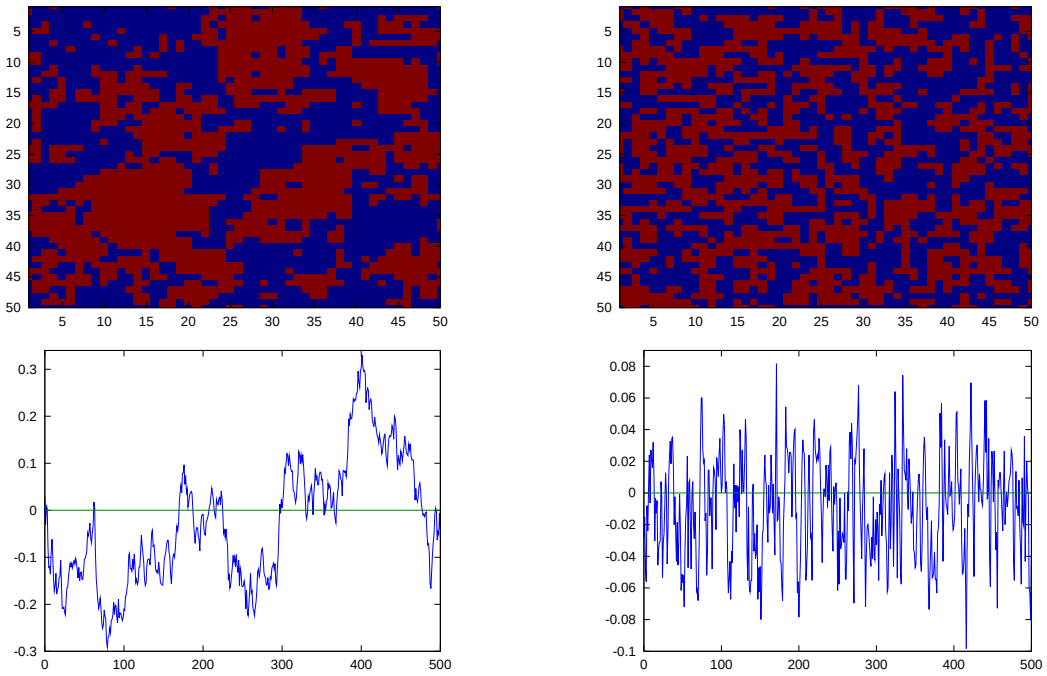

Figure 4: Typical spin configurations (top) and magnetization vs. time plots (bottom) for the Metropolis algorithm for the Ising model. Left: $T = 2.5$, Right: $T = 5$. Spins arrange in clusters which are bigger close to the critical temperature $T_c \approx 2.26$. Note also an increase of the autocorrelation time in the vicinity of T_c.

Numerical simulations for the Metropolis algorithm for the 2d Ising model give $z \approx 2.17$. Note that the critical slowing down effect is somehow visible also from the run shown in Fig. 3: fluctuations around the average magnetization value are correlated for longer times at $T = 0.98T_c$ compared with the case $T = 0.833T_c$.

2.8 Cluster algorithms: the Wolff algorithm for the Ising model

We present here an algorithm, which was introduced by U. Wolff in 1989 [10] and which basically solves the problem of critical slowing down. Differently from the Metropolis case seen above the Wolff algorithm is a *cluster* algorithm, i.e. a whole cluster of spin is flipped in each move instead of a single one. There are other cluster algorithms, which however will not be discussed here. For more details see the book by Newman and Barkema (see footnote 8).

In the Wolff algorithm a cluster is constructed as follows: take a starting seed spin at random and look at its neighboring spins. Those with the same sign as the seed spin are added to the cluster with a probability $P_{\rm add}$, while they are excluded from the cluster with probability $1 - P_{\rm add}$ (opposite spins than the seed spin are ignored). Our main preoccupation is the fulfillment of the detailed balance condition. Let us

[10] U. Wolff, Phys. Rev. Lett. **62**, 361 (1989).

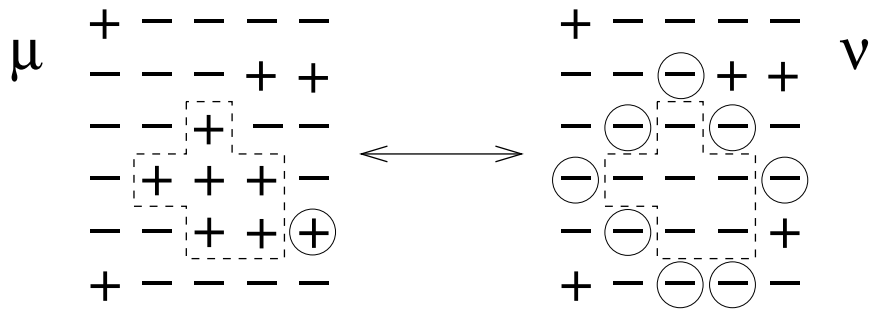

Figure 5: Cluster flipping in the Wolff algorithm. A cluster is composed by a connected set of identical spin; its boundary is indicated as a dashed line in the example. The spins encircled in both configurations are not included in the cluster.

consider two states μ and ν which differ from the flip of a single cluster. Apart from common prefactors we have for the selection probabilities $g(\mu \to \nu) \propto (1 - P_{\rm add})^m$ and $g(\nu \to \mu) \propto (1 - P_{\rm add})^n$, where m and n count the number of bonds between a spin in the cluster and an equal spin outside the cluster (in the example of Fig. 5 one has $m = 1$ and $n = 11$).

The detailed balance condition reads:

$$\frac{P(\nu \to \mu)}{P(\mu \to \nu)} = \frac{g(\nu \to \mu)\ A(\nu \to \mu)}{g(\mu \to \nu)\ A(\mu \to \nu)} = (1 - P_{\rm add})^{n-m} \frac{A(\nu \to \mu)}{A(\mu \to \nu)} = e^{-\beta(E_\mu - E_\nu)} \tag{48}$$

For the Ising model the energy difference is $E_\mu - E_\nu = (n-m)2J$ hence one can rewrite the detailed balance condition as

$$\frac{A(\nu \to \mu)}{A(\mu \to \nu)} = \left[(1 - P_{\rm add})e^{2\beta J}\right]^{m-n} \tag{49}$$

Now if we choose $P_{\rm add}$ as follows:

$$P_{\rm add} = 1 - e^{-2\beta J} \tag{50}$$

then the right hand side of Eq. (49) becomes identically equal to 1. In this case we can choose the acceptance ratios always equal to one for any transition $A(\mu \to \nu) = 1$, i.e. the cluster flipping is always accepted. Note that Eq. (50) implies that spins are added to the cluster with a probability that is temperature dependent. At high temperatures $\beta \to 0$ one has $P_{\rm add} \to 0$ i.e. clusters are very small, comprising typically only the seed spin. In this limit the Wolff algorithm becomes a single spin flipping algorithm. In the limit of very low temperatures $\beta \to \infty$ one has $P_{\rm add} \to 1$. Clusters become as large as the whole lattice. At each time step the system flips almost all spins swapping between states with positive and negative magnetizations. Both in the low and high temperature phases the Wolff algorithm is not very efficient and the simple Metropolis algorithm works better.

A final remark concerning the Wolff algorithm is the unit of time. One could take a cluster flip τ_{flip} as a unit of time, but since the size of the Wolff cluster is different

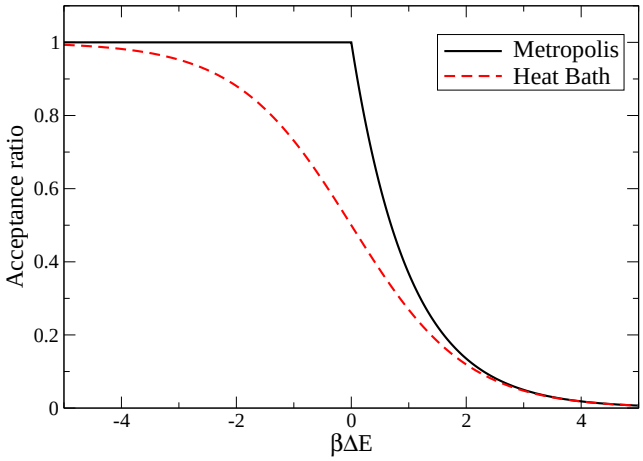

Figure 6: Acceptance ratios as functions of $\beta\Delta E$, where β is the inverse temperature and ΔE the energy difference, for the Metropolis (solid line) and Heat Bath algorithms (dashed line).

at low and high temperatures this is not a good time unit. In a d dimensional lattice model of size L, which contains then L^d spins a more appropriate time unit is

$$\tau = \tau_{flip} \frac{\langle n \rangle}{L^d} \tag{51}$$

where $\langle n \rangle$ is the average number of spins in the cluster. This time unit is directly related to that of the Metropolis algorithm. For instance, at low temperatures the clusters have the whole lattice size, i.e. $\langle n \rangle \to L^d$. Hence $\tau \approx \tau_{flip}$, as a single cluster flip corresponds to a sweep of the whole lattice. When the time unit of Eq. (51) is used one finds that the correlation time at the critical point scales as a power-law as a function of the lattice size, as in Eq. (47). For the two-dimensional Ising model one finds $z \approx 0.25$. This shows that the Wolff algorithm solves the problem of the critical slowing down.

2.9 Other single flip algorithms: the heat bath algorithm

The Metropolis algorithm is not the only single spin flip algorithm possible. We can still impose the condition of detailed balance (Eq. (24)) in many other possible ways. One possibility is given by the heat bath algorithm which for the Ising model has the following form:

$$A(\mu \to \nu) = \frac{e^{-\beta E_\nu}}{e^{-\beta E_\nu} + e^{-\beta E_\mu}} = \frac{e^{-\beta \Delta E}}{e^{-\beta \Delta E} + 1} \tag{52}$$

where we have defined $\Delta E = E_\nu - E_\mu$. This acceptance ratio matches that of the Metropolis algorithm (Eq. (29)) in the two limiting cases $\beta\Delta E \to -\infty$, where $A \to 1$ and $\beta\Delta E$ large and positive which gives $A \approx e^{-\beta\Delta E}$. The difference between heat bath and Metropolis is clear at $\Delta E = 0$. In the Metropolis case $A = 1$, while in the heat

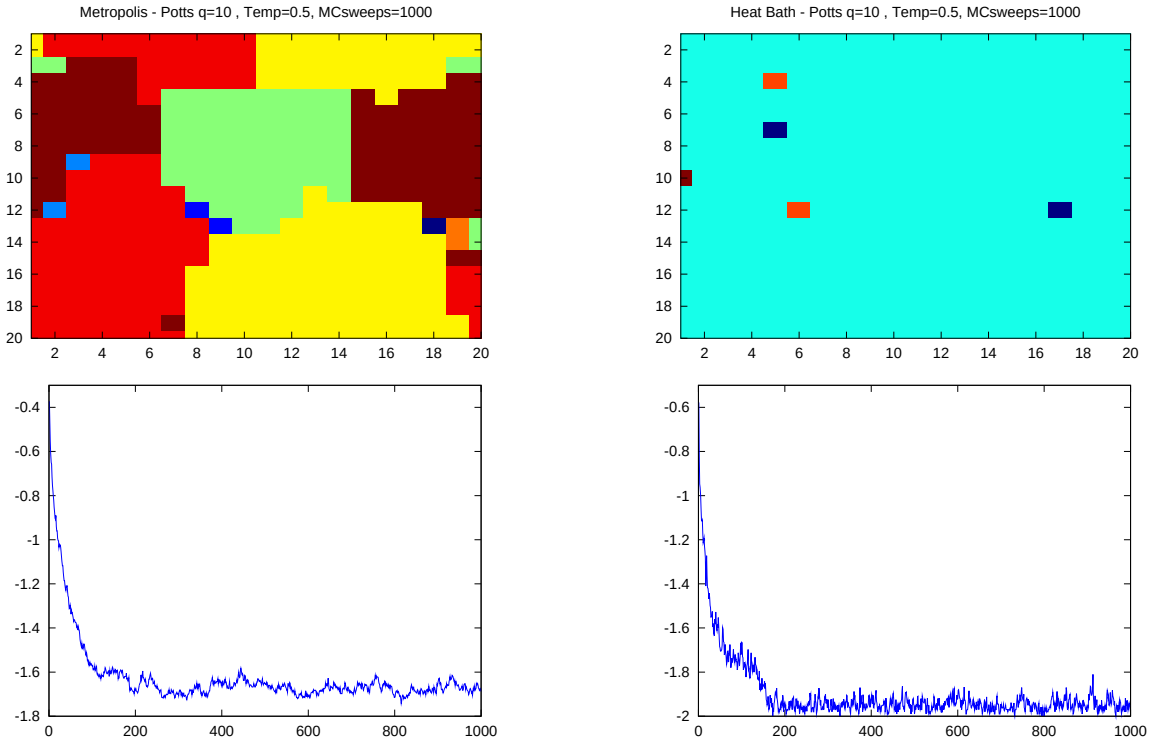

Figure 7: (Top) Configurations of a 20×20 10-state Potts model evolving according to the Metropolis algorithm (left column) and Heat Bath algorithm (right column), after 10^3 Monte Carlo sweeps. The Heat Bath algorithm is more efficient than Metropolis for large values of q (faster equilibration), as seen from the energy vs. time plots (bottom). Here $T = 0.5$ corresponds to a very low temperature.

bath $A = 1/2$. In other words a flip that does no vary energy is always accepted in the Metropolis move, while it is accepted only half of the time in the Heat Bath algorithm.

The Heath Bath algorithm is not very efficient and should not be used in Monte Carlo simulation of the Ising model, however it turns out to be a good algorithm for other models. An example is the q-state Potts model. This model (see Appendix A) is a generalization of the Ising model in which the spins can take q different values, as opposed to the Ising case in which $s_i = \pm 1$. We do not enter in the details on how to define the Heat Bath algorithm for the q-state Potts model (see problems session). This is a generalization of Eq. (52) to a q state case. Figure 7 shows some snapshots of configurations of the $q = 10$ case evolving according to the Metropolis algorithm and to the Heat Bath algorithm. The simulated temperature is very low, so the equilibrium configuration almost all spins are aligned to one of the q-values. Figure 7 shows that the Heat Bath algorithm evolves more rapidly towards the configuration with all aligned spins.

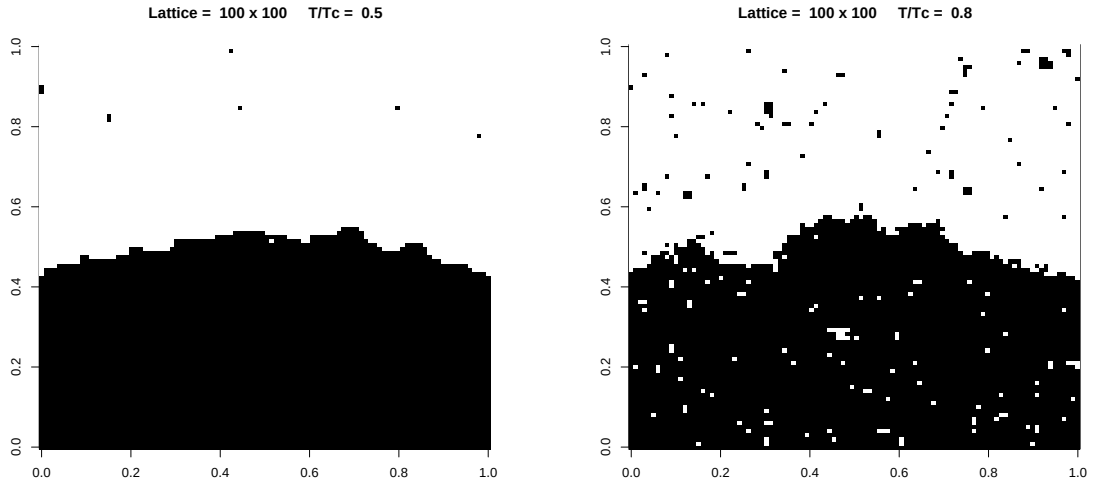

Figure 8: Typical interface configurations in the Ising model with non-local Kawasaki dynamics at two different temperatures: left $T = 0.5\ T_c$, right $T = 0.8\ T_c$.

2.10 Conserved order parameter Ising model

There are some situations in which one would like to study the properties of interfaces between two phases. The Ising model can be useful for that purpose as well. Below the critical temperature $T < T_c$ it has two ferromagnetic phases with average magnetization $\pm m(T)$. In the limit $T \to 0$ one has $m(T) \to 1$. In the example shown in Fig. 3 the system is prepared in a random configuration with average magnetization zero and then evolves towards a positive magnetization. The Metropolis algorithm could have acually chosen the negative magnetization state, with 50% chance, but the system ends up to be dominated by a single phase. If we divide the lattice in two identical symmetrical parts and place spins up on one side and spins down in the other, the single spin flip algorithm will finally select on of the two phases.

To have a stable interface we need something different than the spin flip algorithm. For this purpose it is more adapted the *Kawasaki* algorithm. The principles are very similar as the spin flip, with the only difference that the Kawasaki algorithm select pairs of neighboring spins and swaps them. The acceptance ratio is taken to follow the standard Metropolis rule of Eq. (29). As move consists in interchanging two spins with each other, then the total magnetization

$$M = \sum_i s_i \tag{53}$$

is conserved. Under the Kawasaki dynamics spins move diffusively through the lattice. Hence, it may take a long time before equilibration is reached. A more efficient version of the algorithm is the non-local version in which spins far away are swapped. In this implementation one creates a list of locations up spins and a list of locations of down spins. One then selects a spin from the up list and a spin of the down list, with a

probability independent of their physical position in the lattice. The acceptance ratio is still given by Eq. (29). The non-local moves imply a faster equilibration. Moreover by selecting at each step a plus and minus spin one avoids pairs of equal spins which only leads to a waste of computer time.

2.11 Monte Carlo in continuous time

In some systems we may have defined a Markov chain with the properties that the system remains locked into certain states for many time steps:

$$\mu \to \gamma \to \gamma \to \ldots \to \gamma \to \nu \ldots \tag{54}$$

which gives a very slow dynamics to the system. We could circumvent this problem by computing T_γ, the average time spent into a certain state γ and impose to the system to evolve into a different state. This is what the continuous time Monte Carlo method does. How do we calculate T_γ? Let $p = P(\gamma \to \gamma)$ and $q = 1 - p = \sum_{\nu \neq \gamma} P(\gamma \to \nu)$ (we have used the normalization of Eq. (16)). One then has:

$$T_\gamma = \Delta t \sum_{n=0}^{+\infty} n p^n \, q = \Delta t \, \frac{p}{q} \approx \frac{\Delta t}{\sum_{\nu \neq \gamma} P(\gamma \to \nu)} \tag{55}$$

where Δt is the discrete time unit of the Markov chain and where we have used $q \ll p \approx 1$. The second step is to select the new state, which is chosen according to the probability

$$g(\gamma \to \mu) = \frac{P(\gamma \to \mu)}{\sum_{\nu \neq \gamma} P(\gamma \to \nu)} \tag{56}$$

The disadvantage of this method is that for each given state γ one needs to compute the transition probabilities $P(\gamma \to \nu)$ for all possible ν, in order to compute the quantities in Eqs. (55) and (56). In some cases, with an appropriate choice of the transition probabilities, the continuous time method can be implemented in an efficient way. An example of an implementation for the conserved order parameter Ising model can be found in the book of Newman and Barkema (see footnote 8).

2.12 Continuous systems: Lennard-Jones fluids

All we have discussed for the Ising model can be applied to other types of systems. We discuss briefly here Monte Carlo simulations of classical fluids, in particular Lennard-Jones fluids. These systems of N particles occuping a volume V and at temperature T, subject to a pairwise interaction potential of the type:

$$V_{LJ}(r) = 4\varepsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^{6} \right] \tag{57}$$

where ε is the depth of the minimum of the potential and σ a characteristic distance at which the potential is zero. This potential contains two terms: a short range repulsive force diverging at short distances as $1/r^{12}$ and a long-range attraction decaying as $1/r^6$.

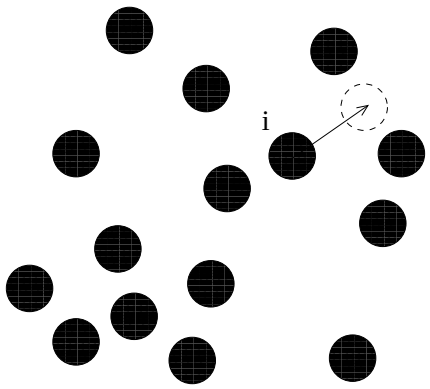

Figure 9: In the Monte Carlo algorithm for a continuous fluid system a particle is selected at random and a move to a new position is attempted. The new configuration is accepted according to the Metropolis rule.

The latter describes dispersive van der Waals forces due to fluctuating dipoles. The short range term is due to Pauli exclusion principle. The term $1/r^{12}$ was just chosen because of ease of the computation (it is the square of the attraction term).

The partition function of the system is expressed as an integral over positions $\vec{r}_i$ and momenta $\vec{p}_i$ of all particles $i = 1, 2, \ldots N$ as:

$$Z = \int d\vec{r}_1 \ldots d\vec{r}_N d\vec{p}_1 \ldots d\vec{p}_N \;\; e^{-\beta\left(\sum_i \frac{p_i^2}{2m} + V(\vec{r}_1 \ldots \vec{r}_N)\right)} \tag{58}$$

where the quantity between parenthesis in exponential is the total energy of the system, the sum of kinetic and potential energies. The average value of an observable A, which is a function of position and momenta ($A(\vec{r}_1 \ldots \vec{r}_N, \vec{p}_1 \ldots \vec{p}_N)$) takes the form:

$$\langle A \rangle = \frac{1}{Z} \int d\vec{r}_1 \ldots d\vec{r}_N d\vec{p}_1 \ldots d\vec{p}_N \;\; A(\vec{r}_1 \ldots \vec{r}_N, \vec{p}_1 \ldots \vec{p}_N) \; e^{-\beta\left(\sum_i \frac{p_i^2}{2m} + V(\vec{r}_1 \ldots \vec{r}_N)\right)} \tag{59}$$

Note that this equation is the continuous conterpart of Eq. (10). We consider observables which are function of either position or momenta. The functions of momenta are easy to deal with since they can be integrated analitically as these are gaussian integrals. The computations of position-dependent quantities cannot be done analytically and this is where numerical simulations as the Monte Carlo method are of interest.

In the case of a Lennard-Jones fluid the potential is a sum of pair potentials

$$V(\vec{r}_1 \ldots \vec{r}_N) = \sum_{i<j} V_{LJ}(r_{ij}) \tag{60}$$

where $r_{ij} = |\vec{r}_i - \vec{r}_j|$. This however does not make computations easier. For instance, to compute the average distance between two given particles $\langle |\vec{r}_1 - \vec{r}_2| \rangle$ one has to integrate over terms containing exponential of the type $\exp(-(V_{LJ}(r_{12}) + V_{LJ}(r_{13}) + V_{LJ}(r_{14}) + \ldots)/k_B T)$, which cannot be computed exactly.

A Monte Carlo algorithm for this system works as follows. A particle is selected at random and a move is attempted $\vec{r}_i \to \vec{r}\,'_i$ (see Fig. 9). This is obtained by selecting a shift of the coordinates δ_x, δ_y and δ_z from a uniform distribution on $[-\Delta, \Delta]$. The value of Δ is chosen to be of the order of the interparticles distance. To compute the acceptance ratio one needs the energy difference between new and old configuration. If the particle i is moved then the energy difference is

$$\Delta E = \sum_{j \neq i} \left[V_{LJ}(r'_{ij}) - V_{LJ}(r_{ij}) \right] \tag{61}$$

where $r'_{ij} = |\vec{r}\,'_i - \vec{r}_j|$. The acceptance ratio is as in the standard Metropolis case: $A(\vec{r}_i \to \vec{r}\,'_i) = 1$ if $\Delta E \leq 0$ and $A(\vec{r}_i \to \vec{r}\,'_i) = \exp(-\Delta E / k_B T)$ if $\Delta E > 0$.

One of the difficulties of this procedure is that the interaction is long range. The total potential energy is obtained by summing over all pairs of particles, which means a sum of $N(N-1)/2$ terms. This is very large in a simulation with a large number of particles. This is why the range of the interaction is often truncated. One simulates for instance the modified potential

$$V_{\text{trunc}}(r) = \begin{cases} V_{LJ}(r) & \text{if } r < r_c \\ 0 & \text{otherwise} \end{cases} \tag{62}$$

where r_c is the truncation radius (typically $r_c \approx 2 - 3\sigma$).

So given a particle, only those at a distance smaller than r_c from it need to be considered. This can be practically implemented by forming a list of all particles which are within a radius r_c from the given particle. There is one list for every particle, which needs to be updated at any accepted move. Apart from these technical difference is therefore the algorithm very similar to that introduced for the Ising model.

3 Out of equilibrium systems

In Monte Carlo simulations for equilibrium processess the two basic requirements that an algorithm has to fulfill are (1) Ergodicity and (2) Detailed Balance. As seen in the Ising model there are many algorithms for which these conditions are both satisfied; the algorithms can be of single spin flip (Metropolis, heath bath ...) or cluster type (Wolff algorithm ...). In non-equilibrium simulations there is unfortunately not such a large choice of algorithms. First of all we need to build up a Markov chain characterized by some transition probabilities for which $P(\mu \to \nu)$, where Δt has the significance of a real time and not of an algorithmic time any longer. For instance in the Ising model the cluster algorithms are of no use when studying dynamical properties: it is not a realistic dynamics one which swaps large clusters at low temperatures and very small ones at high temperatures. The non-local Kawasaki dynamics is equally non-physical as it involves the swapping of opposite spins with rates which do not depend on their physical distance. In conclusion: when studying dynamical properties one has to restict to local algorithms. Differently from equilibrium statistical mechanics, the non-equilibrium systems do not have such a developed formalism which relies on partition

functions, free energies etc ..., hence each system has to be studied separately. In what follows we present an example of Monte Carlo simulations applied to non-equilibrium systems.

3.1 Coupled chemical reactions

In many processes in physics and chemistry one deals with systems which can be described as coupled chemical reactions.

In a most general framework the chemical reactions can be written as

$$m_1A_1 + m_2A_2 + m_3A_3 + \ldots + m_rA_r \xrightarrow{k_i} n_1A_1 + n_2A_2 + n_3A_3 + \ldots + n_pA_p \qquad (63)$$

with m_i, n_i stochiometric coefficients and A_i different types of particles. The reaction rate is defined as follows: $k_i\dot{t}$ is the probability that a given r-uple of particles will react according to reaction Eq. (63) in a time interval $[t, t+dt]$. Note that a given reaction can be reversible, with a forward rate and a possibly different backward rate. In general one needs not to attach to the "particles" the significance of chemical molecules or atoms. The next example, for instance, is a basic model for population dynamics, where A_1 and A_2 can be view as preys and predators.

Example 8
The Lotka-Volterra model, also known as *predator-prey* model, is defined as follows. Consider two species A_1 (the prey) and A_2 (the predator) undergoing the following reactions:

$$A_1 \xrightarrow{k_1} 2A_1 \qquad (64)$$

$$A_1 + A_2 \xrightarrow{k_2} 2A_2 \qquad (65)$$

$$A_2 \xrightarrow{k_3} \emptyset \qquad (66)$$

where k_i, with $i = 1, 2, 3$ are the rates for each reaction.

Let us suppose we have in our system N different types of particles A_1, A_2 ... A_N. We assume the system to be at all times homogenous, so that spatial effects can be neglected. This happens if diffusion is sufficiently fast to efficiently mix the particles. Under this assumption a state of the system at time t is given by the N non-negative integers $(X_1, X_2, \ldots X_N)$. Let us also suppose that there are in total M possible chemical reactions occurring in the system, each characterized by a rate k_i with $i = 1, 2 \ldots M$. Let $P(X_1, X_2, \ldots X_N, t)$ the probability of finding the system on a state $(X_1, X_2, \ldots X_N)$ at time t.

Our aim is to set up a Master equation for P. Let us consider a reaction as Eq. (64), if at time t there are X_1 particles of type A_1 the total probability that the reaction 1 occurs in $[t, t + dt]$ is equal to X_1k_1dt. For the reaction of Eq. (65) one has $X_1X_2k_2dt$. The actual rate for a reaction is

$$a_i = k_i\{\text{number of reagents combinations for reaction } i\} \qquad (67)$$

Hence the Master equation takes the form

$$\frac{\partial}{\partial t}\ P(X_1, X_2, \dots X_N, t) = -\sum_{i=1}^{M} a_i\ P(X_1, X_2, \dots X_N, t) + \sum_{i=1}^{M} B_i \tag{68}$$

where as discussed in Appendix B the two terms in the right hand side of the equation are the loss and gain terms, respectively. In order to simplify the equation somewhat we have introduce a shorthand B_i for the gain terms. If reaction l is of the type $A_j \to A_i$ one has:

$$B_l = k_l X_j P(X_1, X_2, \dots, X_j + 1, \dots, X_i - 1, \dots X_N, t) \tag{69}$$

3.1.1 Rate equations

The rate equations describe the time evolution of the average number of particles. starting from the Master equation one can derive an equation for the averages:

$$\langle X_l \rangle = \sum_{\{X\}} X_l P(X_1, X_2, \dots X_N, t) \tag{70}$$

with $l = 1, 2 \dots N$. The problem is that in general one cannot deduce equations in closed form. Due to the presence of products of the particle numbers X_k in the rates a_i, see Eq. (67), one obtains equations of the type

$$\frac{d}{dt}\langle X_l \rangle = g_l\left(\langle X_1 \rangle, \langle X_2 \rangle, \dots, \langle X_N \rangle \dots, \langle X_i X_j \rangle, \dots, \langle X_k X_p X_q \rangle, \dots\right) \tag{71}$$

Neglecting correlation, i.e. approximating $\langle X_i X_j \dots X_k \rangle \approx \langle X_i \rangle \langle X_j \rangle \dots \langle X_k \rangle$ one obtains differential equations in closed form. These equations are not exact, and describe only the evolution of the average number of particles, in a deterministic way. They do not describe fluctuations around these averages. If the number of all particles is large, i.e. $\langle X_i \rangle \gg 1$, the rate equations are expected to provide an accurate description of the evolution of the system. If the number of particles is low, the fluctuation may dominate and the rate equations description is no longer fulfilled.

Example 9
The rate equations associated to the Lotka-Volterra system are:

$$\frac{d}{dt}X_1 = k_1 X_1 - k_2 X_1 X_2 \tag{72}$$

$$\frac{d}{dt}X_2 = k_2 X_1 X_2 - k_3 X_2 \tag{73}$$

where to simplify the notation we have omitted the parenthesis $\langle . \rangle$.
There are two stationary solutions (i.e. time independent) $X_1 = X_2 = 0$ and $X_1 = k_3/k_2$, $X_2 = k_1/k_2$. The system has oscillatoric solutions. To see this we expand the solution around the stationary point $X_1 = \varepsilon + k_3/k_2$ and $X_2 = \delta + k_1/k_2$, using an expansion to linear orders in ε and δ. We find $\varepsilon(t) = \varepsilon_0 \cos(\omega t + \phi)$, $\delta(t) = \varepsilon_0 \sqrt{k_1/k_3} \sin(\omega t + \phi)$.

3.1.2 Stochastic simulations: The Gillespie algorithm

The Gillespie algorithm [11] provides a method to simulates exactly the Master equation.

Let us consider a configuration given as $(X_1, X_2 \dots X_N)$ at a time t. We ask ourselves what is the probability that the system remains in this state until a time $t+\tau$ and that the next reaction occurs in $[t+\tau, t+\tau+d\tau]$ and that the reaction is of type j. We denote this probability as $P(\tau, j)$. We can write:

$$P(\tau, j) = P_0(\tau)\ a_j d\tau \tag{74}$$

where $a_j d\tau$ is the probability that the reaction j occurs in in the interval $[t+\tau, t+\tau+d\tau]$, while $P_0(\tau)$ is the probability that no reactions occur in $[t, t+\tau]$.

It is simple to set up a differential equation for $P_0(\tau)$. We have

$$P_0(\tau + d\tau) = P_0(\tau) \left[1 - \sum_{i=1}^{M} a_i d\tau \right] \tag{75}$$

which can be easily integrated and with the initial condition $P_0(0) = 1$ yields:

$$P_0(\tau) = e^{-a_0 \tau} \tag{76}$$

where $a_0 = \sum_{i=1}^{M} a_i$. This quantity is the total rate that some of the M reactions occurr.

Gillespie Algorithm

1) Given that at time t the system is in the state $(X_1, X_2 \dots X_N)$. calculate the rates a_i for all M chemical reactions

2) Generate a random τ from the distribution $e^{-a_0 \tau}$ and update the time to $t = t + \tau$

3) Choose one of the M reactions using as probabilities a_i/a_0.

4) If the reaction chosen is j update the particle numbers $(X_1, X_2 \dots X_N)$ accordingly and go to step 1)

[11] Although it is 30 years old it is still worth reading the original paper: D. T. Gillespie, *Exact Stochastic Simulation of Coupled Chemical Reactions*, J. Phys. Chem. **81**, 2340 (1977).

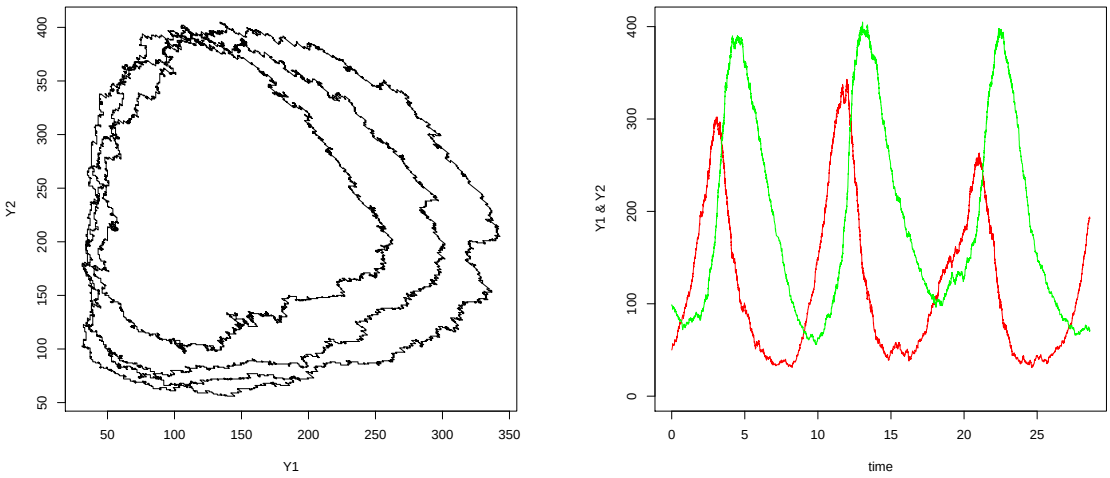

Figure 10: Simulation of the Lotka-Volterra model with the Gillespie algorithm, with $k_1 =$, $k_2 =$ and $k_3 =$. Left: trajectory in the plane Y_1, Y_2. Right: plot of $Y_1(t)$ and $Y_2(t)$.

A The Ising and Potts models

The Ising model is a cornerstone of classical statistical mechanics. It is sufficiently simple that can be handled analytically, through exact computations (in one and two dimensions), or with approximated methods. In particular in two and higher dimensions the model has a phase transition. The Ising model is defined on a lattice in d dimensions, whose sites are occupied by a spin taking two values $s = \pm 1$. The energy of a configuration is given by

$$E = -J \sum_{\langle ij \rangle} s_i s_j \tag{77}$$

where $J > 0$ is the coupling constant and where the previous sum is extended to nearest neighbors. This model has a transition from a low temperature ferromagnetic phase to a high temperature phase which is paramagnetic. The transition temperature for the two dimensional square lattice is known exactly and it is equal to

$$T_c = \frac{2J}{\ln(1+\sqrt{2})} \tag{78}$$

In the ferromagnetic phase the magnetization $m = \langle s_i \rangle \neq 0$ (the average value of a spin). It can be computed exactly for an infinite two dimensional square lattice and has the form

$$m(T) = \left[1 - \sinh^{-4}(2J/T)\right]^{1/8} \tag{79}$$

The magnetization vanishes by approaching T_c from below with some power-law singularity as

$$m \sim (T - T_c)^{\beta} \tag{80}$$

Dimension	α	β	γ	ν
2	0	1/8	7/4	1
3	0.12	0.31	1.25	0.64

Table 1: Summary table of the critical exponents of the Ising model in two and three dimensions. The exponents are associated to the divergences of the specific heat (α), magnetization (β), susceptibility (γ) and correlation length (ν).

Other quantities also show some power-law behavior. For instance the spin-spin correlation function for two spins separated by a distance r in the lattice is given by

$$\langle s_i s_j \rangle \sim e^{-r/\xi} \tag{81}$$

where ξ defines the correlation length, a quantity which diverges when approaching T_c as follows:

$$\xi \sim |T - T_c|^{-\nu} \tag{82}$$

The Ising model has been solved exactly in two dimensions and the critical exponents are given in terms of this exact solution. In three dimensions the value of the exponents is known approximately. Table 1 summarizes their values. The exponents are universal, they do not depend on the type of lattice (e.g. squared, triangular ...) but only on the dimensionality of the system.

The Potts models are generalizations of the Ising model. In the q-state model each spin can take q values $s_i = 0, 1, 2 \ldots q-1$, with an energy given by

$$E = -J \sum_{\langle ij \rangle} \delta_{s_i s_j} \tag{83}$$

and $J > 0$. Here the $\delta_{s_i s_j}$ is the discrete (Kronecker) delta function which is defined as $\delta_{mn} = 1$ if $m = n$ and $\delta_{mn} = 0$ otherwise. It is easy to show that the $q = 2$ model corresponds to the Ising model. We can show this by using the identity:

$$\delta_{mn} = \frac{1}{2}(2m-1)(2n-1) + \frac{1}{4} \tag{84}$$

which is valid for $m, n = 0, 1$. The transformation $\hat{s} = 2s - 1$, takes us from Potts spins $s = 0$, 1, to Ising spins $\hat{s} = 2s - 1 = \pm 1$. So we find $E_{\text{Potts}}(\{s_i\}) = \frac{1}{2} E_{\text{Ising}}(\{\hat{s}_i\}) - \frac{J}{4}$. The Potts model for any dimensions higher than two has a phase transition from a high temperature phase where the spins point to a random direction, to a "ferromagnetic" phase in which they predominantly align along one of the q values of the spins.

B The Master equation

Eq. (18) describes the stochastic evolution of the system evolving at discrete times Δt. By taking the limit $\Delta t \to 0$ we can derive the so-called Master equation in its standard form. This is a differential equation that can be deduced as follows:

$$p(t + \Delta t) - p(t) = (P_{\Delta t} - 1)\, p(t) \tag{85}$$

where 1 is the identity matrix. Dividing by Δt and taking the limit $\Delta t \to 0$ we obtain

$$\frac{d}{dt}p(t) = -Hp(t) \tag{86}$$

where we have introduced the matrix H as

$$H = \lim_{\Delta t \to 0} \frac{1 - P_{\Delta t}}{\Delta t} \tag{87}$$

As we have seen the sum of elements in the column of P is equal to one. Hence the columns of H add up to zero. Moreover the non-diagonal elements of H are:

$$H_{\mu\nu} = -h(\nu \to \mu) \tag{88}$$

where $h(\nu \to \mu)$ are transition rates from ν to μ, i.e. transition probabilities per unit time. We write now explicitely Eq. (86) in terms of matrix elements, separating the diagonal and non-diagonal terms:

$$\frac{d}{dt}p_\mu(t) = -\sum_{\nu \neq \mu} H_{\mu\nu}p_\nu(t) - H_{\mu\mu}p_\mu(t) \tag{89}$$

Using the fact that the sum of the elements on each column equals zero and using Eq. (88) we can rewrite it as

$$\frac{d}{dt}p_\mu(t) = \sum_{\nu \neq \mu} h(\nu \to \mu)p_\nu(t) - \sum_{\nu \neq \mu} h(\mu \to \nu)p_\mu(t) \tag{90}$$

which is the basic form of the Master equation. The two terms on the left hand side of this equation have an obvious interpretation as "gain" and "loss" terms.