An introduction to Monte Carlo methods

J.-C. Walter [a,*], G.T. Barkema [b,c]

[a] *Laboratoire Charles Coulomb UMR 5221 & CNRS, Université Montpellier 2, 34095 Montpellier, France*
[b] *Institute for Theoretical Physics, Utrecht University, The Netherlands*
[c] *Instituut-Lorentz, Universiteit Leiden, P.O. Box 9506, 2300 RA Leiden, The Netherlands*

HIGHLIGHTS

- We describe the main principles of Monte Carlo simulations.
- We illustrate the method with algorithms for the Ising model.
- We discuss dynamical effects such as thermalization and correlation time.

ARTICLE INFO

ABSTRACT

Monte Carlo simulations are methods for simulating statistical systems. The aim is to generate a representative ensemble of configurations to access thermodynamical quantities without the need to solve the system analytically or to perform an exact enumeration. The main principles of Monte Carlo simulations are ergodicity and detailed balance. The Ising model is a lattice spin system with nearest neighbor interactions that is appropriate to illustrate different examples of Monte Carlo simulations. It displays a second order phase transition between disordered (high temperature) and ordered (low temperature) phases, leading to different strategies of simulations. The Metropolis algorithm and the Glauber dynamics are efficient at high temperature. Close to the critical temperature, where the spins display long range correlations, cluster algorithms are more efficient. We introduce the rejection free (or continuous time) algorithm and describe in details an interesting alternative representation of the Ising model using graphs instead of spins with the so-called Worm algorithm. We conclude with an important discussion of the dynamical effects such as thermalization and correlation time.

1. Introduction

Most models in statistical physics are not solvable analytically, and therefore an alternative way is needed to determine thermodynamical quantities. Numerical simulations help in this task, but introduce another challenge: it is not possible, in most cases, to enumerate all the possible configurations of a system; one therefore has to create a set of configurations that are representative for the entire ensemble. In this section, we will illustrate our purpose with the Ising model. This is a renowned model because of its simplicity and success in the description of critical phenomena [1]. The degrees of freedom are spins $S_i = \pm 1$ placed at the vertex i of a lattice. This lattice will be square or cubic for simplicity, with edge size L and

---

* Corresponding author. Tel.: +33 673985696.
E-mail addresses: jean-charles.walter@univ-montp2.fr (J.-C. Walter), g.t.barkema@uu.nl (G.T. Barkema).

dimension D. Thus, the system contains $N = L^D$ spins. The Hamiltonian of the Ising model is:

$$\mathcal{H} = -J \sum_{\langle ij \rangle} S_i S_j, \tag{1}$$

where the summation runs over all pairs of nearest-neighbor spins $\langle ij \rangle$ of the lattice and J is the strength of the interaction. The statistical properties of the system are obtained from the *partition function*:

$$Z = \sum_{\mathcal{C}} e^{-\beta E(\mathcal{C})}, \tag{2}$$

where the summation runs over all the configurations $\mathcal{C}$. The energy of a configuration is denoted by $E(\mathcal{C})$. Here, $\beta \equiv 1/(k_B T)$ is the inverse temperature (temperature T and Boltzmann constant k_B). The Ising model displays a second-order phase transition at the temperature T_c, characterized by a high temperature phase with an average magnetization zero (disordered phase) and a low temperature phase with a non-zero average magnetization (ordered phase). The system is exactly solvable in one and two dimensions. For $D \geq 4$, the critical properties are easily obtained by the renormalization group. In three dimensions no exact solution is available. Even a $3D$ cubic lattice of very modest size $10 \times 10 \times 10$ generates $2^{1000} \approx 10^{301}$ configurations in the partition function. If we want to obtain e.g. critical exponents with a sufficient accuracy, we need sizes that are at least an order of magnitude larger. An exact enumeration is a hopeless effort. *Monte Carlo simulations* are one of the possible ways to perform a sampling of configurations. This sampling is made out of a set of configurations of the configuration space that contributes the most to the averages, without the need of generating every single configuration. This is referred to as *importance sampling*. In this sampling of the configuration space, it is important to choose the appropriate Monte Carlo scheme to reduce the computational time. In that respect, the Ising model is interesting because the different regimes in temperature lead to the development of new algorithms that reduce tremendously the computational time, specifically close to the critical temperature.

We will start these notes by introducing two important principles of Monte Carlo simulations: detailed balance and ergodicity. Then we will review different examples of Monte Carlo methods applied to the Ising model: local and cluster algorithms, the rejection free (or continuous time) algorithm, and another kind of Monte Carlo simulations based on an alternative representation of the spin system, namely the so-called Worm algorithm. We continue with discussing dynamical quantities, such as the thermalization and correlation times.

2. Principles of MC simulations: ergodicity and detailed balance condition

The basic idea of most Monte Carlo simulations is to iteratively propose a small random change in a configuration C_i, resulting in the trial configuration C_{i+1}^t (the index "t" stands for trial). Next, the trial configuration is either accepted, i.e. $C_{i+1} = C_{i+1}^t$, or rejected, i.e. $C_{i+1} = C_i$. The resulting set of configurations for $i = 1 \ldots M$ is known as a Markov chain in the configuration space of the system. We define $P_A(t)$ as the probability to find the system in the configuration A at the time t and $W(A \rightarrow B)$ the transition rate from the state A to the state B. This Markov process can be described by the master equation:

$$\frac{dP_A(t)}{dt} = \sum_{A \neq B} [P_B(t) W(B \rightarrow A) - P_A(t) W(A \rightarrow B)], \tag{3}$$

with the condition $W(A \rightarrow B) \geq 0$ and $\sum_B W(A \rightarrow B) = 1$ for all A and B. The transition probability $W(A \rightarrow B)$ can be further decomposed into a trial proposition probability $T(A \rightarrow B)$ and an acceptance probability $A(A \rightarrow B)$ so that $W(A \rightarrow B) = T(A \rightarrow B) \cdot A(A \rightarrow B)$. A proposed change in the configuration is usually referred to as a Monte Carlo move. Conventionally, the time scale in Monte Carlo simulations is chosen such that each degree of freedom of the system is proposed to change once per unit time, statistically.

The first constraint on this Markov chain is called *ergodicity*: starting from any configuration C_0 with nonzero Boltzmann weight, any other configuration with nonzero Boltzmann weight should be reachable through a finite number of Monte Carlo moves. This constraint is necessary for a proper sampling of the configuration space, as otherwise the Markov chain will be unable to access a part of configuration space with a nonzero contribution to the partition sum.

Apart from a very small number of peculiar algorithms, a second constraint is known as the *condition of detailed balance*. For every pair of states A and B, the probability to move from A to B, as well as the probability for the reverse move, are related via:

$$P_A \cdot T(A \rightarrow B) \cdot A(A \rightarrow B) = P_B \cdot T(B \rightarrow A) \cdot A(B \rightarrow A). \tag{4}$$

The meaning of this condition can be seen in Eq. (3): a stationary probability (i.e. $dP_A/dt = 0$) is reached if each individual term in the summation on the right hand side cancels. This prevents the Markov chain to be trapped in a limit cycle [2]. This is a strong, but not necessary, condition. We mention that generalizations of Monte Carlo process that do not satisfy detailed balance exist. The combination of ergodicity and detailed balance assures a correct algorithm, i.e., given a long enough time, the desired distribution probability is sampled.

The key question in Monte Carlo algorithms is which small changes one should propose, and what acceptance probabilities one should choose. The trial proposition and acceptance probabilities have to be well chosen so that the probability of

sampling of a configuration A (after thermalization) is equal to the *Boltzmann weight*:

$$P_A = \frac{e^{-\beta E_A}}{Z}, \tag{5}$$

in which E_A is the energy of configuration A. The knowledge of the partition function Z is not necessary because the transition probabilities are constructed with the ratio of probabilities. The detailed balance condition (4), using (5), can be rewritten as:

$$\frac{T(B \to A) \cdot A(B \to A)}{T(A \to B) \cdot A(A \to B)} = \frac{P_A}{P_B} = e^{-\beta(E_A - E_B)}. \tag{6}$$

3. Local MC algorithms: Metropolis and Glauber

One often-used approach to realize detailed balance is to propose randomly a small change in state A, resulting in another state B, in such a way that the reverse process (starting in B and then proposing a small change that results in A) is equally likely. More formally, a process in which the condition $T(A \to B) = T(B \to A)$ holds for all pairs of states $\{A, B\}$. For example, taking the example of an Ising model containing N spins, it corresponds to chose randomly one of the spins on the lattice, therefore $T(A \to B) = T(B \to A) = 1/N$. Detailed balance allows for a common scale factor in the acceptance probabilities for the forward and reverse Monte Carlo moves, but being probabilities, they cannot exceed 1. Simulations are then fastest if the largest of the two acceptance probabilities is equal to 1, i.e. either $A(A \to B)$ or $A(B \to A)$ is equal to 1. These conditions (including detailed balance) are realized by the so-called *Metropolis algorithm*, in which the acceptance probability is given by:

$$A_{\text{met}}(A \to B) = \min[1, P_B/P_A] = \min[1, \exp(-\beta(E_B - E_A))]. \tag{7}$$

Thus, a proposed move that does not raise the total energy is always accepted, but a proposed move which results in higher energy is accepted with a probability that decreases exponentially with the increase of the energy difference. For the sake of illustration, let us describe how a simulation of the Ising model looks like:

1. initialize all spins (either random or all up);
2. perform N random trial moves ($N = L^D$):
   (a) randomly select a site
   (b) compute the energy difference $\Delta E = E_B - E_A$ if the trial (here a spin flip) induces a change in energy
   (c) generate a random number **rn** uniformly distributed in $[0, 1]$
   (d) if $\Delta E < 0$ or if **rn** $< \exp(-\Delta E)$: flip the spin;
3. perform measurement of some observables.

The step 2 corresponds to one unit time step of the Monte Carlo simulation. An alternative to the Metropolis algorithm is the *Glauber dynamics* [3]. The trial probability is the same as Metropolis i.e. $T(A \to B) = T(B \to A) = 1/N$. However the acceptance probability is now:

$$A_{\text{gla}}(A \to B) = \frac{e^{-\beta(E_B - E_A)}}{1 + e^{-\beta(E_B - E_A)}}, \tag{8}$$

which also satisfies the detailed balance condition Eq. (6).

4. Cluster algorithms: the example of the Wolff algorithm

Many models encounter phase transitions at some critical temperature. The paradigmatic example for the second order phase transitions is the Ising model defined in Eq. (1). In the vicinity of the critical temperature, the spins display *critical fluctuations*. As shown in Fig. 1 (middle), large aligned spin domains appear. This phenomenon is associated with (i) the divergence of the correlation length ξ of the connected spin–spin correlation function $C(|i - j|) = \langle S_i \cdot S_j \rangle - \langle S_i \rangle^2$ (ii) the divergence of the correlation time of the autocorrelation function $C(|t - t'|) = \langle S_i(t) \cdot S_i(t') \rangle - \langle S_i(t) \rangle \langle S_i(t') \rangle$. Moreover, the correlation time increases with the size of the system like $\tau \sim L^{z_c}$ where z_c is the *critical dynamical exponent*. For the 2D Ising model simulated with the Metropolis algorithm, $z_c = 2.1665(12)$ [5]. This phenomenon of critical slowing down reflects the difficulty to change the magnetization of a correlated spin cluster. Take again the example of a 2D spin system where one spin has four nearest neighbors. If this spin is surrounded by aligned spins, its contribution to the energy is $E_A = -4J$ and after the reversal of this spin, this becomes $E_B = 4J$. Right at $T_c \approx 2.269$, the acceptance probability is low for the Metropolis algorithm: $A(A \to B) = e^{-8\beta_c J} = 0.0294\ldots$. Thus, most of the flipping attempts are rejected. Making matters worse, even if such a spin with aligned neighbors is flipped, the next time it is selected, it will surely flip back. Only spin flips at the edge of a cluster have a significant effect over a longer time; but their fraction becomes vanishingly small when the critical temperature is approached and the cluster size diverges.
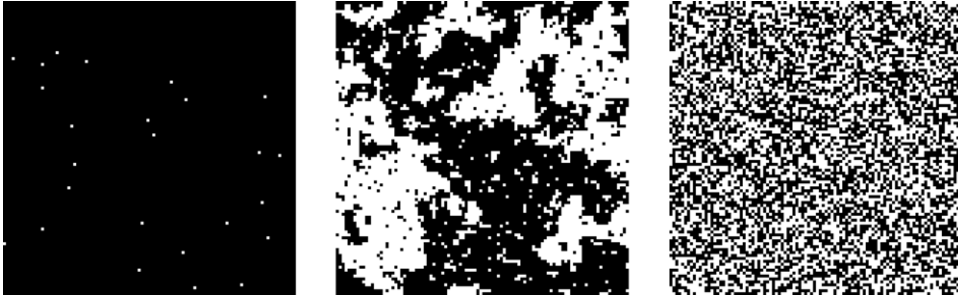
Fig. 1. Snapshots of the 2D Ising model defined in Eq. (1) at three different temperatures: from left to right, $T \ll T_c$, $T \approx T_c$ and $T \gg T_c$ where T_c is the critical temperature. White and blacks dots denote spins up and down. The system size is 200×200. At T_c (middle), we observe large clusters of correlated spins: these are *critical fluctuations* that slow down Monte Carlo simulations when local algorithms such as Metropolis or Glauber are used. This critical slowing down is reduced by non-local (or cluster) algorithms like the Wolff algorithm [4].
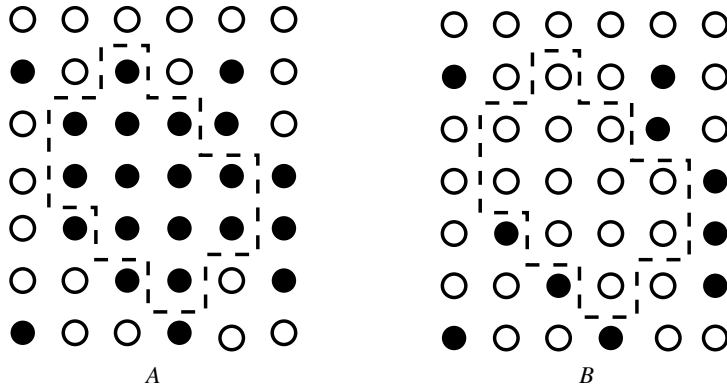

Fig. 2. Sketch of one iteration of the non-local algorithm introduced by Wolff [4] between two spin configurations *A* and *B*. The white and black dots stand for spins of opposite signs. The spins within the loop (dashed line) belong to the same cluster. The steps to form the cluster are: (i) choose randomly a seed spin; (ii) add aligned spins with the probability P_{add} (see text); (iii) add iteratively aligned neighbors of newly added spins with the probability P_{add}; (iv) flip all the spins in the cluster at once when the cluster is completed. This is an efficient algorithm for the Ising model at criticality.

One remedy is to develop a non-local algorithm that flips a whole cluster of spins at once. Such an algorithm has been designed for the Ising model by Wolff [4], following the idea of Swendsen and Wang [6] for more general spin systems. A sketch of this procedure is shown in Fig. 2.

The procedure consists of first choosing a random initial site (seed site). Then, we add each neighboring spin, provided it is aligned, with the probability P_{add}. If it is not aligned, it cannot belong to the cluster. This step is iteratively repeated with each neighbor added to the cluster. When no neighbor can be added to the cluster anymore, all the spins in the cluster flip at once. The probability to form a certain cluster of spins in state *A* before the Wolff move is the same as that in state *B* after the Wolff move, except for the aligned spins that have not been added to the cluster at the boundaries. The probability not to add an aligned spin is $1 - P_{\text{add}}$. If m and n stand for non-added aligned spins to the cluster for *A* and *B*, $T(A \rightarrow B)/T(B \rightarrow A) = (1 - P_{\text{add}})^{m-n}$ and the detailed balance condition (6) can be rewritten as:

$$\frac{T(A \rightarrow B) \cdot A(A \rightarrow B)}{T(B \rightarrow A) \cdot A(B \rightarrow A)} = (1 - P_{\text{add}})^{m-n} \frac{A(A \rightarrow B)}{A(B \rightarrow A)} = e^{-\beta(E_B - E_A)}. \tag{9}$$

Noticing that $E_A - E_B = 2J(n - m)$, it follows that:

$$\frac{A(A \rightarrow B)}{A(B \rightarrow A)} = \left[(1 - P_{\text{add}})e^{2\beta J} \right]^{n-m}. \tag{10}$$

Therefore, choosing $P_{\text{add}} = 1 - e^{-2\beta J}$, the acceptance probabilities simplifies: $A(A \rightarrow B) = A(B \rightarrow A) = 1$. For this reason the spins can be automatically flipped when the cluster is formed. In the vicinity of the critical point, the Wolff algorithm significantly reduces the autocorrelation time and the critical dynamical exponent compared to a local algorithm (such as Metropolis or Glauber). We notice that the time $\tilde{\tau}_W$ measured in units of Wolff iterations involves a subset of spins corresponding to the averaged size $\langle p \rangle$ of a cluster. On the other hand a time τ_M measured in units of Metropolis iterations involves all spins of the network i.e. $N = L^D$ spins. To compare the efficiency of both algorithms fairly, it is therefore necessary to define a rescaled Wolff autocorrelation time $\tau_W = \tilde{\tau}_W \langle p \rangle / L^D$. Moreover, $\langle p \rangle$ and the magnetic susceptibility χ are related via $\chi = \beta \langle p \rangle$ [2]. It is well known that χ behaves at criticality as $\chi \sim L^{\gamma/\nu}$ where γ and ν are critical exponents [1]. We also define critical

dynamical exponents $\tilde{z}_c^W$ and z_c^W for $\tilde{\tau}_W$ and τ_W, respectively. Assuming $\tilde{\tau}_W \sim L^{\tilde{z}_c^W}$, we obtain $\tau_W \sim L^{\tilde{z}_c^W + \gamma/\nu - D}$. It leads to the definition of the dynamical critical exponent $z_c^W = \tilde{z}_c^W + \gamma/\nu - D$. In 2D for example, remarkably, the dynamical exponent is $z_c^W \approx 0$ for the Wolff algorithm (see e.g. Refs. [7,8] and references therein) whereas $z_c^M = 2.1665(12)$ [5] for Metropolis.

5. Continuous-time or rejection free algorithm

As we have seen in the previous subsection, with a local algorithm (like e.g. Metropolis), a spin flip of the Ising model at criticality has a high probability to be rejected, and this holds even more in the ferromagnetic phase. A significant amount of the computational time will therefore be spent without making the system evolve. An alternative way has been proposed by Gillespie [9] in the context of chemical reactions and afterwards applied by Bortz, Kalos and Lebowitz in the context of spin systems [10].

This algorithm lists all possible Monte Carlo moves that can be performed in the system in its current configuration. One of these moves is chosen randomly according to its probability, and the system is forced to move into this state. The time step of evolution during such a move can be estimated rigorously. This time will change from each configuration and cannot be set to unity as in the Metropolis algorithm: it takes a continuous value. This is why this algorithm is sometimes called *continuous time algorithm*. On the one hand, this algorithm has to maintain a list of all possible moves, which requires a relatively heavy administrative task, on the other hand, the new configuration is always accepted and this saves a lot of time when the probability of rejection would otherwise be high. It is also sometimes called the *rejection free algorithm*. The efficiency of this algorithm will be maximized for $T \leq T_c$. In detail, one iteration of the continuous time algorithm looks like:

1. list all possible moves from the current configuration. Each of these n moves has an associated probability P_i;
2. calculate the integrated probability that a move occurs $Q = \sum_{i=1}^{n} P_i$;
3. generate a random number **rn₁** uniformly distributed in $[0, Q]$. This selects the chosen move with probability P_i/Q;
4. estimate the time elapsed during the move: $\Delta t = Q^{-1} \ln(1 - \mathbf{rn_2})$ where **rn₂** is a random number uniformly distributed in $[0, 1[$.[1]

Implementation of this algorithm becomes easier if the probabilities P_i can only take a small number of values. In that case, lists can be made of all moves with the same probability P_i. The selection process is then first to select one of the lists, with the appropriate probability, after which randomly one move is selected from that list. This is the case e.g. in Ising simulations on a square (2D) or cubic (3D) lattice, when the probability P_i is limited to the values 1, $e^{-4\beta J}$, $e^{-8\beta J}$, or $e^{-12\beta J}$ (the latter occurring only in 3D).

6. The Worm algorithm

We present here another example of a local algorithm, the so-called *Worm algorithm* introduced by Prokof'ev, Svistunov and Tupitsyn [15,16]. The difference with the algorithms presented above is an alternative representation of the system, in terms of graphs instead of spins. The Markov chain is therefore performed along graph configurations rather than spin configurations, but always with Metropolis acceptance rates. The principle is based on the high temperature expansion of the partition function. Suppose that we want to sample the magnetic susceptibility of the Ising model. We can access it via the correlation function using the (discrete) fluctuation–dissipation theorem:

$$\chi = \frac{\beta}{N} \sum_{i,j} G(i-j), \tag{11}$$

where $G(i-j) = \langle S_i \cdot S_j \rangle - \langle S_i \rangle^2$ is the connected correlation function between sites i and j. In the high temperature phase, the average value of the spin cancels and $G(i-j) = \langle S_i \cdot S_j \rangle$. The first step is to write the correlation function $G(i-j)$ of the Ising model in the following form:

$$G(i-j) = \frac{1}{Z} \sum_{\{S\}} S_i \cdot S_j \, e^{\beta J \sum_{\langle k,l \rangle} S_k \cdot S_l}, \tag{12}$$

$$= \frac{1}{Z} \cosh(\beta J)^{DN} \sum_{\{S\}} S_i \cdot S_j \prod_{\langle k,l \rangle} (1 + S_k \cdot S_l \tanh(\beta J)). \tag{13}$$

The configurations that contribute to the sum in (13) contain an even number of spins in the product at any given site. Other products involving an odd number of spins in the product contribute zero. Each term can be associated with a path determined by the sites involved in it. A contribution to the sum is made of a (open) path joining sites i and j and closed loops. The sum over the configurations can be replaced by a sum over such graphs. Fig. 3 sketches such a contribution for a given couple of source sites i and j. The importance sampling is no longer made over spin configurations but over graphs

---

[1] The probability of a spin flip is exponential versus $Q : P(\Delta t) = \exp(-Q \Delta t)$.
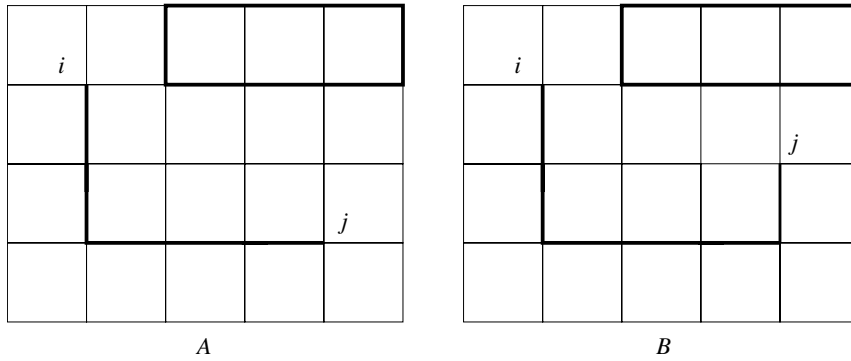
Fig. 3. Illustration of a move with the Worm algorithm. Thick lines stand for one example of graph contributing to the correlation function: one path joining the sites i et j (the sources) and possibly closed loops. These two graphs differ in one iteration of the Worm algorithm. According to (13), the graph on the left and on the right have respective equilibrium probabilities $P_A \propto \tanh^5 \beta J$ and $P_B \propto \tanh^6 \beta J$ (we neglect loops that are not relevant for this purpose). From the detailed balance condition (4), the Metropolis acceptance rates are $A(A \to B) = \min(1, \tanh \beta J)$ and $A(B \to A) = \min(1, 1/\tanh \beta J)$. In both cases $T(A \to B) = T(B \to A) = 1/2D$ where D is the dimension of the (hypercubic) lattice.

that are generated as follows. One of the two sources, say i, is mobile. At every steps, it moves randomly to a neighboring site. Any nearest-neighbor site can be chosen with the trial probability $T(A \to B) = 1/2D$, where D is the dimension of the (hypercubic) lattice. If no link is present between the two sites, then a link is created with the acceptance probability:

$$A(A \to B) = \min(1, \tanh \beta J). \tag{14}$$

If a link is already present, it is erased with the acceptance probability:

$$A(A \to B) = \min(1, 1/\tanh \beta J). \tag{15}$$

Since $0 \leq \tanh x < 1$ for all values $x > 0$, the probability (15) is equal to unity and the link is always erased. These probabilities are obtained considering the Metropolis acceptance rate Eq. (7) and the expression of the correlation function (13). The procedure is illustrated in Fig. 3.

The open paths in the two graphs of Fig. 3 are made of 5 (graph on the left) and 6 (graph on the right) lattice spacings (we neglect the loop that does not contribute in this example). According to (13), the graphs on the left and on the right have equilibrium probabilities $P_A \propto \tanh^5(\beta J)$ and $P_B \propto \tanh^6(\beta J)$, respectively. The transition probability (7) is thus written $W(A \to B) = 1/2D \times \min(1, \tanh \beta J)$ and $W(B \to A) = 1/2D \times \min(1, 1/\tanh \beta J)$, in agreement with (14) and (15).

If the two sources meet, they can move together on another random site with a freely chosen transition probability. When the two sources move together, they leave a closed loop behind that justifies the simultaneous presence of open path and closed loop in Fig. 3. These loops may disappear if the head of the worm meets them. Compared to the Swendsen–Wang algorithm, the Worm algorithm has a dynamical exponent slightly higher in 2D but significantly lower in 3D [11]. The efficiency of this algorithm can be improved with the use of a continuous time implementation [12]. The formalism of the Worm algorithm is suitable for high temperature. In the critical region, the number of graphs that contribute to the correlation function increases exponentially. In order to check the convergence of the algorithm, we can compare it with the Wolff algorithm for the 5D Ising model with different lattice sizes in Fig. 4 (data from Ref. [12]). The two algorithms give results in good agreement, except in the critical region where the convergence of the Worm algorithm is slower as the lattice size L increases.

7. Dynamical aspects: thermalization and correlation time

In order to perform a sampling of thermodynamical quantities at a given temperature, one has to first thermalize the system. Usually, it is possible to set up the system either at infinite temperature (all spins random) or in the ground state (all spins up or down). Let us start from an initial random configuration. If the thermalization takes place above the critical temperature, then the relaxation is exponential. As we come closer to the critical point, the equilibrium correlation length becomes larger and the relaxation becomes much slower and eventually algebraic right at T_c. An example of such process for the 2D Ising model is given in Fig. 5. Initially ($t = 0$), the system is prepared at infinite temperature, all the spins are random. Then the Glauber dynamics is applied at the critical temperature. We see the nucleation and the evolution of correlated spin domains in time ($t = 0, 10, 100, 1000$ from left to right, measured in terms of MC iterations). It is possible to show that in such a quench, the correlation length grows with time like [13]:

$$\xi(t) \sim t^{1/z_c}, \tag{16}$$

where z_c is the critical dynamical exponent ($z_c = 2.1665(12)$ [5] for the 2D Ising model). The time needed to complete thermalization at criticality is therefore $\tau_{\text{th}} \sim L^{z_c}$. In the case of a subcritical quench, the system has to choose between two ferromagnetic states of opposite magnetization. Again, the relaxation is slow because there is nucleation and growth
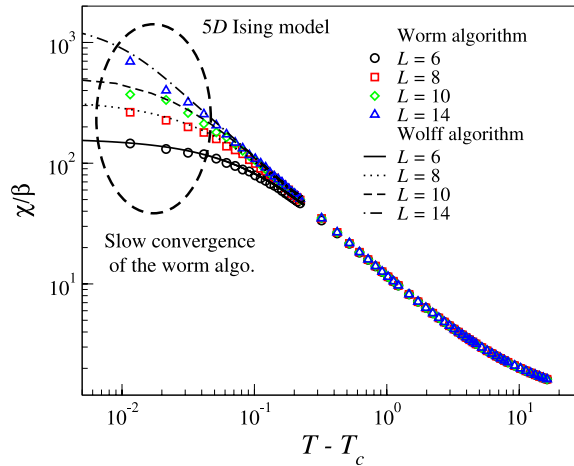
Fig. 4. (color online) Comparison of the magnetic susceptibility χ obtained with the Worm algorithm and the Wolff algorithm for the Ising model in $5D$ with different lattice sizes. The results are in excellent agreement for both algorithms in the high temperature phase. As we move closer to T_c i.e. in the critical regime (dashed line ellipse), the convergence of the Worm algorithm is slower as the lattice size increases (keeping all other parameters fixed). This is explained by an exponential increase in the number of graphs.
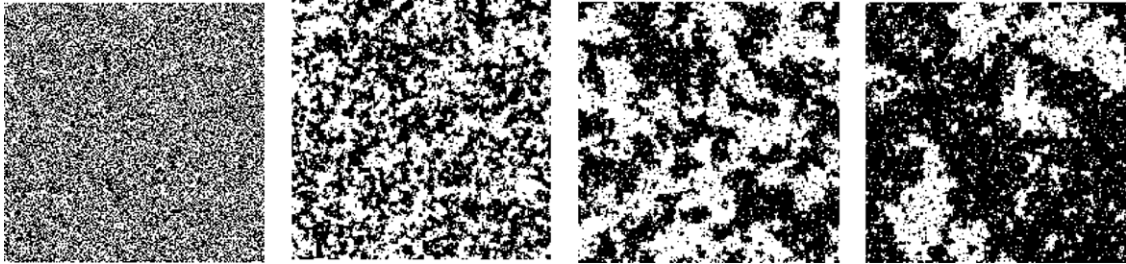*Source:* Data from Ref. [12].

Fig. 5. Snapshots of the evolution of the $2D$ Ising of size 200×200 after a quench from a disordered state until equilibration at the critical temperature T_c with the Glauber dynamics. We see the nucleation and the growth of correlated domains. From left to right, $t =0, 10, 100, 1000$ (expressed in Monte Carlo unit time after the quench). The thermalization is completed when the correlation length reaches its static value $\xi \sim L$. Using Eq. (16), the thermalization time at criticality behaves like $\tau_{\text{th}} \sim L^{z_c}$ (where $z_c \approx 2.1665(12)$ [5] is the critical dynamical exponent) and therefore diverges with system size.

of domain of opposite magnetization. We define the typical size of a domain at a time t by $L_d(t)$. The thermalization process involves the growth (coarsening) of these domains, until eventually one domain spans the whole system. Only then, the equilibrium is reached (in the low temperature phase, the expectation of the absolute value of the magnetization is nonzero). The motion of the domain walls is mostly diffusive, i.e., $L_d(t) \sim t^{1/z}$ with a dynamical exponent $z = 2$ [13]. The walls have to cover a distance $\sim L$, so that the thermalization time scales as $\tau_{\text{th}} \sim L^2$. This time diverges again with system size. Starting from an ordered state does not help for the critical quench (but it does help to start in the ground state to thermalize the system at $T < T_c$).

Once the system is thermalized, one has to be aware of another dynamical effect: the correlation time. This is the time needed to perform sampling between statistically uncorrelated configurations. In the high temperature phase, the correlation time is equal to the thermalization time, up to some factor close to unity. This is not surprising, as proper thermalization requires the configuration to become uncorrelated from the initial state. Practically, in all Monte Carlo simulations, one has to estimate τ at the temperature of sampling to treat properly the error bars.

In the low temperature phase, after thermalization, the magnetization is either positive or negative, and stays like that over prolonged periods of time. So-called magnetization reversals do occur now and then, but the characteristic time between those increases exponentially with system size. Because of the strict symmetry between the parts of configuration space with positive and negative magnetization, in practice one is not so much interested in the time of magnetization reversals, but rather in the correlation time τ within the up- or down-phase; and this time is some temperature-dependent constant, irrespective of the system size provided it is significantly larger than the correlation length.

Let us consider now the two-time spin–spin correlation functions at equilibrium in the framework of dynamical scaling [14]. We will use for this purpose a continuous space, so that the spin S_i on the site i is now denoted by $S_{\vec{r}}$ where $\vec{r}$ is the position vector. Upon a dilatation with a scale factor b, the equilibrium correlation $C(\vec{r}, t, |T - T_c|) = \langle S_0(0) \cdot S_{\vec{r}}(t) \rangle$ is assumed to satisfy the homogeneity relation:

$$C(\vec{r}, t, |T - T_c|) = b^{-2x_\sigma} C\big(r/b, t/b^{z_c}, |T - T_c|b^{1/\nu}\big), \tag{17}$$

where x_σ is the scaling dimension of magnetization density with $2x_\sigma = \eta$ for two-dimensional systems. The motivation for the last two arguments of the scaling function in Eq. (17) comes from the behavior of the correlation length either with time, $\xi \sim t^{1/z_c}$, or with temperature, $\xi \sim |T - T_c|^{-1/\nu}$. Setting $b = t^{1/z_c}$ in Eq. (17), we obtain:

$$C(\vec{r}, t) = t^{-\eta/z_c} \mathcal{C}\left(r/t^{1/z_c}, |T - T_c| t^{1/(\nu z_c)}\right). \tag{18}$$

The algebraic prefactor corresponds to the critical behavior while the scaling function includes all corrections to it. The characteristic time:

$$\tau \sim \xi^{z_c} \sim |T - T_c|^{-\nu z_c}, \tag{19}$$

appears as the relaxation time of the system. Here we are interested only in autocorrelation functions for which $r = 0$. Moreover, we expect an exponential decay of the scaling function $\mathcal{C}(t/\tau)$ in the paramagnetic phase. Therefore, the autocorrelation function can generally be written at equilibrium as:

$$C(t, T) \sim \frac{e^{-t/\tau}}{t^{\eta/z_c}}. \tag{20}$$

The spin–spin autocorrelation function $C(t, T)$ versus time is plotted in Fig. 6 (left) for the 2D Ising model of size $N = 50 \times 50$. The different curves correspond to different inverse temperatures $\beta = 0.35$ to 0.39 (the critical inverse temperature is $\beta_c = 1/2 \ln(1 + \sqrt{2}) \approx 0.44$). We observe an increase in the autocorrelation time as the temperature comes closer to T_c. The autocorrelation time can be obtained from a fit of the curve in the main graph, assuming Eq. (20). The result is plotted versus $|T - T_c|$ in the inset. The numerics tend to the behavior of Eq. (19) as T goes to T_c.

Some other aspects of critical dynamics are interesting to study, for instance, the time evolution of the equilibrium mean-square displacement of the magnetization. It is defined as:

$$h(t) = \left\langle (M(t) - M(0))^2 \right\rangle. \tag{21}$$

At small time differences ($t < 1$), the dynamics consists of sparsely distributed proposed spin flips, each of which has a nonzero acceptance probability. The difference between the number of flipped up-spins and down-spins is on average zero, but shows fluctuations. Since these spin flips are uncorrelated and their number scales as $L^D t$, the squared difference scales also as $L^D t$. Hence, in the short-time regime ($t \approx 1$), $h(t)$ behaves as:

$$h(t) \sim L^D t. \tag{22}$$

The magnetization at long time $t > \tau (\tau \sim L^{z_c})$ is no longer correlated i.e. $\langle M(t) \cdot M(0) \rangle \approx 0$. Moreover, the expectation value of the squared magnetization is directly related to the magnetic susceptibility like $\chi \equiv \beta/N \langle M^2 \rangle$ where $N = L^D$ (for $T \geq T_c$, $\langle M \rangle \approx 0$). It diverges at the critical temperature with system size as $\sim L^{\gamma/\nu}$, implying:

$$h(t) = \left\langle M(t)^2 + M(0)^2 - 2M(t) \cdot M(0) \right\rangle \approx 2 \left\langle M^2 \right\rangle \sim L^{D+\gamma/\nu} \quad (t > \tau). \tag{23}$$

Therefore $h(t)$ has to grow from $h(t \approx 1) \sim L^D$ to $h(t \sim L^{z_c}) \sim L^{D+\gamma/\nu}$. Assuming a power law behavior, it follows that $h(t) \sim t^{\gamma/(\nu z_c)}$. Therefore, in this regime, we can assume the following form for $h(t)$:

$$h(t) \sim L^{D+\gamma/\nu} \mathcal{F}(t/L^{z_c}), \tag{24}$$

where $\mathcal{F}$ is a scaling function with the limit $\mathcal{F}(x) = $ constant when $x \gg 1$ and $\mathcal{F}(x) \sim x^{\gamma/(\nu z_c)}$ at intermediate times. We measured the function $h(t)$ as defined in Eq. (21) in simulations of the 2D Ising model at the critical temperature for various systems sizes. The scaling function $\mathcal{F}$ is plotted in Fig. 6 (right), using the exponents $\gamma = 1.75$, $\nu = 1$ and $z_c = 2.1665$. With increasing system size, the data become increasingly consistent with a simple power law behavior for intermediate times between the early-time behavior given by Eq. (22) and the time of saturation behaving like $\sim L^{z_c}$. This power law behavior corresponds to an instance of *anomalous diffusion* i.e. a mean-square deviation growing as a power law with an exponent $\neq 1$ compatible with:

$$h(t) \sim t^{\gamma/(\nu z_c)} \sim t^{0.81}. \tag{25}$$

Assuming Eq. (25), the magnetization autocorrelation function for intermediate times (i.e. between times of order unity and the correlation time, thus spanning many decades) is compatible with the first terms of the Taylor expansion of a stretched exponential, which might lead to an unusual interpretation of the autocorrelation function:

$$C_M(t) = \langle |M(t)| \cdot |M(0)| \rangle \sim \exp\left(-(t/\tau)^{\gamma/(\nu z_c)}\right). \tag{26}$$

This conjecture compares well with the numerics for the correlation function in Fig. 6 (right, in the inset) and gives evidences that the dynamical critical exponent z_c appears at relatively small times $1 \ll t \ll L^{z_c}$. This yet unpublished result would be interesting to investigate in the future.
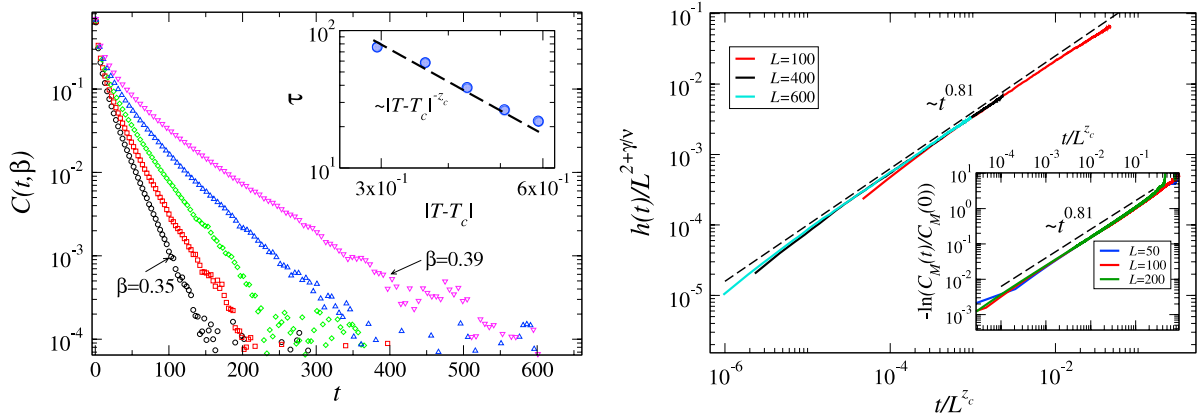
Fig. 6. (color online) Simulation results for the 2D Ising model. (Left) Spin–spin autocorrelation function $C(t)$ at different $\beta = J/(k_B T) = 0.35, 0.36, 0.37,$ 0.38 and 0.39 ($N = 50 \times 50$). The decay is slower as we come closer to $\beta_c \approx 0.44$. Inset: assuming the expression Eq. (20), the correlation time τ is plotted versus $|T - T_c|$ and follows the trend of Eq. (19). (Right) Scaling function $h(t)/L^{2+\gamma/\nu}$ versus t/L^{z_c} at T_c for different lattice sizes. At intermediate times, it displays an *anomalous diffusion* compatible with $h(t) \sim t^{\gamma/(\nu z_c)} = t^{0.81}$. Inset: the autocorrelation $C_M(t) = \langle |M(0)| \cdot |M(t)| \rangle$ is compatible with the stretched exponential Eq. (26), deduced from the behavior of $h(t)$.

8. Conclusion

In these lecture notes, we provide an introduction to Monte Carlo simulations that are a way to produce a set of representative configurations of a statistical system. We start with the basic principles: ergodicity and detailed balance. In the next parts, we present several Monte Carlo algorithms. To illustrate their functioning, we use the example of the Ising model. This model is defined by scalar spins on a lattice that interact via nearest-neighbor interactions. This is the paradigmatic system for second order phase transitions: a critical temperature separates a disordered phase at high temperature and an ordered phase at low temperature. These different regimes induce different strategies for the Monte Carlo simulations. In the disordered phase, local algorithms such as Metropolis or Glauber are efficient. In the critical region, the appearance of long range correlations have set a computational challenge. It has been solved by the use of cluster algorithms such as the Wolff algorithm that flips a whole cluster of correlated spins. Below the critical temperature, when the probability of spin flip is low, it is a gain of computational time to program the continuous time algorithm. It forces the system into new configurations, with a jump in time according to the probability of transition. The last description is an interesting algorithm based on an alternative representation of the model in terms of graphs instead of spins. We conclude with important considerations on the dynamics with the thermalization and the correlation time.

Acknowledgments

References

[1] J.J. Binney, N.J. Dowrick, A.J. Fisher, M.E. Newman, The Theory of Critical Phenomena, Clarendon Press, Oxford, 1995.
[2] M.E.J. Newman, G.T. Barkema, Monte Carlo Methods in Statistical Physics, Oxford University Press, 1999.
[3] R.J. Glauber, Time-dependent statistics of the Ising model, J. Math. Phys. 4 (1963) 294.
[4] U. Wolff, Collective Monte Carlo updating for spin systems, Phys. Rev. Lett. 62 (1989) 361.
[5] M.P. Nightingale, H.W.J. Blöte, Dynamic exponent of the two-dimensional Ising model and Monte Carlo computation of the subdominant eigenvalue of the stochastic matrix, Phys. Rev. Lett. 76 (1996) 4548.
[6] R.H. Swendsen, J.S. Wang, Nonuniversal critical dynamics in Monte Carlo simulations, Phys. Rev. Lett. 58 (1987) 86–88.
[7] S. Gündüç, M. Dilaver, M. Aydin, Y. Gündüç, A study of dynamic finite size scaling behavior of the scaling functions-calculation of dynamic critical index of Wolff algorithm, Comput. Phys. Comm. 166 (2005) 1.
[8] J. Du, B. Zheng, J.-S. Wang, Dynamic critical exponents for Swendsen–Wang and Wolff algorithms obtained by a nonequilibrium relaxation method, J. Stat. Mech. Theory Exp. (2006) P05004.
[9] D.T. Gillespie, A general method for numerically simulating the stochastic time evolution of coupled chemical reactions, J. Comput. Phys. 22 (1976) 403–434;
Exact stochastic simulation of coupled chemical reactions, J. Phys. Chem. 81 (1977) 2340–2361.
[10] A.B. Bortz, H.M. Kalos, J.L. Lebowitz, A new algorithm for Monte Carlo simulation of Ising spin systems, J. Comput. Phys. 17 (1975) 10–18.
[11] Y. Deng, T.M. Garoni, A.D. Sokal, Dynamic critical behavior of the worm algorithm for the Ising model, Phys. Rev. Lett. 99 (2007) 110601.
[12] B. Berche, C. Chatelain, C. Dhall, R. Kenna, R. Low, J.-C. Walter, Extended scaling in high dimensions, J. Stat. Mech. (2008) P11010.
[13] A.J. Bray, Theory of phase-ordering kinetics, Adv. Phys. 43 (1994) 357.

[14] P.C. Hohenberg, B.I. Halperin, Theory of dynamic critical phenomena, Rev. Modern Phys. 49 (1977) 435.
[15] N.V. Prokof'ev, B.V. Svistunov, I.S. Tupitsyn, "Worm" algorithm in quantum Monte Carlo simulations, Phys. Lett. A 238 (1998) 253–257.
[16] N. Prokof'ev, B. Svistunov, Worm algorithms for classical statistical models, Phys. Rev. Lett. 87 (2001) 160601.