6 Monte Carlo Methods with Applications to Spin Systems

Werner Krauth
Laboratoire de physique statistique
Ecole normale supérieure
24 rue Lhomond, 75005 Paris, France

Contents

E. Pavarini, E. Koch, and U. Schollwöck
Emergent Phenomena in Correlated Matter
Modeling and Simulation Vol. 3
Forschungszentrum Jülich, 2013, ISBN 978-3-89336-884-6
http://www.cond-mat.de/events/correl13

1 Introduction

In my lecture at the Jülich Autumn School 2013, I discuss Monte Carlo methods, and their application to spin systems, in the language of my recent text book [1], but in much less detail. During the lecture, I first concentrate on the relation between Monte Carlo methods and the concept of incompressible flow (Section 2), which allows one to understand the crucial detailed and global balance conditions. I then introduce to the "a priori" probabilities, which are at the basis of the Metropolis-Hastings algorithm (Section 3). After an introduction to the local algorithms (heatbath, local Metropolis), and the classic faster-than-the-clock approaches (Section 4), I then discuss the Swendsen-Wang and Wolff cluster algorithms for spin systems (Section 5), as an illustration of the great liberty one has in designing Monte Carlo algorithms. I finish with the coupling approach (Section 6) which has provided essential for rigorously understanding for how long a simulation must be run until the computational output reflects thermal equilibrium (the classical Boltzmann distribution or the quantum density matrix) rather than the initial state of the Markov chain.

2 Markov chains and incompressible flow

Markov-chain Monte Carlo strives to sample a distribution π by starting from another distribution π_0 and using an incremental algorithm $p_{i\to j}$ to move between configurations i and j. In each time step (iteration) $t = 1, 2, 3, \ldots$, the probability distribution π^t is sampled. The target distribution is approached in the limit of infinite simulation time $\pi = \pi^\infty$. Convergence is always exponential in time, and it can be assured as long as we satisfy (in addition to an ergodicity requirement) the global balance condition. This means that the flow into each configuration i equals the flow out of it:

$$\sum_k P_{i\to k} = \sum_l P_{l\to i} \quad \text{(global balance).} \tag{1}$$

Here, the flow $P_{k\to i}$ denotes the product of the stationary probability π_k with the conditional probability $p_{k\to i}$ to move from k to i given that the system is at k. The global flow condition can be easily understood under the stationarity condition, as the first sum in Eq. (1) equals $\sum_k P_{i\to k} = \pi_i$ (see [1], Section 1.1.4). Eq. (1) is satisfied in particular if we balance the flow between each pair of configurations i and k individually:

$$P_{i\to k} = P_{k\to i} \quad \forall i, k \quad \text{(detailed balance).} \tag{2}$$

This detailed balance condition is satisfied for example by the Metropolis algorithm

$$P_{i\to k} = \min(\pi_i, \pi_k) \tag{3}$$

(the right-hand-side of this equation is manifestly symmetric in i and k, so is the left-hand-side). Usually, Eq. (3) is written in terms of the probabilities $p_{i\to k}$, etc, that is by dividing by $\pi_i > 0$. This yields

$$p_{i\to k} = \min(1, \pi_k/\pi_i), \tag{4}$$

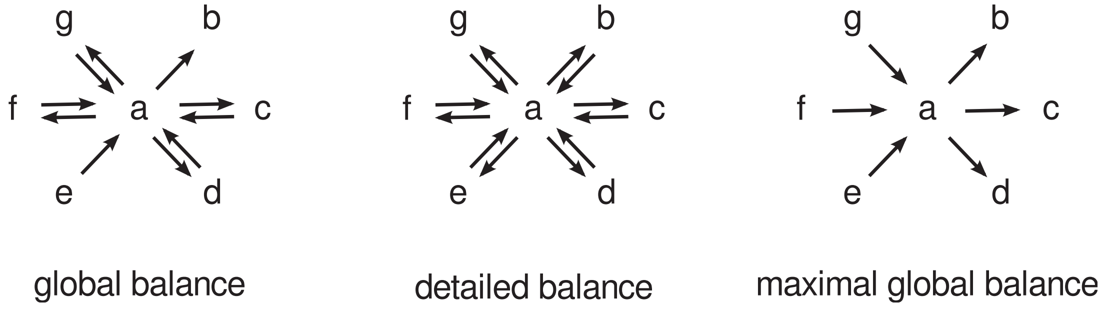

Fig. 1: *Balance condition for Markov-chain Monte Carlo moves. The global balance condition (left) is necessary for the convergence of the Markov chain towards the stationary probability measure π. The time-honored detailed balance condition (center) is often realized through the Metropolis algorithm of Eq. (3) and Eq. (4). The unidirectional motion of the maximal global balance condition (right) is realized in a number of modern algorithms (see [2]. Figure adapted from [3]).*

an equation that has been written down in thousands of papers and a huge number of computer programs. There are many choices of P which satisfy these balance conditions. This implies that a great many valid Monte Carlo algorithms are out there, some of them slow and some of them fast, most of the latter ones yet to be discovered.

Much more could be said, and has been written, about the detailed and global balance conditions, but the most important finding is the following: While convergence towards the equilibrium distribution π is generally exponential ($\propto \exp(-t/\tau)$), even for continuum systems, it is very difficult to estimate the correlation time τ. This question — what is the correlation time — is one of the three great conceptual issues in Monte Carlo algorithms and, really, the most serious one. The other two are finite-size scaling, and the estimation of the variance for observables (see [1], Section 1.4). We will take up the convergence issue in Section 6, in the context of coupling algorithms.

3 A priori probabilities

How do we move from configuration i to configuration j, and what do we mean with the (conditional) probability $p_{i\to j}$ to go from configuration i to configuration j (see our discussion in Section 1)? This question is best discussed with the help of *a priori probabilities*: While the flow $P_{i\to j}$ is a product of the stationary probability π_i and the conditional probability $p_{i\to k}$, the latter is put together from a probability to *propose* the move from i to j (the "a priori" probability) times the conditional probability to *accept* it:

$$p_{i\to j} = \mathcal{A}_{i\to j}\, p^{\text{acc}}_{i\to j}. \tag{5}$$

This equation can be illustrated by a simple example, as the "triangle algorithm" for a particle in a box (see [1], Sect. 1.1.6). As the balance condition of Eq. (1) only considers the flow, we are free to choose an arbitrary a priori probability $\mathcal{A}$ and bias it with a given acceptance

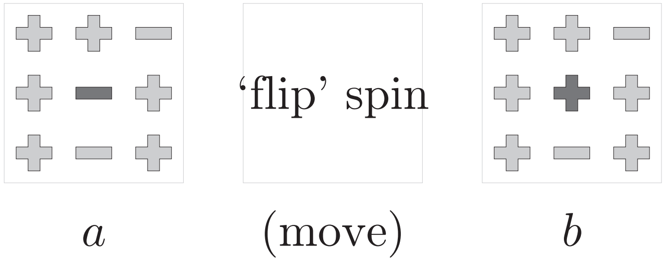

Fig. 2: *Local Monte Carlo move $a \to b$ in the Ising model. The spin to be flipped is sampled uniformly through the lattice and the flip should be accepted with probability* $\min(1, \exp(-\beta(E_b - E_a))$ *(from [1]).*

probability in order to fall back on our feet (obtain the correct flow). With a priori probabilities, the Metropolis Monte Carlo algorithm takes the form

$$p^{\text{acc}}_{i\to j} = \min\left(1, \frac{\mathcal{A}_{j\to i}}{\pi_i}\frac{\pi_j}{\mathcal{A}_{i\to j}}\right). \tag{6}$$

If the a priori probability $\mathcal{A}_{i\to j}$ to propose the move from i to j is equal to π_j, the acceptance probability $p^{\text{acc}}_{i\to j}$ is equal to one and, in a sense, we have already solved the problem. This means two things: (i) we have no rejections and (ii) we can usually make quite large moves, as the reason for making small ones was the presence of rejections. See [1], Section 1.1.6, for a discussion of the relation between this Metropolis-Hastings algorithm and perturbation theory of quantum physics.

4 Local algorithm, faster-than-the-clock algorithms

Let us now do our first real Monte Carlo simulation of Ising spins $\sigma_k = \pm 1, k = 1 \dots N$ on a lattice, say the two-dimensional square lattice. We use the energy

$$E = -J \sum_{\langle i,j\rangle} \sigma_i \sigma_j, \tag{7}$$

where i and j are neighbors on the lattice. In the following, we choose $J = 1$, and the stationary probability of a configuration $\boldsymbol{\sigma}$ is given by $\pi_{\boldsymbol{\sigma}} = \exp[-E(\boldsymbol{\sigma})/kT)$, where k is the Boltzmann constant and T the temperature, with $\beta = 1/kT$.

The Ising model can be simulated by the Metropolis algorithm, by uniformly sampling at each time step a site i, as in Fig. 2. If we call “a” the original configuration and “b” the configuration obtained through the flip $\sigma_i \to -\sigma_i$, then the Metropolis probability for accepting the move is given by

$$p = \min[1, \exp(-\beta(E_b - E_a)]. \tag{8}$$

One may picture the spin σ_i in the “molecular” field h_i equal to the sum of the neighboring spins, and the flipping probability turns out to be

$$p = \min[1, \exp(-2\beta\sigma_i h_i)]. \tag{9}$$

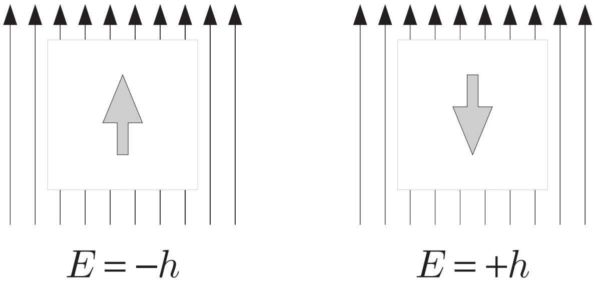

Fig. 3: *A single spin in an external field: building block of local Monte Carlo algorithms (figure adapted from [1]).*

Instead of the Metropolis algorithm, we may also use the heatbath algorithm, which consists in taking out the spin i and in replacing it with thermalized values of σ_i

$$\pi_h^+ = \frac{\mathrm{e}^{-\beta E^+}}{\mathrm{e}^{-\beta E^+} + \mathrm{e}^{-\beta E^-}} = \frac{1}{1 + \mathrm{e}^{-2\beta h}}, \tag{10}$$

$$\pi_h^- = \frac{\mathrm{e}^{-\beta E^-}}{\mathrm{e}^{-\beta E^+} + \mathrm{e}^{-\beta E^-}} = \frac{1}{1 + \mathrm{e}^{+2\beta h}}. \tag{11}$$

(See [1], Section 5.2.1 for simple programs, and the author's website for a basic Python program.) Practically, we sample a random number Υ uniformly distributed in the unit interval. If $\Upsilon < \pi_h^+$, we set $\sigma_i = 1$, otherwise $\sigma_i = -1$). In the long run, the program will converge towards the Boltzmann distribution, that is, spin configurations $\boldsymbol{\sigma}$ will be visited with their Boltzmann probability $\pi_{\boldsymbol{\sigma}}$.

In the local Metropolis or heatbath algorithm, the flip concerns a single spin subject to the molecular magnetic field produced by its neighbors. Let us analyze this problem, a single spin in a magnetic field, in more detail (see Fig. 3). At low temperature, the acceptance probability of the algorithm is very small: We consider the spin, and try to flip it (but reject the flip), then try again, reject again, etc. Many calculations will be wasted, as we compute probabilities and draw random numbers, but most often reject the move. There is a better way to organize the calculation: it consists in finding out how many times we will reject the flip before accepting it. This is called the faster-than-the-clock approach (in a single iteration, we may determine that the flip in Fig. 3 is rejected 7 times before it is accepted). As discussed in detail in [1], Section 7.2.2, the faster-than-the-clock approach is at the base of the classic Bortz-Kalos-Lebowitz (BKL) algorithm [4] among others. The general lesson is that very small acceptance probabilities, as they appear at low temperatures in the Ising model, can generally be avoided through a clever trick. The inverse is also true: in some recent approaches, almost all Monte Carlo move are accepted, and their results can be easily added up [5, 6]. We can then use the faster-than-the-clock approach to sample the first time when a long streak of events (or non-events) comes to an end.

5 Cluster algorithms

Close to the critical point, the transition between the paramagnetic and the ferromagnetic phases, local Monte Carlo algorithms slow down considerably. This is due to a complicated interplay between several factors: On the one hand, close to the critical point, finite-size effects are large and the physical observables at one system size N can be quite different from the ones for larger or smaller systems (in consequence, we must simulate very large systems, which is naturally CPU-time intensive). On the other hand, computing these observables for fixed N may take quite some CPU time because of critical slowing down. In a computational context, this phenomenon describes the appearance of slow variables (in the Ising model it is the total magnetization), whose distribution becomes very wide (at the critical point, total magnetizations from $-N$ through $+N$ appear with sizable probability). As the local Monte Carlo algorithm only flips a single spin at a time, it takes a long time to move across the probability distribution of the magnetization, and we experience slowing down (see [1], Section 5.2.1, for a detailed discussion).

To speed up the simulation, we should flip more than one spin concurrently, but this cannot be achieved by parallel flipping of uncorrelated spins: The attempted moves are indeed larger, but with the increased rejection rate, the effective move will get smaller instead of larger. We understand that we should propose larger moves but, quite importantly, we must get them accepted. The Wolff cluster algorithm [7] (an efficient version of the Swendsen-Wang cluster algorithm [8]) does this marvelously. It provides a prime example for the use of the Metropolis-Hastings paradigm. Our discussion closely mirrors [1], Section 5.2.3.

In the Wolff cluster algorithm, we start with one site, then add any of the neighboring sites with probability p, if they carry the same spin, etc. The construction in Fig. 4a then came to a halt because all of the 14 links $++$ across the cluster boundary were rejected. The move consists in flipping the cluster.

To apply Eq. (6), we compute the a priori probability $\mathcal{A}_{a\to b}$, the probability of stopping the cluster construction process at a given stage rather than continuing and including more sites (see the configuration a in Fig. 4). $\mathcal{A}_{a\to b}$ is given by an interior part (the two neighbors inside the cluster) and the stopping probability at the boundary: each site on the boundary of the cluster was once a pocket site and the construction came to a halt because none of the possible new edges was included. Precisely, the boundary $\partial\mathcal{C}$ of the cluster (with one spin inside and its neighbor outside) involves two types of edge:

$$\left\{\begin{array}{c}\text{cluster in } a\\ \text{in Fig. 4}\end{array}\right\}: \quad \overbrace{\left[\begin{array}{ccc}\text{inside} & \text{outside} & \#\\ + & - & n_1\\ + & + & n_2\end{array}\right]}^{\text{edges across } \partial\mathcal{C}} \quad E|_{\partial\mathcal{C}} = n_1 - n_2 \tag{12}$$

(in Fig. 4, $n_1 = 17$, $n_2 = 14$). In order to acquire all information needed for Eq. (6), we must analyze likewise the return flip in Fig. 4 from configuration b back to configuration a. the same interior part as before, but a new boundary part $\mathcal{A}_{b\to a} = \mathcal{A}_{\text{in}} \cdot (1-p)^{n_1}$, because

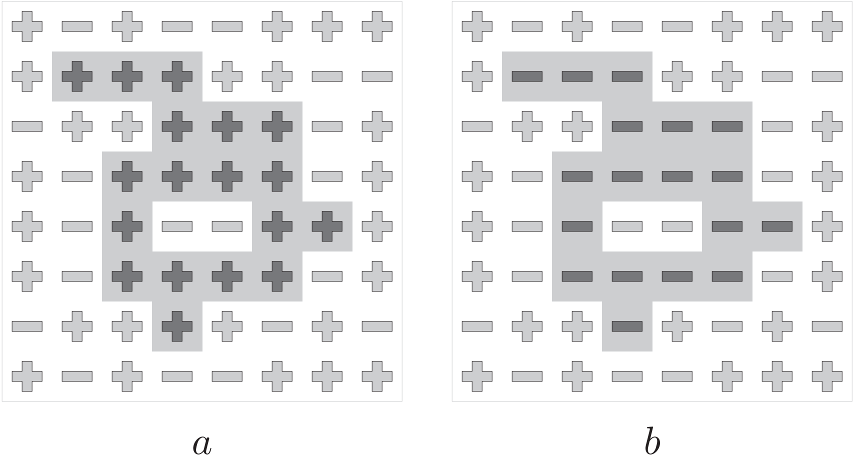

Fig. 4: *Wolff cluster algorithm in the Ising model. Starting from an initial site, we move on to neighboring sites with the same value of the spin, and add them to the cluster with probability p. The cluster construction terminates, and the cluster is flipped, when no more neighboring sites can be added. In order to set up the algorithm correctly, we have to consider both the initial configuration a and the final configuration b (see Eq. (6)). For the special value of p given in Eq. (14), this algorithm is rejection free (figure taken from [1]).*

there were n_1 opportunities to let the cluster grow and again none was accepted. We also note that $\pi_a = \pi_{\text{in}}\pi_{\text{out}} \exp(-\beta(n_1 - n_2))$, where π_{in} takes into account all the edges with both sites inside the cluster and π_{out} all those at its outside. By an argument similar to that above, the statistical weight of configuration b is $\pi_b = \pi_{\text{in}}\pi_{\text{out}} \exp(-\beta(n_2 - n_1))$. The interior and exterior contributions to the Boltzmann weight are the same as for configuration a. All the ingredients of the detailed-balance condition in Eq. (6) are now known:

$$\mathrm{e}^{-\beta(n_1-n_2)}(1-p)^{n_2} p^{\text{acc}}_{a\to b} = \mathrm{e}^{-\beta(n_2-n_1)}(1-p)^{n_1} p^{\text{acc}}_{b\to a}. \tag{13}$$

which leads to

$$p^{\text{acc}}_{a\to b} = \min\left[1, \left(\frac{\mathrm{e}^{-2\beta}}{1-p}\right)^{n_2} \left(\frac{1-p}{\mathrm{e}^{-2\beta}}\right)^{n_1}\right]. \tag{14}$$

The algorithm is at its peak efficiency for $p = 1 - \exp(-2\beta)$, where the acceptance probability is equal to one: we can construct the cluster, then flip it, build another one, turns it over .... This Wolff cluster algorithm [7], one of the first of its kind, can be implemented in a dozen or so lines of code: see the website of [1].

6 Coupling approach

As final subject, we now consider the problem of correlation times: that is we try to understand at what time the sampled configurations reflect the stationary distribution π^∞ and no longer the starting configuration π^0. For many real-life problems, this complex and crucial question has no truly satisfactory solution: Although we know that the corrections to the equilibrium state decay as $\mathrm{e}^{-i/\tau}$, we cannot really compute τ. Quite often whole communities underestimate τ,

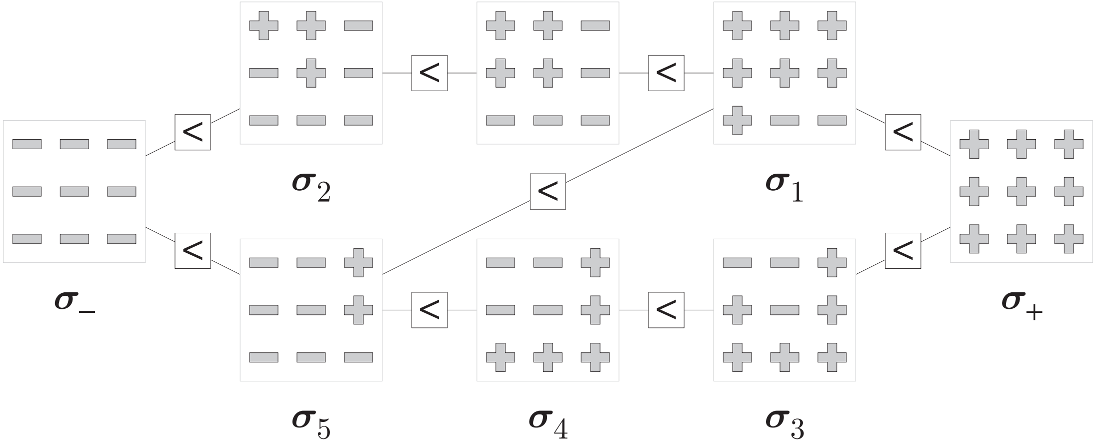

Fig. 5: *Partial order in a two-dimensional* 3×3 *Ising spin model. Order relations are indicated, but not all configurations are related to each other. The all + configuration* $\boldsymbol{\sigma}_+$ *is larger, and the all - configuration* $\boldsymbol{\sigma}_-$ *is smaller than all other configurations. The partial order is preserved by the heatbath algorithm (figure taken from [1]).*

and it is the advent of new algorithms that clearly make that point (as in [2, 9, 10]). Beginners often think it is sufficient to start the simulation with some "random" initial condition, but this is a misconception, as it is normally impossible to sample an initial configuration from a distribution π^0 anywhere close to π. In the Ising model, a time-honored practice has consisted in starting *two* initial configurations, one from the all up configuration $\sigma_k = +1 \ \forall k$ and another one from an all down configuration $\sigma_k = -1 \ \forall k$. According to popular wisdom, when both simulations reached similar magnetizations, they reached equilibrium. This idea turned out to be basically correct. To explain why, we first have to understand the concept of partial order: One configuration a is "smaller" than another one, b, if $\sigma_i^a \leq \sigma_i^b \ \forall i$. Not all configurations can be compared to each other, but all can be compared to the all up and the all down configuration (see Fig. 5). If we use the same random numbers for choosing the site i and for updating the spin σ_i (coupling), and update the configurations using the heatbath algorithm of Eq. (11), then the partial order is invariant under the dynamics. This means that the children of the all $+1$ configuration will always stay above the descendants of the all -1 configuration. As soon as both match, at the coupling time τ_{coup}, all possible initial configurations will have produced the same output. This means that the memory of the initial configuration has been lost completely, with the implication that the coupling time is a rigorous upper bound for the convergence time. Propp and Wilson, in a classic paper [12], succeeded in turning the coupling approach into a rigorous exact sampling method (from the distribution π, without *any* corrections). This has far-reaching consequences, most of which have not been explored so far. See [1], Section 1.1.7 and [13] for a detailed discussion of the coupling approach, and of its most surprising finding, namely that a Markov chain can be used to sample π^∞, without any correction whatsoever.

References

[1] W. Krauth, *Statistical Mechanics: Algorithms and Computations* (Oxford University Press, 2006)

[2] E.P. Bernard, W. Krauth, and D.B. Wilson, Phys. Rev. E **80**, 056704 (2009)

[3] S.C. Kapfer and W. Krauth, arXiv:1301.4901 (2013)

[4] A.B. Bortz, M.H. Kalos, and J.L. Lebowitz, J. Comput. Phys. **17**, 10 (1975)

[5] E.A.J.F. Peters and G. de With, Phys. Rev. E **85**, 026703 (2012)

[6] M. Michel, S.C. Kapfer, W. Krauth, Manuscript in preparation (2013)

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

[8] R.H. Swendsen, J.S. Wang, Phys. Rev. Lett. **58**, 86 (1987)

[9] E.P. Bernard, W. Krauth, Phys. Rev. Lett. **107**, 155704 (2011)

[10] M. Engel, J.A. Anderson, S.C. Glotzer, M. Isobe, E.P. Bernard, W. Krauth, Phys. Rev. E **87**, 042134 (2013)

[11] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953)

[12] J.G. Propp and D.B. Wilson, Rand. Struct. and Alg. **9**, 223 (1996)

[13] C. Chanal, W. Krauth, Phys. Rev. E **81**, 016705 (2010)