PY 502, Computational Physics, Fall 2018

Monte Carlo simulations in classical statistical physics

Anders W. Sandvik, Department of Physics, Boston University

1 Introduction

Monte Carlo simulation is a very important class of stochastic methods for calculating thermal properties of many-particle systems—arguably these are the most important numerical techniques in statistical physics. Monte Carlo simulation methods are related to the elementary Monte Carlo integration methods that we discussed earlier, but are based on more efficient non-uniform sampling schemes. By using *importance sampling*, the configurations (particle positions, spin directions, etc.) of a finite but large many-body system (up to millions of degrees of freedom) can be generated according to the Bolzmann distribution, so that thermal expectation values are obtained as simple arithmetic averages of functions "measured" on the configurations.

As a simple illustration of the advantages of non-uniform Monte Carlo sampling, consider a one-dimensional integral similar to a thermal expectation value in statistical physics (the discussion here can be directly generalized to multi-dimensional integrals);

$$\langle A\rangle = \int_{-L}^{L} P(x)A(x)dx, \qquad \int_{-L}^{L} P(x)dx = 1, \tag{1}$$

where $P(x)$ is an arbitrary probability distribution. By randomly sampling M points $x_1,\ldots,x_M$ in the range $[-L,L]$, the expectation value is estimated as

$$\langle A\rangle \approx \frac{2L}{M}\sum_{i=1}^{M} P(x_i)A(x_i). \tag{2}$$

As we discussed before, if $P(x)$ is sharply peaked in a small region, the statistical fluctuations of this estimate will be large as only a small fraction of the generated points will fall within the dominant region. If we instead sample the points according to some probability distribution $W(x)$, i.e., the probability of picking a point in an infinitesimal range $[x,x+dx]$ is $W(x)dx$ (we now assume that this can be done for arbitrary $W(x)$, and leave for later discussion how this is accomplished in practice), then the estimate for the expectation value is

$$\langle A\rangle \approx \frac{1}{M}\sum_{i=1}^{M} \frac{P(x)}{W(x)}A(x_i). \tag{3}$$

This has less statistical fluctuations than the estimate (2) of the uniform sampling if $W(x)$ is peaked in the same region as $P(x)$ and if the function $A(x)$ is well-behaved, in the sense of being reasonably smooth and not very small where $P(x)$ is large and vice versa. More precisely, the fluctuation in the values sampled using the distribution $W(x)$ is given by

$$\sigma^2_W[A] = \int_L^L \left(\frac{P(x)}{W(x)}A(x) - \langle A\rangle\right)^2 W(x)dx, \tag{4}$$

which in principle can be minimized by choosing a particular $W(x)$. In general (for the multi-dimensional integrals or sums encountered in statistical physics) it is not possible in practice to find the optimal $W(x)$ that minimizes the fluctuations, but if $P(x)$ has much larger variations than $A(x)$ a very good solution is to use $W(x) = P(x)$. The expectation value is then just the simple arithmetic average of $A(x)$ over the sampled configurations

$$\langle A \rangle \approx \frac{1}{M} \sum_{i=1}^{M} A(x_i), \tag{5}$$

and the expected fluctuation of the measured values is

$$\sigma_P^2[A] = \int_L^L [A(x) - \langle A \rangle]^2 \, P(x) dx. \tag{6}$$

It should be noted that the distribution of the values A_i is typically not Gaussian, and hence to calculate the statistical errors of $\langle A \rangle$ estimated as (5) the values should first be binned, in the same way as we discussed previously in the chapter on Monte Carlo integration.

In statistical physics, P is a sharply peaked exponential function $\mathrm{e}^{-\mathrm{E}/\mathrm{k_B T}}$ of the energy and A is typically a linear or low-order polynomial function of the system degrees of freedom. The fluctuations in P are thus very large relative to those of A and the sampling using P as the probability distribution is then close to optimal. This is what is normally meant by the term *importance sampling*. Using importance sampling instead of uniform random sampling is crucial when a small fraction of the configuration space dominates the partition function, which is always the case with the Bolzmann probability in statistical mechanics models at temperatures of interest. How to achieve the correct distribution in practice is the main theme of this chapter; we will discuss importance sampling schemes for both lattice and continuous-space models.

One of the primary utilities of Monte Carlo simulation is in studies of phase transitions and critical phenomena. This will be the focus of applications discussed here. Although there are analogous simulations methods available also for quantum systems (called quantum Monte Carlo methods), we will here consider only Monte Carlo simulations of classical many-body models. In addition to describing simulation algorithms, we will also discuss how simulation data is analyzed in order to locate phase transitions and extract critical exponents.

In the following sections we will first briefly review the the expressions for thermal expectation values in systems of particles identified by coordinates in continuous space. We then consider models with discrete degrees of freedom on a lattice, focusing on spin models, the Ising model in particular. We will discuss the general form of the *detailed balance* condition that can be used to sample configurations according to any desired probability distribution. Monte Carlo simulation algorithms are for instructional purposes often developed in the context of the Ising model, and we will follow this path here as well (it should also be noted that Ising models are of continued importance in research). We will develop a standard program for simulations using the *Metropolis algorithm*, which is based on evolving (updating) configurations by flipping individual spins. We will study the performance of this method using *autocorrelation functions*, which characterize the way in which generated configurations gradually become statistically independent of the past configurations. This leads us to the problem of *critical slowing down*, which makes accurate studies close to phase transitions difficult. Critical slowing down can be often be greatly reduced, in some cases completely

eliminated, using *cluster algorithms*, where clusters of spins are flipped collectively (using cluster-building rules that satisfy detailed balance). We will develop a cluster Monte Carlo program and use it to study the ferromagnetic phase transition in the two-dimensional Ising model. For this purpose, we will also discuss *finite size scaling* methods used to extract critical points and exponents. Finally, we will return to problems involving particles in continuous space; we will develop a program for simulating a mono-atomic gas and its phase transition into the liquid state.

2 Statistical mechanics of many-body systems

We here briefly review the mathematical formalism used for evaluating thermal expectation values in classical many-body physics, considering first particles in continuous space and after that focusing on the lattice spin models for which we will develop Monte Carlo simulation algorithms initially. We will also discuss the magnetic phase transitions that can be studied using Monte Carlo simulations.

2.1 Particles in continuous space

For a system of N particles, with position coordinates $\vec{x}_i$ and momenta $\vec{p}_i$ in a d-dimensional space, the thermal expectation value of a quantity A at temperature T is given by

$$\langle A \rangle = \frac{1}{Z} \int \prod_{i=1}^{N} dx_i^d \int \prod_{i=1}^{N} dp_i^d A(\{\vec{x}_i, \vec{p}_i\}) \mathrm{e}^{H(\{\vec{x}_i,\vec{p}_i\})/k_B T}, \tag{7}$$

where Z is the partition function

$$Z = \int \prod_{i=1}^{N} dx_i^d \int \prod_{i=1}^{N} dp_i^d \mathrm{e}^{H(\{\vec{x}_i,\vec{p}_i\})/k_B T}, \tag{8}$$

k_B is Bolzmann's constant, and H is the Hamiltonian. For identical particles of mass m in a potential $U(\vec{x}_i)$ and a two-particle interaction $V(\vec{x}_i, x_j)$, the Hamiltonian is

$$H(\{\vec{x}_i, \vec{p}_i\}) = \sum_{i=1}^{N} \frac{p_i^2}{2m} + \sum_{i=1}^{N} U(\vec{x}_i) + \sum_{i \neq j} V(\vec{x}_i, x_j). \tag{9}$$

If the observable A is velocity independent (i.e., a function only of the positions x_i), the momentum integrals cancel in (7), leading to

$$\langle A \rangle = \frac{1}{Z} \int \prod_{i=1}^{N} dx_i^d A(\{\vec{x}_i\}) \mathrm{e}^{E(\{\vec{x}_i\})/k_B T}, \tag{10}$$

$$Z = \int \prod_{i=1}^{N} dx_i^d \mathrm{e}^{E(\{\vec{x}_i\})/k_B T}, \tag{11}$$

i.e., only the potential energy,

$$E(\{\vec{x}_i\}) = \sum_{i=1}^{N} U(\vec{x}_i) + \sum_{i \neq j} V(\vec{x}_i, x_j), \tag{12}$$

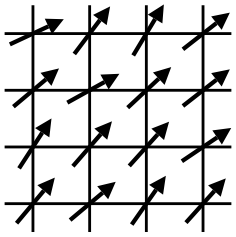

Figure 1: Spins living on the sites of a square lattice. In this case the spin vectors are confined to a plane; they are then often referred to as XY-spins.

is relevant for the static properties [the density $\rho(\vec{x})$, density fluctuations, equal-time correlation functions, etc.] of the system. Often the only velocity dependent quantity considered in equilibrium statistical mechanics is the kinetic energy, which for a single particle is given by

$$K_i = \left\langle \frac{p_i^2}{2m} \right\rangle = \frac{1}{Z_p} \int dp_i^d \frac{p_i^2}{2m} \mathrm{e}^{-p_i^2/2mk_BT}, \tag{13}$$

$$Z_p = \frac{1}{Z_p} \int dp_i^d \mathrm{e}^{-p_i^2/2mk_BT}, \tag{14}$$

since all integrals except those over $\vec{p}_i$ cancel. This gives the equipartition theorem;

$$K_i = \frac{d}{2} k_B T. \tag{15}$$

In general it is not possible to analytically calculate expactation value of more complicated function of the particle momenta or positions, except in one dimension. In a Monte Carlo simulation, real-space expectation values are evaluated by importance sampling of a finite number of the configurations $\{\vec{x}_i\}$. Before discussing how this is done for particles in continuous space, we will consider the slightly simpler case of lattice models.

2.2 Lattice and spin models

In a lattice model the degrees of freedom of the system "live" on the vertices of a lattice; these degrees of freedom can be continuous or discrete. Spin models constitute an important class of lattice models in which the degrees of freedom correspond to magnetic moments of fixed magnitude S and variable orientation; the energy is a function of the orientation angles. In nature, spin models have direct realizations in crystals of atoms with unpaired electronic spins that are localized at the atomic sites, i.e., in insulators where the spins are not carried by delocalized conduction electrons but can be associated with individual atoms ($S = 1/2$ for single unpaired electrons; higher spins can result from Hund's rule and/or electronic states with non-zero angular momentum). If the spin quantum number S is relatively large, quantum fluctuations can be neglected to a good approximation and the spins can be described with classical vectors (however, for small spin, $S = 1/2$ and 1 in particular, quantum effects do play a big role and classical models may give quantitatively, and often even qualitatively, wrong results). One of the most important models of this kind is the *Heisenberg model*, where the interaction between spins at sites i and j is proportional to their scalar product;

$$E = \sum_{i,j} J_{ij} \vec{S}_i \cdot \vec{S}_j. \tag{16}$$

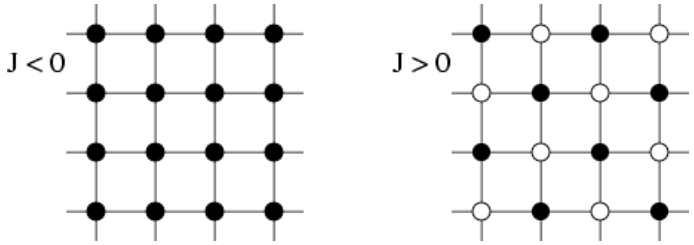

Figure 2: Lowest-energy states of the two-dimensional Ising model with ferromagnetic (left) and antiferromagnetic (right) interactions. Solid and open circles correspond to up an down spins, respectively.

The coupling constants J_{ij} are often restricted to be non-zero only for lattice sites i, j that are nearest neighbors. Here the spin vectors are three dimensional, but anisotropies can lead to effective spin models in which the spin orientations are confined to within a plane, as illustrated in Fig. 1, or along a single axis.

The simplest spin model is the *Ising model*, in which the spins have only two possible orientations along a chosen axis; "up" or "down". Denoting the degrees of freedom $\sigma_i = \pm 1$, the energy is

$$E = \sum_{i,j} J_{ij}\sigma_i\sigma_j - h\sum_i \sigma_i, \tag{17}$$

where we have also included an external magnetic field. The interaction J_{ij} is again often (but not always) non-zero only between nearest neighbors. Ising couplings can arise in a system of $S = 1/2$ quantum spins when anisotropies make the interactions in one spin direction dominant, e.g., only $S_i^z S_j^z$ may have to be considered. There is also a plethora of other physical situations that can be mapped onto Ising models with various forms of the interaction J_{ij} and the field h in Eq. (17), e.g., binary alloys (where σ_i correspond to the two species of atoms) and atoms adsorbed on surfaces (where σ_i correspond to the presence of absence of an atom on a surface).

Considering nearest-neighbor interactions only and zero external field, the energy is

$$E = J\sum_{\langle i,j\rangle} \sigma_i\sigma_j, \tag{18}$$

where $\langle i, j\rangle$ denotes a pair of nearest-neighbor sites i, j. In sums like thse one normally counts each interacting spin pair only once, i.e., if the term $\langle i, j\rangle$ is included in the sum, the term $\langle j, i\rangle$ is not. Denoting by σ the whole set of spin configurations $\{\sigma_1, \ldots, \sigma_N\}$, where N is the total number of spins in the system, the thermal expectation value of a function $A(\sigma)$ is

$$\langle A\rangle = \frac{1}{Z}\sum_\sigma A(\sigma)\mathrm{e}^{-E(\sigma)/T}, \tag{19}$$

$$Z = \sum_\sigma \mathrm{e}^{-E(\sigma)/T}. \tag{20}$$

For ferromagnetic interactions (i.e., $J < 0$) when $T \to 0$ there are only two contributing spin configurations; those with all spins pointing up or down. For antiferromagnetic interactions ($J > 0$) there

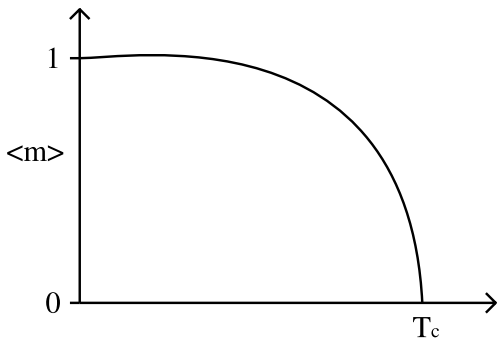

Figure 3: Schematic magnetization versus temperarure curve for a system undergoing a ferromagnetic transition at temperature $T = T_c$.

are also two lowest-energy configurations if the lattice is *bipartite*, i.e., if the system can be subdivided into two sublattices such that all interacting paris $\langle i, j \rangle$ have one member on each sublattice. For example, on a two-dimensional square lattice the lowest-energy configurations have alternating up and down spins in a checkerboard pattern (the up and down spins form the two sublattices). The two-dimensional ferromagnetic and antiferromagnetic ground states are illustrated in Fig. 2. In this case (and for all other bipartite lattices) there is a simple mapping between the antiferromagnet and the ferromagnet; by performing a spin rotation $\sigma_i \to -\sigma_i$ on one of the sublattices we effectively achieve $J \to -J$, and hence all the properties of these two models at any T/J are simply related to each other. This is not the case for a non-bipartite lattice, e.g., a triangular lattice, in which case antiferromagnetic interactions are said to be *geometrically frustrated* (meaning that the individual energies of the interacting spin pairs cannot all be simultaneously minimized). The lowest-energy states of the antiferromagnet are then non-trivial, while the completely polarized states remain the lowest-energy states of the ferromagnet. Frustrated antiferromagnets are of great interest in current magnetism research, but we will here for simplicity consider mainly ferromagnetic interactions.

In dimensions $d > 1$, the Ising model exhibits a phase transition between a disordered (paramagnetic) state at high temperatures and an ordered (ferromagnetic) state at low temperatures (in one dimension, thermal fluctuations prohibit order at $T > 0$, and the system then exhibits true long-range order only exactly at $T = 0$). The *order parameter* of this phase transition is the magnetization,

$$m = \frac{1}{N} \sum_{i=1}^{N} \sigma_i. \tag{21}$$

Fig. 3 shows the expectation value of the magnetization versus the temperature for an infinitely large ferromagnet, in which the spin-reversal symmetry can be broken, i.e., if the system has been prepared with the spins predominantly in one direction below $T < T_c$ it cannot spontaneously (in a finite time) fluctuate through a series of local spin flips into the phase with the opposite magnetization.[1] In a finite system such fluctuations are possible and then $\langle m \rangle = 0$ for all T. In simulations of finite lattices one can instead define the order parameter as $\langle |m| \rangle$ or $\sqrt{\langle m^2 \rangle}$; such magnetization curves will be smooth and non-zero for all T. As the system size is increased the

[1] The symmetry breaking can be eachieved by applying a weak magnetic field, which is removed once equilibrium has been reached. The system will order also spontaneously, without an external symmetry-breaking field, through fluctuations into a state with excess magnetization in one direction, but then large domains with different magnetizations will typically from.

magnetization sharpens close to the critical temperature and approaches the infinite-size symmetry-broken $\langle m(T)\rangle$, which has the form $\langle m(T)\rangle \sim (T_c - T)^\beta$ for $T \to T_c$ from below. Here β is an example of a *critical exponent*. In subsequent sections we will discuss Monte Carlo simulation algorithms for the Ising model and learn how to extract transition temperatures and critical exponents from simulation data. Before that, we will consider the general conditions for importance sampling according to a desired distribution.

3 Importance sampling and detailed balance

We will here consider a discrete configuration space $\{C\} = C_1, C_2, \ldots, C_{\mathcal{N}}$ (where $\mathcal{N}$ can be finite or infinite), but the discussion can be directly generalized to a continuum of configurations as well (we will mention an example of this as well). For a system at temperature T, an expectation value is given by

$$\langle A\rangle = \sum_i P(C_i)A(C_i), \quad P(C_i) = \frac{1}{Z}\mathrm{e}^{-E(C_i)/T}, \tag{22}$$

where we work in units such that $k_B = 1$ (i.e., we measure energies in degrees Kelvin). In a simulation we start with some arbitrary configuration $C_{i(0)}$ and from it stochastically generate a sequence $C_{i(1)}, C_{i(2)}, \ldots, C_{i(M)}$, which we use to approximate various expectation values of interest. Our goal is for the configurations to be distributed according to P.

We use some stochastic process in which a configuration $C_{i(k+1)}$ is obtained from the previous configuration $C_{i(k)}$ by making some kind of random change in the latter. We consider a sequence of configurations constituting a *Markov chain*, i.e., the probability of making a transition from $C_{i(k)}$ to $C_{i(k+1)}$ is not dependent on how we arrived at $C_{i(k)}$ (its history). We will discuss conditions on the transition probabilities $P(C_i \to C_j)$ for the desired distribution $P(C)$ to be achieved. It should be noted that P can be any probability distribution; not necessarily the Bolzmann probability that we are interested in here.

Instead of considering a single sequence of configurations, it is useful to first imagine an *ensemble* of a large number of configurations. If this ensemble is distributed according to P, then the number $N_0(C_i)$ of configurations C_i in the ensemble is proportional to $P(C_i)$. At a given time (step) we apply some scheme to change (update) the configurations, with the probability of changing C_i to C_j denoted $P(C_i \to C_j)$. The number of configurations C_i after updating all the configurations is

$$N_1(C_i) = N_0(C_i) + \sum_{j\neq i}[N_0(C_j)P(C_j \to C_i) - N_0(C_i)P(C_i \to C_j)], \tag{23}$$

where the two terms for each j in the sum correspond to the number of configurations that were changed into and out of C_i, respectively. This is called the *master equation*. If we want the ensemble to remain distributed according to P, we clearly must have, for all $i = 1, \ldots, \mathcal{N}$,

$$\sum_{j\neq i}[N_0(C_j)P(C_j \to C_i) - N_0(C_i)P(C_i \to C_j)] = 0, \tag{24}$$

or, since $N_0(C_i) \propto P(C_i)$,

$$\sum_{j\neq i}[P(C_j)P(C_j \to C_i) - P(C_i)P(C_i \to C_j)] = 0. \tag{25}$$

This equation may have many solutions, and in general it would be very difficult to find all solutions. However, we can obtain a particular solution by satisfying the above condition term-by-term;

$$P(C_j)P(C_j \to C_i) - P(C_i)P(C_i \to C_j) = 0, \tag{26}$$

which gives a condition, called *detailed balance*, on the ratio of the transition probabilities;

$$\frac{P(C_i \to C_j)}{P(C_j \to C_i)} = \frac{P(C_j)}{P(C_i)}. \tag{27}$$

In statistical mechanics the configuration probabilities $P(C_i)$ is given by

$$P(C_i) = \frac{1}{Z}W(C_i), \qquad W(C_i) = \mathrm{e}^{-E(C_i)/T}, \tag{28}$$

where $W(C_i)$ is referred to as the configuration weight. Since the partition function cancels in the ratio on the right hand side of Eq. (27) we can also write

$$\frac{P(C_i \to C_j)}{P(C_j \to C_i)} = \frac{W(C_j)}{W(C_i)}, \tag{29}$$

which is the way the detailed balance condition normally is written.

Although we have derived the detailed balance condition starting from an ensemble of configurations, it is valid for a single Markov chain as well. This statement would clearly be true if the Markov chains formed by the time evolution of all of the individual configurations in the ensemble would have the same distribution over time, in which case they clearly all would be distributed according to P. For this to hold the sampling has to be *ergodic*, i.e., the types of updates made must be such that from an arbitrary configuration any configuration can be reached by a series of updates. Most Monte Carlo simulation schemes are based on the principles of detailed balance and ergodicity.

We have shown that detailed balance maintains the desired distribution of configurations if we start from an ensemble that is already in that distribution. In practice one starts a Markov chain from some arbitrary state, which may be an improbable state of the target distribution. It will then take some time before the generated configurations are correctly distributed, but with detailed balance and ergodicity fulfilled we are guaranteed to reach the correct distribution after some time. This can be seen in the master equation (23), where it is clear that if we have an ensemble with an excess of configurstions C_i (implying an over-all deficit of other onfigurations), then after one updating step the excess is reduced because when $N_0(C_i)$ is large there are also more configurations changing out of C_i than into it. The time needed for *equilibration* depends on the system under study and one should make sure that a sufficient number of updates are carried out before the configurations are used to measure observables.

In a simulation, one typically does not consider all possible transitions $C_i \to C_j$, but only a subset of all transitions corresponding to making certain small changes in C_i. For example, for an Ising configuration with N spins one can consider flipping a randomly selected spin, in which case N new configurations can be reached. For a system of particles in continuous space, one can consider moving a randomly chosen particle by a displacement vector $\vec{\delta}$, with $\vec{\delta}$ chosen randomly within a sphere of radius Δ. These updates are illustrated in Fig. 4; these clearly constitute erogidic processes as we can create any configuration by repeating the update many times.

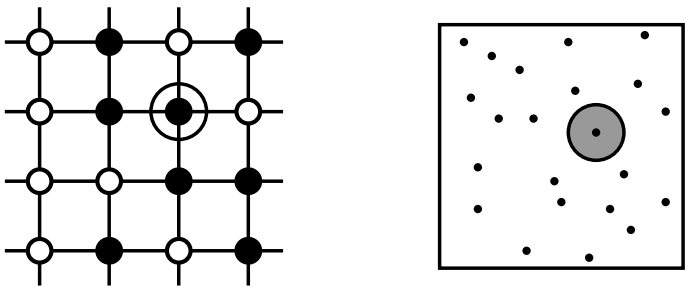

Figure 4: Updating attempts for the Ising model (left), where a spin to be flipped is selected at random (here the one indicated by a circle), and for particles occupying a volume (right), where a particle to be moved is selected at random and its new position is selected randomly within a sphere surrounding it (indicated by a gray circle).

The transition probability $P(C_i \to C_j)$ in the examples given above can be written as a product of two probabilities; one for attempting a certain update (selection of the spin to be flipped, or the particle to be moved and the displacement vector $\vec{\delta}$) and one for actually carrying out the change (accepting it). We hence write

$$P(C_i \to C_j) = P^{\text{attempt}}(C_i \to C_j) P^{\text{accept}}(C_i \to C_j). \tag{30}$$

It is often the case, as it is in the examples mentioned above, that the probability of attempting each of the possible updates is trivially uniform, i.e., $P^{\text{attempt}}(C_i \to C_j) = \text{constant}$, independent of i, j. This part of the transition probability then drops out of the detailed balance condition (29) and we are left with a detailed-balance condition for the acceptance probabilities;

$$\frac{P^{\text{accept}}(C_i \to C_j)}{P^{\text{accept}}(C_j \to C_i)} = \frac{W(C_j)}{W(C_i)}. \tag{31}$$

This condition can be fulfilled in a number of ways, among which the most commonly used is the *Metropolis acceptance probability*;

$$P^{\text{accept}}(C_i \to C_j) = \min\left[\frac{W(C_j)}{W(C_i)}, 1\right]. \tag{32}$$

In other words, if the new configuration weight is higher (corresponding to lowering the energy of the system) we always accept the update, whereas if it is lower we accept it with a probability equal to the ratio of the new and old weights. It can easyly be checked that this Metropolis acceptance probability indeed satisfies the detailed balance condition (31). To determine whether or not to accept the update when $P^{\text{accept}}(C_j) < 1$, the acceptance probability can be compared with a ranom number $r \in [0, 1)$; if $r < P^{\text{accept}}(C_i \to C_j)$ the update is accepted, and otherwise it is rejected. If an update is rejected, the old configuration C_i should be considered the next configuration in the sequence. The whole procedure of attempting updates and accepting or rejecting them using the above scheme goes under the name of the *Metropolis algorithm*, after the first author of the paper where this method was first introduced.[2]

[2] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, *Equations of state calculations by fast computing machines*, J. Chem. Phys. **21**, 1087 (1953). This paper is recommended reading!

Another often used acceptance probability that can be used with the Metropolis algorithm is

$$P^{\text{accept}}(C_i \to C_j) = \frac{W(C_j)}{W(C_i) + W(C_j)}, \tag{33}$$

which is a special case of a *heat bath* probability involving selection among a number m of choises

$$P^{\text{select}}(C_{j(k)}) = \frac{W(C_{j(k)})}{\sum_{l=1}^{m} W(C_{j(l)})}. \tag{34}$$

Here the current configuration is $C_i = C_{j(l)}$ for some $l \in \{1, \ldots, m\}$ and there is no explicitly rejected update, i.e., one of the options $l = 1, \ldots, m$ is always chosen, according to the above probabilities. Since one of the choises equals the old configuration the update may still lead to no change. This approach is useful, e.g., for lattice models where each lattice site can be in $m > 2$ different states. The acceptance probability (33) can be considered a special case of the heat-bath approach when there are just two states to be selected among.

4 Metropolis algorithm for the Ising model

It was already indicated above how the Metropolis algorithm works in the case of the Ising model, for which the energy in the presence of a magnetic field is given by Eq. (17); a configuration update amounts to selecting a spin at random and flipping it with probability (32). When updating an Ising configuration; $C \to C'$, by flipping any number of spins, the weight ratio $W(C')/W(C)$ in the acceptance probability is given explicitly by

$$\frac{W(C')}{W(C)} = \exp\left[-\frac{J}{T}\sum_{\langle i,j\rangle}(\sigma_i'\sigma_j' - \sigma_i\sigma_j) + \frac{h}{T}\sum_i(\sigma_i' - \sigma_i)\right], \tag{35}$$

where $\{\sigma_i'\}$ are the spins of the updated configuration. Flipping a single spin j, we get

$$\frac{W(C')}{W(C)} = \exp\left[\frac{2J}{T}\sigma_j\left(\sum_{\delta[j]}\sigma_{\delta[j]} - \frac{h}{J}\right)\right], \tag{36}$$

where $\delta[j]$ denotes a nearest neighbor of site j (of which there are $2d$ on a d-dimensional cubic lattice). Since the accept/reject criterion in practice amounts to comparing the above ratio with a random number $0 \le r < 1$, these ratios can be used directly without taking the minimum with 1, which required in the actual probability (32). In order to avoid repeated time-consuming evaluations of exponential functions, the weight ratios should be precalculated and stored in a table.

It should be pointed out that it is actually not necessary to select the spin to be flipped at random; one can also go through all spins one-by-one. In this case detailed balance is not fulfilled for each step, but with some more effort one can show that the correct distribution is nevertheless obtained. It is likely, however, that the random spin selection makes the simulation less sensitive to flaws in the random number generator (whether or not this statement really holds clearly depends on the random number generator used) and hence this is the preferred way to do it.

4.1 Program implementation

The ising spins $\sigma_i = \pm 1$ can be stored in a one-dimensional vector; typically it is best to start the indexing from 0, i.e., to allocate `spin(0:n-1)`. For a simple cubic lattice with dimensionality $d = 1, 2$, or 3, and the number of spins $N = L^d$, the correspondence between the index `i` and the coordinates of the lattice sites can then be conveniently chosen as `x=i` for $d = 1$, `x=mod(i,l)`, `y=i/l` for $d = 2$, and `x=mod(i,l)`, `y=mod(i,l**2)/l`, `z=i/l**2` for $d = 3$. The neigbors $\delta[i]$ can be easily calculated on the run using these coordinates. For more complicated non-cubic lattices it may be better to precalculate the neighbors and store them in an array.

It is useful to define a size-normalized "time" unit of a simulation, so that the probability of carrying out an update of a spin during a time unit is independent of the system size. Hence we define a *Monte Carlo step* as N attempts to flip randomly selected spins.

As an example we here consider the two-dimensional ferromagnetic Ising model with `n=lx*ly` spins and $J = -1$. We discuss the main elements of the program 'ising2d.f90' which is available on the course web site. The spins `spin(s)` can be initially set to arbitrary values ± 1, e.g., chosen at random. The precalculated weight ratios (36) are stored in a matrix `pflip(s0,ss)` where the indexes `s0` and `ss` correspond to the values of $\sigma_j = \pm 1$ and $\sum_{\delta[j]} \sigma_{\delta[j]} = -4, -2, 0, 2, 4$, respectively (since we work on a periodic lattice in which each spin has four neighbors, the sum is always even). We use a simple array (which has more elements than needed); with `S0=-1,0,1` and `ss=-4,-3,...,4`. With the temperature stored in a floating-point variable `temp` the probability matrix can be constructed by

```
do ss=-4,4,2
   pflip(-1,ss)=exp(+(ss+h)*2./temp)
   pflip(+1,ss)=exp(-(ss+h)*2./temp)
end do
```

Using the Fortran 90 intrinsic random number subroutine `random_number(r)`, a Monte Carlo step can be carried out as follows;

```
do i=0,n-1
   call random_number(r); s=int(r*n)
   x=mod(s,lx); y=s/lx
   s1=spin(mod(x+1,lx)+y*lx)
   s2=spin(x+mod(y+1,ly)*lx)
   s3=spin(mod(x-1+lx,lx)+y*lx)
   s4=spin(x+mod(y-1+ly,ly)*lx)
   call random_number(r)
   if (r<pflip(spin(s),s1+s2+s3+s4)) spin(s)=-spin(s)
end do
```

A full simulation consists of a number of equilibration steps and a much larger number of steps after each of which measurements of physical quantities of interest are carried out (however, in some cases it may not be statistically worthwile to measure after each step, as we will discuss further

below in connection with autocorrelation functions). The main part of the simulation should be subdivided into bins, for which separate averages are calculated that can be subsequently used for calculating statistical errors (in the same way as we discussed in the chapter on Monte Carlo integration). The statistical analysis can be carried out using a separate program; hence all the bin averages are stored on disk (this is useful if one later wants to increase the number of bins without having to redo the bins already completed). The main part of a simulation program can then look like this:

```
do i=1,binsteps
   call mcstep
end do
do j=1,bins
   call resetdatasums
   do i=1,binsteps
      call mcstep
      call measure
   end do
   call writebindata(binsteps)
end do
```

where equilibration is done for a number of steps corresponding to one bin and the subroutine names are self-explanatory (with the exception perhaps of `resetdatasums`, which sets to zero all the variables used to accumulate measurements). As a general rule, the number of bins should be at least 10 in order to calculate the statistical errors reliably. In order not to generate unnecessarily large data files, one should adapt the number of steps per bin so that the number of bins required in order to reach a satisfactory statistical error is not too large (how many bins are too many is somewhat a matter of taste, but to have more than 1000 bins would generally be considered excessive, unless one has some specific reason for this, e.g. to construct high-precision histograms of bin averages).

We discussed data binning also in the context of Monte Carlo integration, where its purpose was to obtain data following a normal distribution (which is the case in the limit of a large number samples per bin). In Monte Carlo simulations based on a Markov chain, where by construction the configurations are time-correlated, an additional purpose of binning is to achieve statistically independent data; this will also be the case if the bins are sufficiently long; a statement which can be made more precise in terms of autocorrelation functions, which we will discussed below in Sec. 5.

4.2 Measuring physical observables

A quantity of natural interest in the context of the ferromagnetic Ising model is the magnetization, which is the order parameter of the phase transition occuring at a temperature $T_c > 0$ in two and three dimensions. We denote by M the full magnetization and by m the corresponding size-normalized quantity;

$$M = \sum_{i=1}^{N} \sigma_i, \qquad m = \frac{M}{N}. \tag{37}$$

As we discusseed in Sec. 2.2, on a finite lattice the spin-reversal symmetry is not broken in a simulation running for a long time and hence $\langle m \rangle = 0$ (although in practize the time to "tunnel" between states with positive and negative magnetization becomes very long in Metropolis simulations of large systems when $T < T_c$; in practice one will then measure $\langle m \rangle \neq 0$). One can instead measure $\langle |m| \rangle$ or $\sqrt{\langle m^2 \rangle}$, since in the thermodynamic limit the become equal to the symmetry-broken $\langle m \rangle$. Another quantity of great interest is the magnetic susceptibility, defined as

$$\chi = \frac{d\langle m \rangle}{dh}, \tag{38}$$

i.e., the linear response of the magnetization to a unifrom magnetic field. In the thermodynamic limit, the susceptibility of a system in zero field ($h = 0$) diverges at the ferromagnetic transition. We can take the derivative in the statistical expression for $\langle m \rangle$ to obtain an estimator for χ. Writing the energy (17) as $E = E_0 - hM$, we have

$$\langle m \rangle = \frac{1}{Z} \sum_S m \mathrm{e}^{-(E_0 - hM)/T}, \qquad Z = \sum_S \mathrm{e}^{-(E_0 - hM)/T}. \tag{39}$$

The susceptibility (38) is then

$$\chi = -\frac{dZ/dh}{Z^2} \sum_S m \mathrm{e}^{-(E_0 - hM)/T} + \frac{1}{Z} \frac{1}{T} \sum_S m M \mathrm{e}^{-(E_0 - hM)/T}, \tag{40}$$

and with

$$\frac{dZ}{dH} = \frac{1}{T} \sum_S M \mathrm{e}^{-(E_0 - hM)/T}, \tag{41}$$

we obtain

$$\chi = \frac{1}{N} \frac{1}{T} \left(\langle M^2 \rangle - \langle M \rangle^2 \right). \tag{42}$$

This expression can be used for $h \neq 0$ at any T. In the zero field case $\langle M \rangle = 0$ for $T \geq T_c$ and thus

$$\chi = \frac{1}{N} \frac{1}{T} \langle M^2 \rangle, \qquad (h = 0, T \geq T_c). \tag{43}$$

For $T < T_c$ we can again substitute $|M|$ for M in (42) and use

$$\chi = \frac{1}{N} \frac{1}{T} \left(\langle M^2 \rangle - \langle |M| \rangle^2 \right). \tag{44}$$

This formula can also be used for $T > T_c$, since there $\langle |M| \rangle \to 0$ as $N \to \infty$, and hence the two expressions (43) and (44) are equivalent in the thermodynamic limit.

The specific heat can be derived in a similar way;

$$C = \frac{1}{N} \frac{dE}{dT} = \frac{1}{N} \frac{d}{dT} \frac{1}{Z} \sum_C E(C) \mathrm{e}^{-E(C)/T} = \frac{1}{N} \frac{1}{T} \left(\langle E^2 \rangle - \langle E \rangle^2 \right). \tag{45}$$

Like the susceptibility, the specific heat is singular at the ferromagnetic phase transition. Another quantity of interest is the spin-spin correlation function

$$C(\vec{r}_j - \vec{r}_i) = \langle \sigma_i \sigma_j \rangle - \langle \sigma_i \rangle^2, \tag{46}$$

which can be averaged over pairs of sites with the same separation $\vec{r}_j - \vec{r}_i$. For $T \neq T_c$ the correlation function decays exponentially to zero as $|\vec{r}| = r \to \infty$; $C(\vec{r}) \sim \mathrm{e}^{-r/\xi}$, where ξ is the correlation length. As $T \to T_c$ the correlation length diverges, and exactly at T_c the correlation function decays to zero as a power-law. We will discuss these issues further in Sec. 7.2.

In the example program `ising2d.f90` the energy and the magnetization, as well as their squares, are measured, and their bin averages are stored in a file on disk for later processing. This is the main part of the measurement routine:

```
real(8) :: enrg1,enrg2,magn1,magn2
common/measurments/enrg1,enrg2,magn1,magn2

e=0
do s=0,n-1
   x=mod(s,l); y=s/l
   e=e-spin(s)*(spin(mod(x+1,l)+y*l)+spin(x+mod(y+1,l)*l))
end do
m=m+abs(sum(spin))
enrg1=enrg1+dble(e);  enrg2=enrg2+dble(e)**2
magn1=magn1+dble(m);  magn2=magn2+dble(m)**2
```

After each completed bin, the averages $\langle E\rangle/N$, $\langle E^2\rangle/N$, $\langle|M|\rangle/N$, and $\langle M^2\rangle/N^2$ are written to the file `bindata.dat`. This file should be processed by the program `average.f90`, which calculates the final averages and statistical errors. The susceptibility and the specific heat are computed based on bin averages according to Eqs. (44) and (45).

5 Autocorrelation functions

The Metropolis algorith generates statistically independent configurtions through a series of spin flips. How many Monte Carlo steps are required between two configurations before they can be considered statistically independent? This important question can be answered by studying *autocorrelation functions*. For a quantity Q, the autocorrelation function is defined as

$$A_Q(\tau) = \frac{\langle Q_k Q_{k+\tau}\rangle - \langle Q_k\rangle^2}{\langle Q_k^2\rangle - \langle Q_k\rangle^2}, \tag{47}$$

where the averages are over the time-index k (counting the number of Monte Carlo steps) and the normalization is such that $A_Q(0) = 1$. If the configurations $C_{i(k)}$ and $C_{i(k+\tau)}$ are statistically independent for all k, the correlation function $\langle Q_k Q_{k+\tau}\rangle = \langle Q_k\rangle\langle Q_{k+\tau}\rangle = \langle Q_k\rangle^2$. For an ergodic simulation we expect statistical independence as $\tau \to \infty$, and hence the autocorrelation function (47) should approach 0 as $\tau \to \infty$. For large τ the approach to zero is exponential,

$$A_Q(\tau) \to \mathrm{e}^{-\tau/\Theta}, \tag{48}$$

where Θ is called the autocorrelation time. We can also define an integrated autocorrelation time,

$$\Theta_{\mathrm{int}} = \frac{1}{2} + \sum_{\tau=1}^{\infty} A_Q(\tau), \tag{49}$$

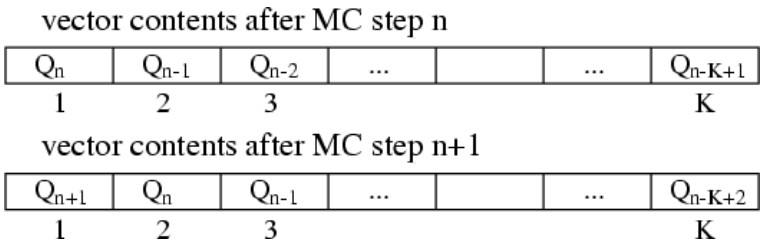

Figure 5: Contents after Monte Carlo steps n and $n+1$ of the vector `tobs()` used for storing individual measurements of a quantity Q.

which corresponds to the trapezoidal integration formula in which the term $1/2$ comes from $A_Q(0)/2$. For a purely exponential autocorrelation function we would have $\Theta_{\mathrm{int}} = \Theta$ (strictly only for large Θ, in which case the discrete sum approaches the integral of $\mathrm{e}^{-\tau/\Theta}$), but the short-time behavior is typically of a different form and then the two autocorrelation times differ. The autocorrelation times also depend on what quantity Q is considered. There is, however, some intrinsic slowest fluctuation mode of the system, which depends on the physics of the model as well as on the simulation algorithm used. The presence of this mode is reflected in most measured quantities, which hence share the same asymptotic autocorrelation time Θ. The slowest mode corresponds to fluctuations of the largest structures in the system (ordered domains), and hence it will be reflected most strongly in quantities sensitive to changes in this structure, e.g., the order parameter close to a phase transition and the closely related long-distance correlation function. Quantities that do not depend strongly on the large-scale structure (e.g., short-distance correlation functions) may have much faster decaying autocorrelations, and the eventual long-time decay governed by the slowest mode may then be difficult to detect.

In order to calculate the autocorrelation function up to a Monte Carlo step separation (time) $\tau = K$, we need to store $K+1$ consecutive measurements Q_k. Storing these in a a vector `tobs()`, we first have to fill all the elements during $K+1$ steps, and after that we can begin accumulate contributions to the averages $\langle Q_k Q_{k+\tau} \rangle$ and $\langle Q_k \rangle$. After each Monte Carlo step the contents of the vector have to be shifted, so that a new measurement can be inserted as the first element. This is illustrated in Fig. 5. The following section of code carries out the shift of the elements in the storage vector and the measurements of the autocorrelation function:

```
do t=2,k+1
   tobs(t)=tobs(t-1)
end do
tobs(1)=q
do t=0,k
   acorr(t)=acorr(t)+tobs(1)*tobs(1+t)
end do
```

On the course web site, there is a version `ising2d_a.f90` of the 2D Ising simulation program which carries out autocorrelation function measurements for the energy and the magnetization. The output file of this program should be processed with the program `autoaverage.f90`, which calculates the averages and statistical errors of the binned autocorrelation functions, as well as the corresponding integrated autocorrelation times.

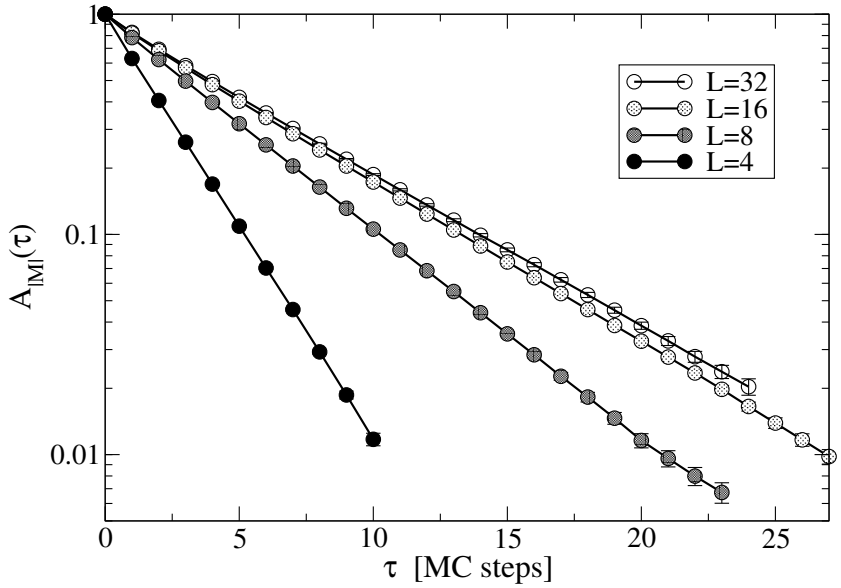

Figure 6: Autocorrelation function for the magnetization magnitude $|M|$ in 2D Ising models of different sizes at $T = 3$ ($> T_c = 2/\ln(1+2\sqrt{2}) \approx 2.269$)

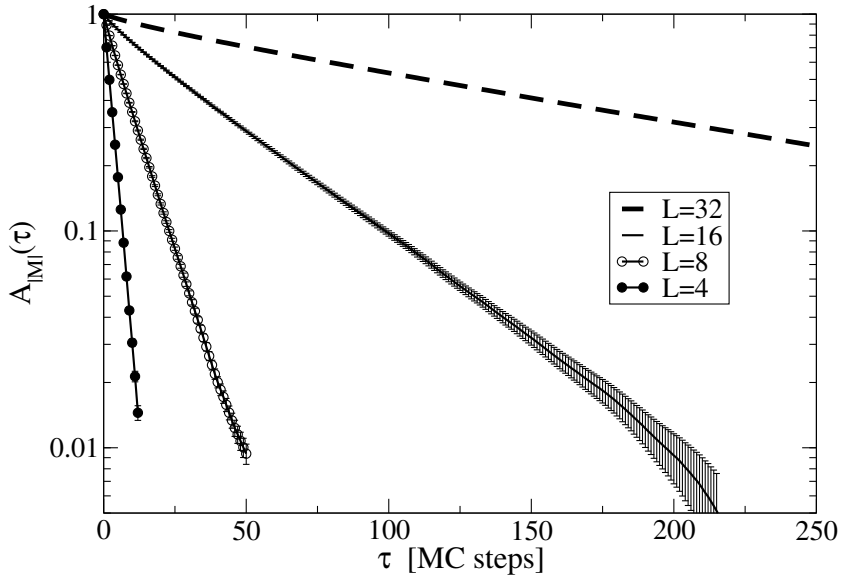

Figure 7: Autocorrelation function for the magnetization magnitude $|M|$ in 2D Ising models of different sizes at $T = 2.269$ ($\approx T_c = 2/\ln(1+2\sqrt{2})$).

Some autocorrelation results obtained using `ising2d_a.f90` are shown in Figs. 6 and 7. Here it can be seen that the autocorrelations for the magnetization depend almost purely exponentially on Monte Carlo time (a linear decay on the linear-log scale used in the figures), with the autocorrelation time (the slope) depending on the system size L. Above T_c, the autocorrelation time converges, whereas close to T_c it apparently continues to grow considerably with L. This phenomenon is called *critical slowing down*. Quite generally, a simulation algorithm utilizing local updates, such as the single-spin flips of the Metropolis algorithm, becomes inefficient close to a critical point, where the divergent correlation length (increasingly large ordered domains) makes the Markov-chain evolution through the contributing part of the configuration space slow; the local spin updates can only slowly change the structure of a configuration containing large ordered domains. This is illustrated in Fig. 8, which shows Ising spin configurations obtained after three consecutive Monte Carlo steps at three different temperatures (in all cases the simulations were first equilibrated). The center row of configurations were generated at $T = 2.5$, which is approximately the temperature at which the size of the ordered domains becomes comparable with the system length for the system size $L = 32$ considered (this temperature can be taken as a critical temperature $T_c(L)$ for a given system size, which is higher than the value in the thermodynamic limit, known exactly from Onsager's solution to be $T_c = 2/\ln(1 + 2\sqrt{2}) \approx 2.269$) Here it can be seen that although a large number of spins have been flipped, the global cluster structure does not change much between the first and third configuration, wheareas below and above T_c the clusters are small; their locations and shapes are then much more easily changed in the Metropolis updating procedure.

Away from the critical point the autocorrelation function (like other properties of the system) will converge when the system size becomes considerably larger than the correlation length. Exactly at the critical point the correlation length diverges and the system length L is the only relevant length scale. In this case the autocorrelation time Θ (as well as Θ_{int}) diverges as a power of the system size;

$$\Theta, \Theta_{\mathrm{int}} \sim L^z, \tag{50}$$

where z is called the dynamic exponent. The integrated autocorrelation times corresponding to the autocorrelation functions shown in Figs. 6 and 7 are graphed in Fig. 9. In the case of the Metropolis algorithm, the dynamic exponent is known to be $z \approx 2.2$ for the 2D Ising model. The results for $T = 2.269 \approx T_c$ are consistent with this.

For calculating Θ_{int}, the summation in Eq. (49) was carried out (by the program `autoaverage.f90`) only up to the τ at which $A_Q(\tau)$ begins to be dominated by statistical errors. This will in principle lead to an underestimated Θ_{int}, but for most purposes it is a sufficiently accurate method.

The effective number of independent configurations generated during M Monte Carlo steps is roughly M/Θ_{int}, and hence it takes increasingly long to generate good statistics in simulations as the autocorrelation time increases. In cases where the autocorrelation time becomes very long, one should also pay attention to the number of steps used in each simulation bin. The statistical analysis based on calculating standard deviations of bin averages assumes that the bin averages constitute independent data, which is only true if the number of Monte Carlo steps per bin is much larger than the autocorrelation time. If this is not the case, the calculated standard deviations of the averages will be smaller than the actual statistical errors.

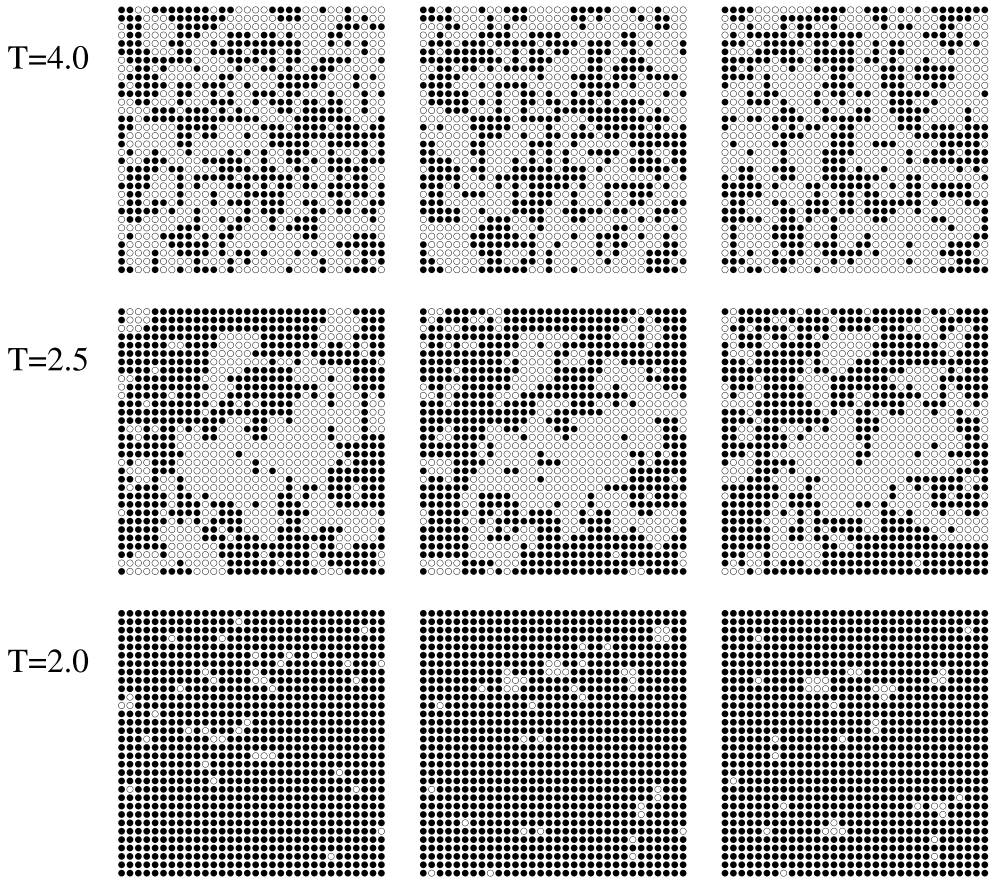

Figure 8: Ising spin configurations obtained with the Metropolis algorithm for two-dimensional $L = 32$ lattices at three different temperatures. The left, center, and right graphs at each T were generated after three consecutive Monte Carlo steps.

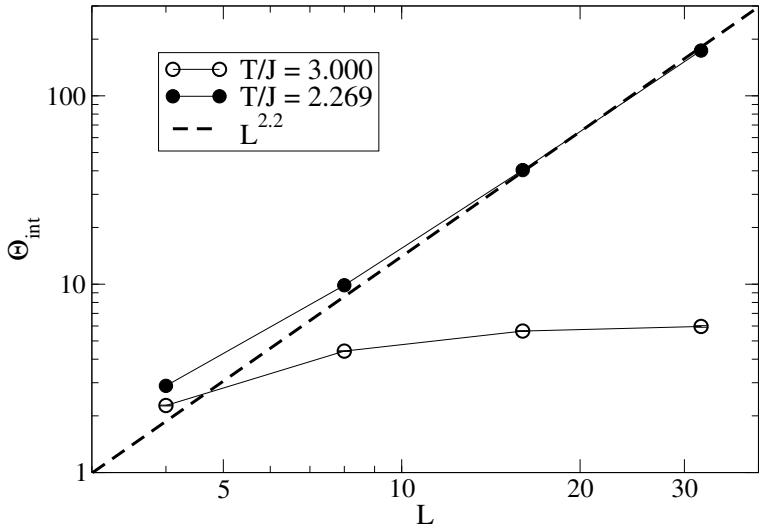

Figure 9: Integrated autocorrelation time versus system size for the 2D ising model at two different temperatures.

6 Cluster algoritms for the Ising model

Critical slowing down severaly hampers studies of systems close to critical points, and even away from critical points the autocorrelation times can be long in many systems. In some cases, these problems can be significantly reduced, or even completely eliminated, using *cluster algorithms*, where a large number of spins can be flipped simultaneously to achieve a faster evolution of the configurations.

It should be noted that simultaneous flips of several spins in the Metropolis method are also possible in principle, but then the weight ratio (35) becomes very small on average, reflecting the fact that the new energy is very likely to go up if several spins are flipped, and very few such updates will be accepted. In a cluster algorithm one constructs clusters of spins in such a way that the whole cluster can be flipped with a high probability (1/2 or 1 depending on the formulation).

We will here be considering only the case of zero magnetic field ($h = 0$ in the Ising energy (17), as the cluster method does not work in practice when $h \neq 0$.

6.1 Swendsen-Wang algorithm

To construct the cluster algorithm for the Ising model first developed by Swendsen and Wang, we introduce a bond index b corresponding to a pair of interacting spins $\sigma_{i(b)}\sigma_{j(b)}$; $b = 1, 2, \ldots, N_b$, where the number of bonds $N_b = dN$ for a d-dimensional cubic lattice with N sites and periodic boundary conditions. We can then write the energy of the ferromagnetic Ising model as,

$$E(\sigma) = -|J| \sum_{b=1}^{N_b} [\sigma_{i(b)}\sigma_{j(b)} + 1] = \sum_{b=1}^{N_b} E_b. \tag{51}$$

A constant $-|J|$ has been added to the energy of each bond, for reasons that will become clear below. Using the bond energies E_b, we can write the partition function as

$$Z = \sum_{\sigma} \mathrm{e}^{-E(\sigma)/T} = \sum_{\sigma} \prod_{b=1}^{N_b} \mathrm{e}^{E_b/T} = \sum_{\sigma} \prod_{b=1}^{N_b} [1 + (\mathrm{e}^{E_b/T} - 1)]. \tag{52}$$

We now define a *bond function* with arguments $0, 1$ corresponding to the two terms on the right hand side above;

$$\begin{aligned} F_b(0) &= 1, \\ F_b(1) &= \mathrm{e}^{E_b/T} - 1, \end{aligned} \tag{53}$$

and write the partition function as

$$Z = \sum_{\sigma} \prod_{b=1}^{N_b} [F_b(0) + F_b(1)]. \tag{54}$$

We now intruduce a set of *auxilary bond variables* $\tau_b = \pm 1$ to be used as arguments in the bond function (53) for each bond b. We use the notation $\tau = \{\tau_1, \tau_2, \ldots, \tau_{N_b}\}$ to refer to a whole *bond*

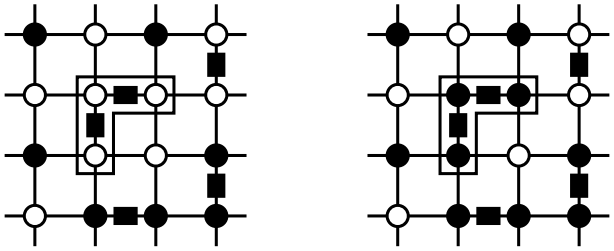

Figure 10: A spin configuration (circles) in which filled bonds (thick lines between circles) have been cast between equal spins according to the appropriate probability. A cluster consisting of three spins is identified and its two orientations are shown. The weight in the partition function is the same for these two configurations.

configuration, in a way analogous to a spin configuration σ. We can now write Z as sum over spins and bonds;

$$Z = \sum_{\sigma} \sum_{\tau} \prod_{b=1}^{N_b} F_b(\tau_b). \tag{55}$$

The bond function F_b depends implicitly on the spins connected by bond b;

$$F_b(0) = 1, \quad \text{independent of } \sigma_{i(b)}, \sigma_{j(b)} \tag{56}$$

$$F_b(1) = \mathrm{e}^{E_b/T} - 1 = \begin{cases} \mathrm{e}^{2|J|/T} - 1, & \text{if } \sigma_{i(b)} = \sigma_{j(b)}, \\ 0, & \text{if } \sigma_{i(b)} \neq \sigma_{j(b)}. \end{cases} \tag{57}$$

For a non-vanishing contribution to the partition function (55), the bond variable $\tau_b = 1$ is hence allowed only between parallel spins; we will refer to $\tau_b = 1$ as a *filled bond*. In the combined space of spins and bonds, the configuration weight in the partition function (55) is

$$W(\sigma, \tau) = \prod_{b=1}^{N_b} F_b(\tau), \tag{58}$$

which if we have no "illegal" filled bonds is simply

$$W(\sigma, \tau) = (\mathrm{e}^{2|J|/T} - 1)^{N_1}, \tag{59}$$

where N_1 is the number of filled bonds. Hence the spin configuration affects the weight only by imposing restrictions on where the filled bonds can be placed. This scheme relies critically on the added constant $-|J|$ in each bond energy in Eq. (51); without this term there could be (with a different probability) filled bonds also between antiparallel spins, and the weight function would have a more complex dependence on this spins. As we will see, the key feature of the Swendsen-Wang scheme is that the weight is exactly zero if a filled bond is placed between antiparallel spins, and that if no such illegal bonds are present the weight is independent on the spin configuration.

The objective now is to construct a scheme for generating spin and bond configurations distributed according to the weight function (58). For a given spin configuration, we define the probability of a bond configuration corresponding to the weight (58);

$$P(\tau) = \prod_{b} P_b(\tau_b), \tag{60}$$

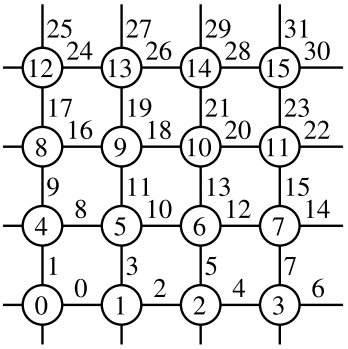

Figure 11: Labeling convention for the spins and bonds of a 2D square lattice (here of size 4×4) with periodic boundary conditions.

where the individual bond probabilities are

$$P(\tau_b) = \frac{F_b(\tau_b)}{F_b(0) + F_b(1)}. \tag{61}$$

The probability of a filled bond is hence

$$P(\tau_b = 1) = 1 - \mathrm{e}^{-2|J|/T}, \quad \text{if } \sigma_{i(b)} = \sigma_{j(b)}, \tag{62}$$

$$P(\tau_b = 1) = 0, \quad \text{if } \sigma_{i(b)} \neq \sigma_{j(b)}. \tag{63}$$

The next key observation underlying the Swendsen-Wang algorithm is that for a configuration of spins and bonds, we can form clusters of spins connected by filled bonds, and if all spins in such a cluster are flipped collectively the weight (59) is unchanged (the number of bonds does not change, and since all spins are equal also in the flipped cluster, no instances of forbidden filled bonds will result). A single spin connected to no filled bond can also be considered a cluster.

We have now completed all steps needed to formulate the cluster algorithm:

1) Start with an arbitrary spin configuration.
2) Cast filled bonds according to the probabilities (62) and (63).
3) Identify all clusters of spins; flip each of them with probability 1/2.
4) Repeat from 2).

Here the reason to flip the clusters with probability 1/2 is that this implies that on average 50% of the spins will be flipped in every step. In principle any probability < 1 could be used (flipping with probability 1 would not be good, as every spin belongs to a cluster and hence all spins would be flipped), but 1/2 can, by symmetry, be expected to be the best choise.

6.2 Program implementation of the Swendsen-Wang method

An important feature to notice is that the Swendsen-Wang algorithm is essentially independent of the dimenionality and the type of lattice; we only need to define the list of interacting sites $[i(b), j(b)]$, a list containing the bonds b connected to a given site j, and also a table of the nearest neighbors of each site. We call the list of sites connected by bonds `spinbond(i,b)`, where the bond

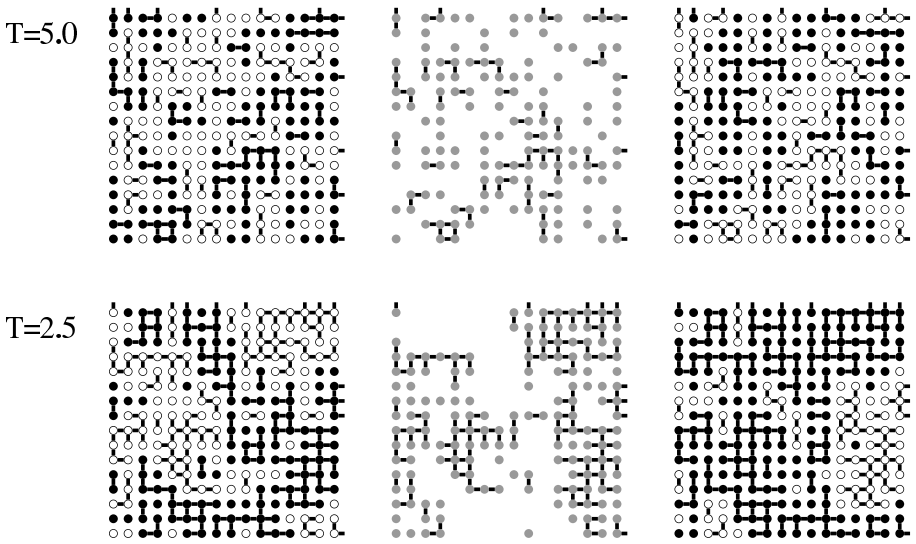

Figure 12: Spin/bond configurations of a 2D Ising model generated with the Swendsen-Wang algorithm at two different temperatures (left). The flipped clusters are indicated by gray sircles (center). The configurations obtained after flipping the clusters are also shown (right).

index `b=0,...,nb-1` and `i=1,2` correspond to the two sites $i(b), j(b)$. The bonds connected to spin `s` are stored as `bondspin(i,s)`, where `i=1,...,nbors`, where `nbors` is the number of neighbors (if the number of neighbors is not the same for every site, e.g., in the case of open boundary conditions, we would also have to store a list of the number of neighbors of each spin). The nearest neighbors of spin s are stored in `neighbor(i,s)`, where again `i=1,...,nbors`. As an example of how to construct these lists, we consider the 2D square lattice with `n=lx*ly` sites, using the labeling convention for the sites and bonds shown in Fig. 11. This code segment carries out the construction of the lattice tables:

```
do s0=0,n-1
   x0=mod(s0,lx)
   y0=s0/lx
   x1=mod(x0+1,lx)
   x2=mod(x0-1+lx,lx)
   y1=mod(y0+1,ly)
   y2=mod(y0-1+ly,ly)
   s1=x1+y0*lx
   s2=x0+y1*lx
   s3=x2+y0*lx
   s4=x0+y2*lx
   neighbor(1,s0)=s1
   neighbor(2,s0)=s2
   neighbor(3,s0)=s3
   neighbor(4,s0)=s4
   bondspin(1,2*s0)=s0
   bondspin(2,2*s0)=s1
```

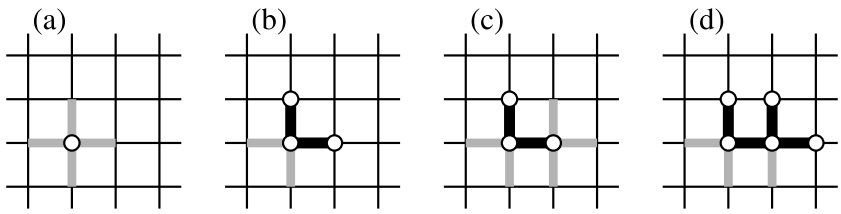

Figure 13: The first steps of building a cluster out of a seed spin which by definition belong to the cluster. In (a) the bonds connected to the seed spin are indicated; a spin at the other end of a bond belongs to the cluster if the bond is a filled one. (b) shows the result of the examination of the bonds, with the filled bonds indicated by thicker black lines. The spins visited (per definition these are spins that are added to the cluster) are shown as circles. (c) indicates the bonds to be examined that are connected to one of the new spins in the cluster, and (d) shows the result of this step.

```
    bondspin(1,2*s0+1)=s0
    bondspin(2,2*s0+1)=s2
    spinbond(1,s0)=2*s0
    spinbond(2,s0)=2*s0+1
    spinbond(3,s1)=2*s0
    spinbond(4,s2)=2*s0+1
end do
```

To generate a bond configuration we need the list `bondspin(i,b)` of spins connected by the bonds. We store the spins ± 1 in an integer vector `spin(0:n-1)` and the bonds in a logical vector `bond(0:nb-1)`, with `.true.` and `.false.` corresponding to filled and empty bonds, respectively. This segment of code casts the filled bonds with a probability `bprob` that has been preset to $1 - \mathrm{e}^{-2|J|/T}$, using a random number generator `rand()`;

```
do b=0,nb-1
   if (spin(bondspin(1,b))==spin(bondspin(2,b))) then
      if (rand()<bprob) then
         bond(b)=.true.
      else
         bond(b)=.false.
      end if
   else
      bond(b)=.false.
   end if
end do
```

The next step is to identify and flip clusters. There are several cluster finding algorithms, the most efficient being the so-called Hoshen-Koppelmann algorithm. Here we will use a simpler method which is only slightly less efficient. The aim is to construct all clusters and to flip each of them with probability 1/2. The search for the first cluster is done using spin 0 as the "seed" spin; all its nearest-neighbor spins connected to it by filled bonds are to be identified, and the neighbors of

those spins must be checked in turn, etc., until no new neighbors of spins that have been identified as belonging to the cluster are connected to it by filled bonds. The first steps of this procedure are illustrated in Fig. 13. Each new spin that is added to the cluster is stored on a stack (a last-in-first-out list); when the bonds of one spin has been examined a new spin is drawn from the stack, until the stack is empty, in which case the cluster is completed. A vector of flags should be kept to indicate whether a spin already has been "visited" or not, i.e., whether or not it has already been added to a cluster. Using these flags, one can make sure that no spin is visited more than once. The decision of whether to flip the cluster or not can be made before the cluster building starts. If the decision is to flip, spins are flipped when they are put on the stack. When a cluster has been completed a new seed site that has not previouisly been visited is searched for using the vector of flags, starting from the previous seed spin position plus one. Calling the flag for spin `s` `notvisited(s)`, with boolean values `.false.` and `.true.` for spins that have and have not been visited, and storing the seed spin in `cseed`, we initialize the cluster search/flip procedure by

```
notvisited(:)=.true.
cseed=0
```

Then the flip decision is made and the first spin is added to the cluster (which corresponds to adding it to the stack), and a loop for building the cluster to completion is executed;

```
1 if (rand()<0.5d0) then
   flipclus=.true.
else
   flipclus=.false.
end if
notvisited(cseed)=.false.
if (flipclus) spin(cseed)=-spin(cseed)
stack(1)=cseed;  nstack=1

do
   if (nstack==0) exit
   s0=stack(nstack);  nstack=nstack-1
   do i=1,nbors
      s1=neighbor(i,s0)
      if (bond(spinbond(i,s0)).and.notvisited(s1)) then
         notvisited(s1)=.false.
         if (flipclus) spin(s1)=-spin(s1)
         nstack=nstack+1; stack(nstack)=s1
      end if
   end do
end do
```

After completion of the cluster (exit from the loop when the staack is empty) we search for a not previously visited spin to be used as the seed of the next cluster;

```
do i=cseed+1,n-1
```

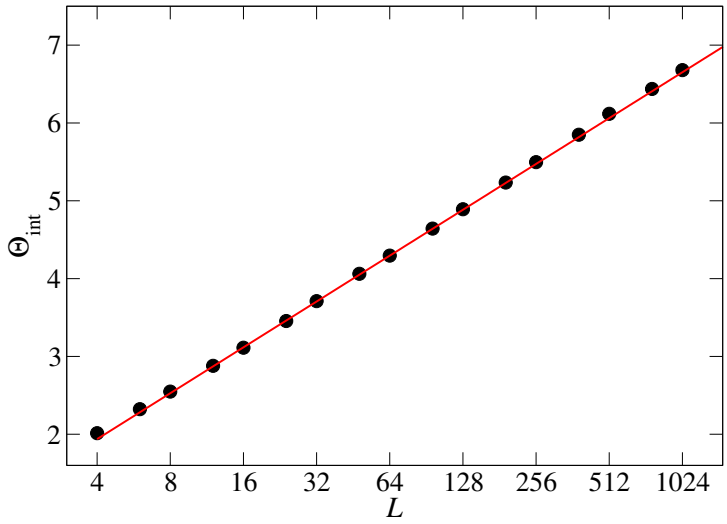

Figure 14: Integrated autocorrelation time of the magnetization of the 2D ising model at the critical temperature; $T_c = 2/\ln(1+2\sqrt{2})$. The system size dependence corresponding to logarithmic divergence is shown as the straight line [$\Theta_{\rm int} = 0.75 + 0.85\ln(L)$].

```
    if (notvisited(i)) then
       cseed=i
       goto 1
    end if
 end do
```

This continues until all sites have been visited, which implies that all clusters have been constructed (as each spin belongs uniquely to one cluster).

We have now covered all the important elements of a Swendsen-Wang simulation. A program, `sw.f90`, using these elements is available on the course web site.

6.3 Autocorrelations

The simulation dynamics of the Swendsen-Wang algorithm has been investigated extensively. While there are some claims that the dynamic exponent is small but not equal to zero, a more likely scenario is that the autocorrelation time diverges logarithmically. This can be considered as $z = 0$ with a logarithmic correction to a size-independent autocorrelation time . Later, we will see that such logarithmic behaviors occur in some physical quantities of the 2D Ising model as well.

With the logarithmic divergence of the autocorrelation time, there is still a mild critical slowing down problem, but the improvement is dramatic compared to the Metropolis algorithm (for which $z \approx 2.2$ in two dimensions, as shown in Fig. 9.

6.4 Wolff algorithm

There is a slightly different variant of the cluster algorithm, called the *Wolff algorithm*, in which only one cluster is constructed at each step and flipped with probability one, using a randomly selected seed spin. The bonds are in this case cast during the cluster building process, i.e., instead of examining whether or not the bonds are filled, they are at that stage filled, with the same probability as in the Swendsen-Wang algorithm. The types of clusters generated this way are naturally exactly the same as in the Swendsen-Wang algorithm, but on average the generated clusters will be larger because the probability of a given spin (the seed spin) belonging to a large cluster is higher than it belonging to a small cluster. This may in some cases further reduce the dynamic exponent slightly. The Wolff scheme can also more easily be generalized to other spin models.

7 Critical phenomena and Finite-size scaling

We here give a brief review of critical phenomena and scaling, and the finite-size scaling properties that can be used to quantitatively study critical phenomena using Monte Carlo simulations. In most cases we will not be deriving results here, but simply define the quantities that we will be studying numerically and discuss their properties. For more background on these issues, consult a book on critical pheonomena, e.g., *Scaling and Renormalization in Statistical Physics*, by J. Cardy (Cambridge University Press).

The 2D Ising model is one of the few models in statistical physics that can be exactly solved and has a non-trivial phase transition. Simulation results for this model (obtained using the Swendsen-Wang program discussed in the previous section) will be used to illustrate the finite-size scaling concepts. Thanks to the known exact results for T_c and the critical exponents, we can check whether the simulation results for large system sizes indeed converge to the correct thermodynamic limit behavior.

7.1 Critical exponents and universality

The most fundamental concept underlying the theory of critical phenomena is that of a correlation length, which is a measure of a typical length-scale of a system. The correlation length can be defined in terms of the correlation function, which in the case of the Ising model is given by (46). The correlation function decays exponentially at long distances, and is given by the so-called *Ornstein-Zernicke* form

$$C(\vec{r}) \sim \frac{\mathrm{e}^{-r/\xi}}{r^{(d-2)/2}}, \tag{64}$$

where ξ is the correlation length. The asymptotic Ornstein-Zernicke form is a result of mean-field theory, but it turns out that it is valid also more generally when $r \gg \xi$; at shorter distances there are corrections to this form. The correlation length ξ roughly corresponds to the typical size of the ordered domains in the system. In the Ising model above T_c there are on average equal numbers of ordered domains with spin up and down, and their typical size corresponds to the correlation length. Below T_c the correlation length corresponds to the typical size of domains of spins in an

ordered background of oppositely directed spins. Examples of Ising configurations illustrating these cases are shown in Fig. 8.

Quite generally, as a critical point (continuous phase transition) is approached, the correlation length diverges, in the thermodynamic limit, according to a power-law;

$$\xi \sim t^{-\nu}, \tag{65}$$

where t is the reduced temperature measuring the distance from the critical point,

$$t = \frac{|T - T_c|}{T_c}, \tag{66}$$

and ν is an example of a critical exponent. The exponent is the same on approaching T_c from above or below [the prefactor in (65) is in general different for $t \to t^+$ and $t \to t^-$, however].

Exactly at the critical point the correlation function decays purely as a power-law, but not with the power that might be expected from Eq. (64). The Ornstein-Zernicke form is valid only when $r \gg \xi$, which is never true at the critical point, where ξ has diverged. The power-law is instead

$$C(\vec{r}) \sim \frac{1}{r^{d-2+\eta}}, \tag{67}$$

where the exponent $\eta = 0$ in mean-field theory. The exact value of η is also typically small.

Another important critical exponent is the one determining the onset of order at the critical point, e.g., the magnetization of a ferromagnet (as illustrated in Fig. 3). The order parameter is zero above T_c, and below T_c it emerges as

$$\langle m \rangle \sim (T_c - T)^{\beta}. \tag{68}$$

The order parameter susceptibility exhibits a divergence at T_c. In the case of the ferromagnet the susceptibility is given by (42), and on approaching T_c from below or above it is given by

$$\chi \sim t^{-\gamma}. \tag{69}$$

Hence, the system becomes infinitely sensitive to a magnetic field h as the critical point is approached, and exactly at T_c the linear response form $\langle m \rangle = \chi h$ ceases to be valid. Instead, exactly at T_c the magnetization depends on the (weak) field as $h^{1/\delta}$, where δ is yet another critical exponent.

The specific heat is also singular, with

$$C \sim t^{-\alpha}. \tag{70}$$

The exponent α can be positive or negative, i.e., in some cases, when $\alpha < 0$, the specific heat does not diverge but instead exhibits a cusp singularity.

The set of exponents $\nu, \eta, \beta, \gamma, \delta, \alpha$ for different systems fall into *universality classes*. Systems belonging to the same universality class have the same exponents. We have here exemplified the critical exponents using the Ising model, but in most continuous phase transitions there is an order parameter, an associated correlation length, and an external field which couples to the order parameter. The universality class does not depend on microscopic details related to the

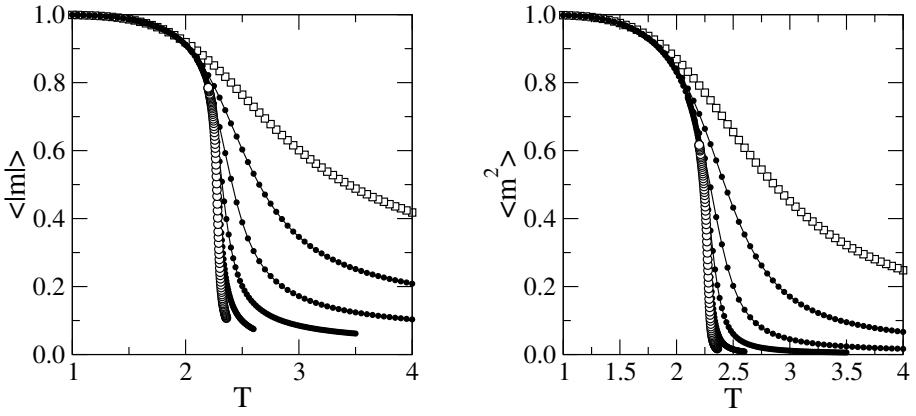

Figure 15: Temperature dependence of the absolute value of the magnetization (left) and its square (right) in $L \times L$ Ising lattices with $L = 4$ (open squares), 8, 16, 32, 64, and 128 (open circles). The statistical errors of these Monte Carlo results are much smaller than the graphing symbols.

system constituents and their interactions (as long as the interactions are short-ranged; long-range interactions can change the universality class), only on the dimensionality of the system and the nature of the order parameter. In the case of the Ising model the exponents are known exactly, and many other universality classes have been studied accurately over the years, using both analytical and numerical approaches. In the majority of cases the most precise estimates come from Monte Carlo simulations.

The critical exponents are not independent of each other, but are given, according to results of the renormalization group theory of critical phenomena, in terms of two more fundamental exponents. This leads to exponent relations, e.g.,

$$\gamma = \nu(2 - \eta), \tag{71}$$
$$\gamma = \beta(d - 1), \tag{72}$$
$$\gamma d = 2 - \alpha, \tag{73}$$
$$\alpha + 2\beta + \gamma = 2, \tag{74}$$
$$\alpha + \beta(1 + \delta) = 2, \tag{75}$$

which were found much before the renormalization group theory using other means. These relations are useful for checking the consistency of numerical results for the exponents.

7.2 Finite-size scaling

In a system of finite size, the correlation length cannot grow beyond the system length L. All divergent quantities (such as χ and C) saturate when ξ approaches L. The order parameter measured using $\langle |m| \rangle$ or $\sqrt{\langle m^2 \rangle}$ cannot vanish for $T \geq T_c$, and the temperature dependence instead shows a rounding over the region in which the infinite-size correlation length exceeds L. This finite-size rounding of the phase transition in the 2D Ising model is illustrated using simulation data in Fig. 15.

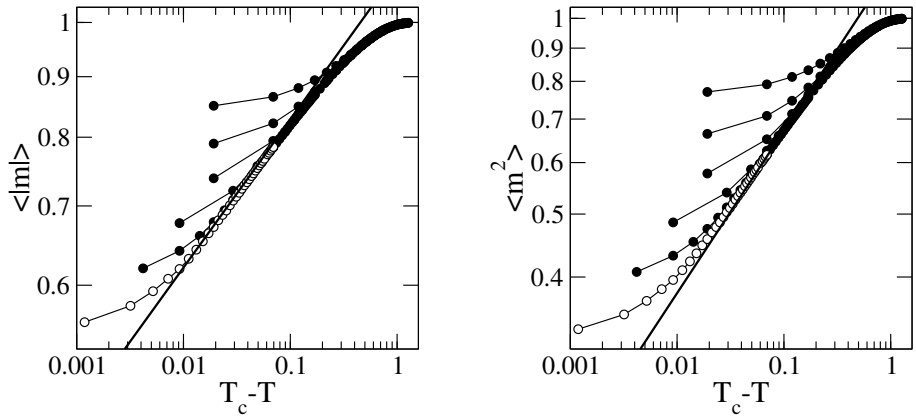

Figure 16: Log-log plot of the temperature dependence of the absolute value of the magnetization (left) and its square (right) of the 2D Ising model at $T < T_c$. The system sizes are the same as in Fig. 15. The straight lines show the expected asymptotic behaviors $\langle |m| \rangle \sim (T_c - T)^{\beta}$ and $\langle |m| \rangle \sim (T_c - T)^{2\beta}$, with the known 2D ising exponent $\beta = 1/8$.

In order to extract critical exponents from Monte Carlo data, one can in principle carry out calculations for increasingly large L until the results converge. For a given reduced temperature $t \neq 0$ there is some system size $L \approx \xi(t)$ above which measured expectation values converge exponentially fast to their thermodynamic limit values. Results for, e.g., $L = 4\xi$ may be in practice (considering statistical errors) indistinguishable from the thermodynamic limit values. By carrying out such size-converged calculations close to T_c, one can extract the critical exponents.

For a quantity A that has a power-law form $A \sim t^{\lambda}$, we have $\ln(A) = c + \lambda \ln(t)$, where c is a constant. Thus the exponent λ can be extracted from simulation results for A by fitting a straight line to $\ln(A)$ versus $\ln(t)$. Since $t \sim |T - T_c|$, such a procedure requires that we know the critical temperature. If we do not know T_c, which is normally the case (finding T_c is often a purpose of the simulations), we can treat it as a variable that is adjusted to give the best linear form $\ln(A) = c + \lambda \ln(t)$ (a rough estimate of T_c can be easily obtained from plots such as those shown in Fig. 15). Since the pure power-law critical form of A is only valid asymptotically as $t \to 0$, one may have to go to very large system sizes in order to obtain high-precision estimate of T_c and the critical exponents.

We can use the exact results for the 2D Ising model; $T_c = 2/\ln(1 + \sqrt{2})$ and the exponents $\nu = 1, \beta = 1/8$, $\gamma = 7/4$, $\alpha = 0$, to test various finite-size approaches. Fig. 16 shows the data of Fig. 15 for $T < T_c$ plotted on a log-log scale, along with the linear behavior expected. For $T_c - T < 0.2$, the expected asymptotic behavior is rather well reproduced, down to the temperatures at which the finite-size rounding sets in. However, even for the largest size, $L = 128$, some deviations can be noticed; there is no extended region where the slope is exactly in agreement with the exact result. With larger system sizes we would find and increasingly good agreement as the finite-size rounding ocuurs closer and closer to T_c.

What we have described above is the most straight-forward finite-size approach. The deviations from scaling due to the finite size were seen as a nuisance that we have to overcome by going to increasingly large system sizes. In a *finite-size scaling* study one instead uses the regularity in these

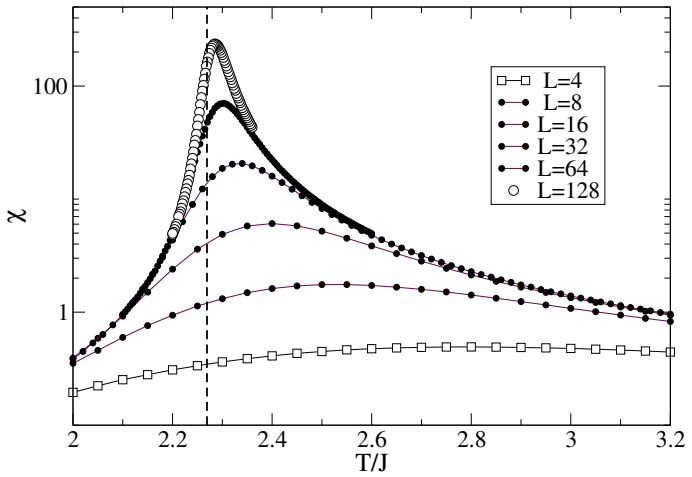

Figure 17: The temperature dependence of the magnetic susceptibility of the 2D Ising model for different lattice sizes. The dashed line indicates the critical temperature.

deviations to extract information.

The basis of finite-size scaling is that deviations from the infinite-size critical behavior occurs when the correlation length ξ becomes comparable with the system length L. The way these finite-size deviations affect other quantities can be studied by expressing their temperature dependencies using the correlation length as the variable, i.e., by inverting Eq. (65) we get;

$$t \sim \xi^{-1/\nu}. \tag{76}$$

For instance, in the asymptotic power-law form of the susceptibility, Eq. (69), the substitution $t \to \xi^{-1/\nu}$ gives

$$\chi \sim \xi^{\gamma/\nu}. \tag{77}$$

From this form we can deduce that the maximum value of the susceptibility for a given system size L should be

$$\chi(L) \sim L^{\gamma/\nu}. \tag{78}$$

From (65) we can also deduce that the reduced temperature at which ξ reaches L is $\sim L^{-1/\nu}$, and hence the maximum value of the susceptibility should occur at

$$t_{\max}(L) \sim L^{-1/\nu}. \tag{79}$$

Fig. 17 shows 2D Ising Monte Carlo results for the susceptibility versus the temperature for the different lattice sizes. Since the peak value of χ grows very rapidly with L, in accord with Eq. (78) where $\gamma/\nu = 7/4$ for the 2D Ising model, it is useful to use a log-scale for χ. Here the shift in the peak temperature with L can also be clearly seen.

If we study a model for which we wish to extract T_c and the exponents, we could write Eq. (78) as $\ln \chi(L) = a + \frac{\gamma}{\nu} \ln(L)$, and hence obtain the exponent ratio γ/ν from a linear fit to the logarithm of the peak height. Adjusting T_c so that $\ln[t_{\max}(L)]$ is a linear function of $\ln(L)$ one can extract ν (and T_c), and hence also γ from the estimated ratio γ/ν.

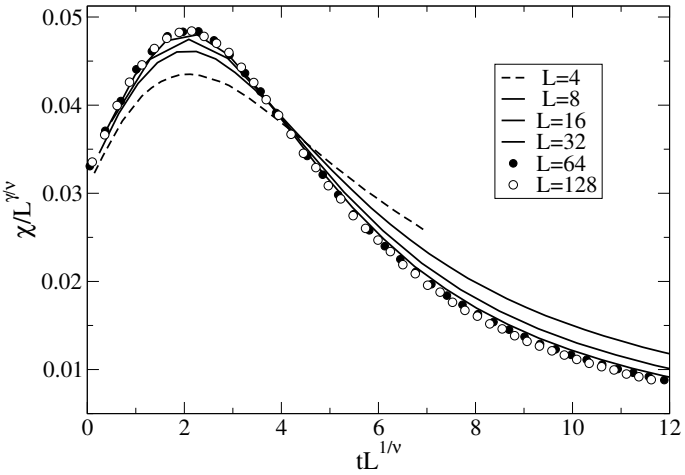

Figure 18: Finite-size scaling plot of the magnetic susceptibility of the 2D Ising model for different system sizes. The exponents $\gamma = 7/4$ and $\nu = 1$.

The above finite-size scaling forms for $\chi_{\max}$ and $t_{\max}$ (which hold generally for a quantity which diverges at T_c) follow also from a more general *finite-size scaling hypothesis*. The hypothesis, which can be proven using the renormalization group theory, is that an observables close to T_c is power of L multiplied by a non-divergent function of ξ/L (the ratio of the two relevant length scales in a finite system), i.e., for the susceptibility we should then have

$$\chi(t, L) = L^{\sigma} f(\xi/L), \tag{80}$$

which, using $\xi \sim t^{-1/\nu}$, we can also write as

$$\chi(t, L) = L^{\sigma} g(tL^{1/\nu}). \tag{81}$$

To determine the exponent σ in the prefactor, we can use the fact that for fixed t close to 0 the infinite-size form has to be $\chi(t, L \to \infty) \sim t^{-\gamma}$. To get this form, the *scaling function* $g(x)$ must have a form $g(x) \sim x^{-\gamma}$ for $x \to \infty$, and $\sigma = \gamma/\nu$, i.e., the finite-size scaling form should be

$$\chi(t, L) = L^{\gamma/\nu} g(tL^{1/\nu}). \tag{82}$$

To extract the scaling function $g(x)$, one can plot $\chi/L^{\gamma/\nu}$ versus $x = tL^{1/\nu}$ for different system sizes. If the scaling hypothesis is correct, data for different (large) system sizes should fall onto the same curve, which then is the scaling function (this is referred to as curves *collapsing* onto each other). Fig. 18 illustrates this with the 2D Ising data of Fig. 17. The curves for large L ($L = 64$ and 128) indeed collapse almost onto each other, whereas this is not the case for the smallest system sizes. Clearly the scaling forms for the peak size, Eq. (78), and the peak position, Eq. (79) are contained in this graph, but these are just two features of the finite-size scaling. In particular, the scaling exactly at the critical temperature can clearly also be used to extract γ/ν and T_c;

$$\chi(T_c, L) \sim L^{\gamma/\nu}, \tag{83}$$

by adjusting T_c so that a linear behavior is seen on a log-log plot.

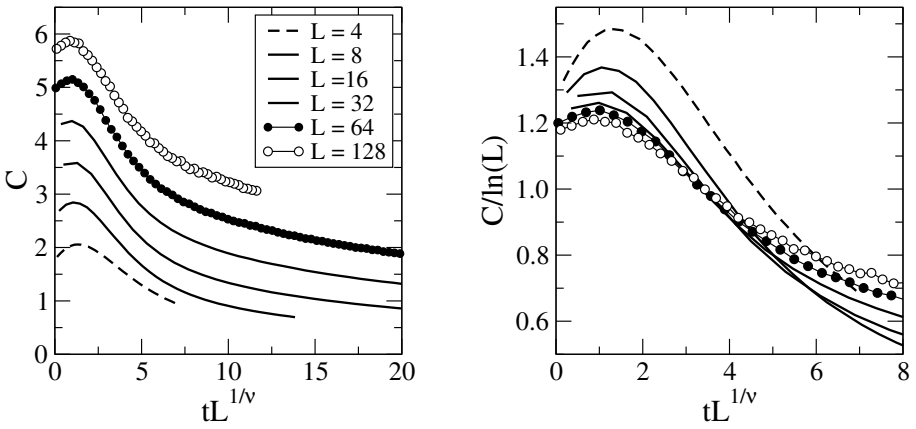

Figure 19: Finite-size scaling of the specific heat of the 2D Ising model. In the left panel C has been divided by L^α, which with the exponent $\alpha = 0$ equals 1. In the right panel C has been divided by $\ln(L)$.

We expect the same type of finite-size scaling to hold also for the specific heat;

$$C(t, L) = L^{\alpha/\nu} g(tL^{1/\nu}). \tag{84}$$

In the case of the 2D Ising model, $\alpha = 0$. The left panel of Fig. 19 shows $C/L^\alpha = C$ graphed versus $tL^{1/\nu}$. The curves do not collapse onto each other and hence the scaling form (84) does not hold. This is understood as being due to a logarithmic scaling when $\alpha = 0$, i.e., (84) should be replaced by

$$C(t, L) = \ln(L) g(tL^{1/\nu}). \tag{85}$$

A scaling plot assuming this form is shown in the right panel of Fig. 19; this indeed leads to an approximate collapse of the curves onto each other as L grows. Normally $\alpha \neq 0$, and the logarithmic scaling is thus an anomaly of the 2D Ising model.

Finite-size scaling studies aimed at extracting critical exponents would clearly be made easier if we had some other method to obtain the critical temperature, i.e., one that would not invole simultaneously adjusting T_c and an exponent until curves collapse or we see linear dependence of a divergent quantity on a log-log scale. This two-parameter scaling can be easily affected by corrections to the leading finite-size scaling forms. A quantity which allows for an independent estimation of T_c without knowledge of exponents will be discussed next.

7.3 Binder ratios

The ratio of two quantities which have the same finite-size scaling exponents at T_c should be size-independent at T_c. The most useful of such ratios are the *Bainder ratios*, which are ratios of powers of the magnetization. From the susceptibility definition (43), it follows that the size-normalized squared magnetization $\langle m^2 \rangle$ scales as $t^{-(\gamma/\nu - d)}$ at T_c. It turns out that any expectation value $\langle m^p \rangle$ simply scales as $\langle m^2 \rangle^{p/2}$ (where in the case of odd p we have to replace m by $|m|$). This implies that the Binder ratios, defined as

$$Q_{2p} = \frac{\langle m^{2p} \rangle}{\langle |m|^p \rangle^2}, \tag{86}$$

should become size independent at T_c for large L. In many cases, the size independence as $T \to T_c$ is approached in such a way that curves $Q(T)$ for different L cross each other, and the crossing point is only weakly dependent on L, being quite close to T_c already for moderate system sizes. Such crossing points are very useful for obtaining precise estimates of T_c. We will here consider the lowest-order binder ratio Q_2, which we simply call Q.

It is instructive to first look at behavior of Q in the limits $T \to 0$ and $T \to \infty$. In the case of the Ising model, the $T = 0$ limit is trivial; here we have $m = \pm 1$, which implies $Q = 1$. In the $T = \infty$ limit, all spin configurations have equal weight in thermal expectation values, and to calculate the Binder ratio we hence have to find the magnetization distribution $P(M)$ for N uncorrelated spins. Denoting the number of up and down spins $N_\uparrow$, $N_\downarrow$, we have $N = N_\uparrow + N_\downarrow$ and $M = N_\uparrow - N_\downarrow$, or

$$\begin{aligned} N_\uparrow &= (N + M)/2 \\ N_\downarrow &= (N - M)/2 \end{aligned} \tag{87}$$

There are in all 2^N spin configurations, and the probability of a configuration with $N_\uparrow$ up spins is

$$P(N_\uparrow) = \frac{1}{2^N} \binom{N}{N_\uparrow} = \frac{1}{2^N} \frac{N!}{N_\uparrow! N_\downarrow!}, \tag{88}$$

and hence the probability as a function of the magnetization is

$$P(M) = \frac{1}{2^N} \frac{N!}{[\frac{1}{2}(N+M)]! [\frac{1}{2}(N-M)!]}. \tag{89}$$

It is clear that this distribution will be sharply peaked around $M = 0$, i.e., the dominant configurations have small M/N, and hence for large N we can use Stirling's formula,

$$n! = \sqrt{2\pi} n^{n+1/2} \mathrm{e}^{-n}, \quad (n \to \infty), \tag{90}$$

for the three factorials. This gives

$$P(M) = \frac{1}{2^N} \frac{1}{\sqrt{2\pi}} \frac{N^{N+1/2}}{[\frac{1}{2}(N+M)]^{\frac{1}{2}(N+M+1)} [\frac{1}{2}(N-M)]^{\frac{1}{2}(N-M+1)}}. \tag{91}$$

We now work with the logarithm of this expression;

$$\begin{aligned} \ln[P(M)] &= -N \ln(2) - \ln(\sqrt{2\pi}) + (N + \tfrac{1}{2}) \ln(N) \\ &\quad -\tfrac{1}{2}(N+M+1) \ln[\tfrac{1}{2}(N+M)] - \tfrac{1}{2}(N-M+1) \ln[\tfrac{1}{2}(N-M)] \\ &= -\ln(2) - \ln(\sqrt{2\pi}) - \tfrac{1}{2} \ln(N) \\ &\quad -\tfrac{1}{2}(N+M+1) \ln(1+M/N) - \tfrac{1}{2}(N-M+1) \ln(1-M/N). \end{aligned} \tag{92}$$

Here we can again use the fact that the distribution should be sharply peaked around $M/N = 0$ and thus we can expand the logarithms;

$$\begin{aligned} \ln[P(M)] &= \ln(2) - \ln(\sqrt{2\pi}) - \tfrac{1}{2} \ln(N) \\ &\quad -\tfrac{1}{2}(N+M+1)\left(\frac{M}{N} + \frac{M^2}{N^2} + \ldots\right) - \tfrac{1}{2}(N-M+1)\left(-\frac{M}{N} + \frac{M^2}{N^2} + \ldots\right). \end{aligned} \tag{93}$$

Keep only terms up to order $1/N$ we get

$$\ln\left[P(M)\right] = \ln\left(2\right) - \ln\left(\sqrt{2\pi}\right) - \tfrac{1}{2}\ln\left(N\right) - \frac{1}{2}\frac{M^2}{N}, \tag{94}$$

and the distribution is thus

$$P(M) = \sqrt{\frac{2}{\pi N}}\mathrm{e}^{-M^2/2N}. \tag{95}$$

For the Binder ratio Q, we need the expectation values $\langle|m|^n\rangle = \langle|M|^n\rangle/N^n$, with $n = 1, 2$;

$$\langle|M|^n\rangle = P(0) + 2\sum_{M=1}^{N} M^n P(M). \tag{96}$$

Assuming that N is even, the sum only contains even values of M. Converting the sum to an integral for large M we therefore get

$$\langle|M|^n\rangle = \sqrt{\frac{2}{\pi N}}\int_0^N M^n \mathrm{e}^{-M^2/2N}. \tag{97}$$

Since $P(M)$ is sharply peaked at $M = 0$, we can extend the upper integration limit to ∞ and use the definite integrals

$$\int_0^\infty dx \mathrm{e}^{-ax^2} = \frac{1}{2}\sqrt{\frac{\pi}{a}}, \tag{98}$$

$$\int_0^\infty dx x \mathrm{e}^{-ax^2} = \frac{1}{2a}, \tag{99}$$

$$\int_0^\infty dx x^2 \mathrm{e}^{-ax^2} = \frac{1}{4a}\sqrt{\frac{\pi}{a}}. \tag{100}$$

The result for $n = 0$ confirms that the distribution $P(M)$ is normalized. For $n = 1$ and 2 we obtain

$$\langle|M|\rangle = \sqrt{2N/\pi}, \tag{101}$$

$$\langle M^2\rangle = N, \tag{102}$$

and hence the Binder ratio in the high-temperature limit is

$$Q = \frac{\langle m^2\rangle}{\langle|m|\rangle^2} = \frac{\langle M^2\rangle}{\langle|M|\rangle^2} = \frac{\pi}{2}. \tag{103}$$

Fig. 20 shows results for the 2D Ising model; these results confirm the high and low temperature limits $Q = 1$ and $\pi/2$, and also show that the Binder ratio is a monotonic function. The curves for different L cross each other very close to T_c. The crossing points, looking at, e.g., crossings of curves for L and $2L$, do show some L dependence, but the convergence is clearly very rapid compared to, e.g., the peak position in the susceptibility graphed in Fig. 18. The Binder ratio is the preferred way to extract T_c in many systems.

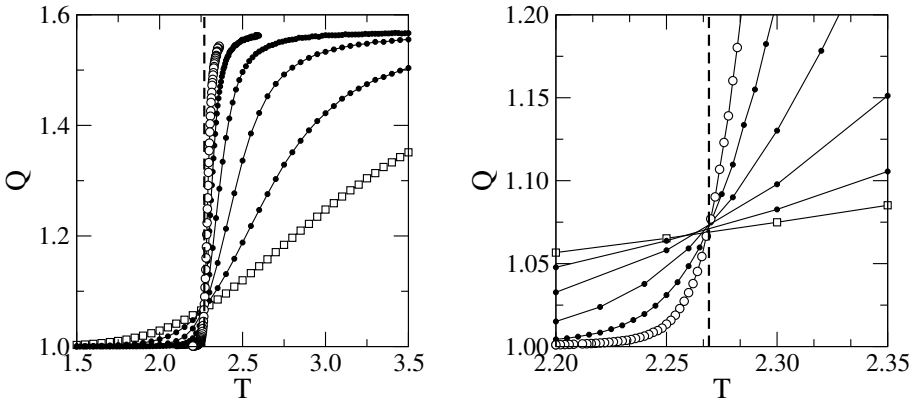

Figure 20: Binder ratio for the 2D Ising model with $L = 4$ (open squares), 8, 16, 32, 64, and 128 (open circles) graphed using two different scales. The dashed lines indicate T_c.