Christine Gabriel

Dynamical properties of the Worm Algorithm

DIPLOMARBEIT

zur Erlangung des akademischen Grades
Diplom-Ingenieurin der Technischen Physik
eingereicht an der

Technischen Universität Graz

Begutachter:

Ao.Univ.-Prof.Dr. Hans Gerd Evertz
Institut für Theoretische Physik
Technische Universität Graz

Herbst 2002

Contents

Chapter 1

The Ising Model

1.1 Description of the model

Equilibrium statistical mechanics is based upon the idea of a partition function which contains all the essential information about the system under consideration. The general form of the partition function for a classical system is

$$Z = \sum_{\text{all states}} e^{-\beta \mathcal{H}} \tag{1.1}$$

where $\mathcal{H}$ is the Hamiltonian of the system, $\beta = 1/(k_B T)$, T is the temperature, and k_B is the Boltzmann constant.

The Ising model[1] describes an interacting system of classical spins. A spin variable s_i is placed on each lattice site and allowed to take the values ± 1 (see figure 1.1). The spins interact according to a Hamiltonian

$$\mathcal{H} = -J \sum_{\langle ij \rangle} s_i s_j - H \sum_i s_i \tag{1.2}$$

The first term in eq. 1.2 is responsible for the cooperative behavior and the possibility of a phase transition. J is the exchange energy: positive J favors parallel, negative J anti-parallel alignment of the spins (which means, the first case describes a ferromagnet and the second case an antiferromagnet). We use $\langle ij \rangle$ to denote a sum over nearest neighbor spins. If $J = 0$, eq. 1.2 is the Hamiltonian of a paramagnet. In this case the only influence ordering the spins is the field H. The spins do not interact, there are no cooperative effects and hence no phase transition.

The Ising model has been solved for one [1] and (with much more effort) two dimensions [2]. The two-dimensional Ising model in a magnetic field and the three-

[1] We follow reference [1]

dimensional model, even in zero field, remain unsolved, although their properties are known very precisely from numerical work.

Despite its simplicity the Ising model is widely applicable because it describes the critical properties of many interacting two-state systems.

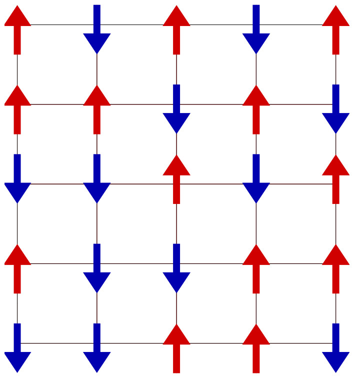

Figure 1.1: Schematical depiction of the Ising Model

1.2 Some applications of the model

1.2.1 Order-disorder transition in binary alloys

An example of a binary alloy is beta-brass. It consists of an equal number of copper and zinc atoms located on the sites of a bcc lattice. At high temperatures a lattice site is occupied randomly by either a copper or a zinc atom. As the temperature is lowered, there is, at $T_c = 733K$, a continuous phase transition to an ordered state where each atomic species preferentially occupies one of the two sublattices of the bcc lattice. In order to describe the interactions in beta-brass, we assign the variables

$$s_i = 1 \text{ if the site } i \text{ is occupied by a copper atom}$$
$$s_i = -1 \text{ if the site } i \text{ is occupied by a zinc atom.}$$

Defining J_{CuCu}, J_{ZnZn} and J_{CuZn} as the interaction between two copper, two zinc and a copper and a zinc atom respectively, a simple ansatz for the energy of beta-brass is:

$$
\begin{aligned}
\mathcal{H} = &-\frac{1}{4}\sum_{\langle ij\rangle} J_{CuCu}(1+s_i)(1+s_j) + \frac{1}{4}\sum_{\langle ij\rangle} J_{ZnZn}(1-s_i)(1-s_j) \\
&+\frac{1}{4}\sum_{\langle ij\rangle} J_{CuZn}\{(1+s_i)(1-s_j)+(1-s_i)(1+s_j)\} \\
&= -J\sum_{\langle ij\rangle} s_i s_j - H\sum_i s_i + C \qquad (1.3)
\end{aligned}
$$

where $J = \frac{1}{4}(J_{CuCu} + J_{ZnZn} - 2J_{CuZn})$, $H = \frac{1}{2}(J_{CuCu} - J_{ZnZn})$, C is a spin-independent term, and, because there are equal numbers of copper and zinc atoms, $\sum_i s_i = 0$. Thus this description is eqivalent to an Ising model.

As long as there are no impurities or vacancies, the use of an Ising variable is not an approximation (as each lattice site is strictly in one of two states: occupied by copper or occupied by zinc). Moreover in this particular example, further-neighbor interactions, multi-spin terms (such as $s_i s_j s_k s_l$) and long-range interactions turn out to be unimportant. For beta-brass the most significant correction to the Ising model comes from the variation of the exchange interaction J with temperature which results from the thermal expansion of the lattice. Allowing for this, the agreement of the simple model with experiment is excellent [1].

1.2.2 Lattice gas models

The Ising model turns out to be mathematically equivalent to the following highly stylized model of a non ideal gas. We divide the d-dimensional space occupied by the gas up into cells of a lattice, where a lattice cell can either be occupied by an atom or vacant. A variable $n_i = 1, 0$ is used to represent an occupied or unoccupied site respectively (see figure 1.2). As n_i is a two-state variable, it is possible to map it on to an Ising spin, $s_i = \pm 1$. This is achieved by the transformation

$$n_i = (1 - s_i)/2. \qquad (1.4)$$

Substituting 1.4 in 1.2, one gets a Hamiltonian of the form

$$\mathcal{H} = -J\sum_{\langle ij\rangle} n_i n_j - \mu\sum_i n_i + C \qquad (1.5)$$

where J is a nearest neighbor interaction which favors neighbouring sites being occupied, μ is a chemical potential and C a spin-independent constant. It controls the number of atoms: a large positive μ will lead to most sites being occupied whereas a large negative μ will favor vacancies.

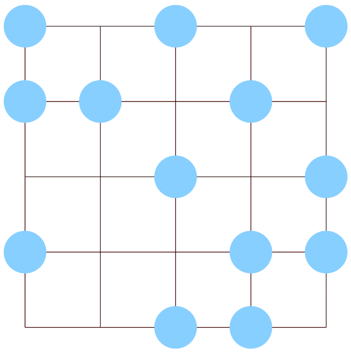

Figure 1.2: Schematical depiction of the Lattice Gas Modell

A system which is well modeled by a lattice gas, is hydrogen adsorbed on the (110) surface of iron. The potential wells between the iron atoms form a triangular lattice, whose sites can either be occupied or vacant. The coverage, respectively the number of occupied sites, is determined by the pressure of the hydrogen gas in contact with the surface.

1.3 The Fortuin-Kasteleyn representation

In the Ising model, the contribution of a nearest neighbor pair of spins i and j to the weight of a configuration, multiplied by the constant factor $e^{-\beta J}$, can be written as

$$e^{-\beta J} e^{\beta J s_i s_j} = \delta_{s_i s_j} p + (1-p), \text{ where } p = 1 - e^{-2\beta J} \tag{1.6}$$

We now introduce a new set of variables f_{ij} which can take the values 0 or 1 and rewrite the right hand side of 1.6:

$$p + (1-p)\delta_{s_i s_j} = \sum_{f_{ij}=0,1} \delta_{f_{ij},1} \delta_{s_i s_j} p + \delta_{f_{ij},0}(1-p) \tag{1.7}$$

Substituting 1.6 and further 1.7 into the partition function of the Ising model

$$Z_{Ising} = \sum_{\{s_i\}} e^{\beta J \sum_{\langle i,j \rangle} s_i s_j} \tag{1.8}$$

leads to the *Edwards-Sokal-Fortuin-Kasteleyn representation* [3].

$$Z = \sum_{\{f\}} \sum_{\{s\}} \prod_{\langle i,j \rangle} e^{\beta J} [\delta_{f_{ij},1} \delta_{s_i s_j} p + \delta_{f_{ij},0} (1-p)] \tag{1.9}$$

Summing over spins gives the *Fortuin-Kasteleyn random bond representation*[3]

$$Z = 2^{\#clusters} e^{-\beta J N_{nn}} \sum_{\{f\}} (e^{2\beta} - 1)^{n_f}, \tag{1.10}$$

where N_{nn} stands for the number of nearest neighbor pairs,$\{f\}$ denotes the bond configurations, n_f is the number of bond variables $f_{ij} = 1$ and $\#$ clusters is the number of clusters formed by bonds $f_{ij} = 1$.

1.4 The High-temperature representation

Using 1.2 without outer field H we can write

$$Z_{Ising} = \sum_{\{s_i\}} \prod_{\langle i,j \rangle} e^{\beta J s_i s_j} \tag{1.11}$$

Since $s_i s_j$ can only take two different values, $s_i s_j = \pm 1$ and since

$$e^{\pm A} = \cosh A \pm \sinh A = \cosh A (1 \pm \tanh A) \tag{1.12}$$

we can write

$$\begin{aligned} Z_{Ising} &= \sum_{\{s_i\}} \prod_{\langle i,j \rangle} \cosh(\beta J) [1 + s_i s_j \tanh(\beta J)] \\ &= (\cosh \beta J)^{N_{nn}} \sum_{\{s_i\}} \prod_{\langle i,j \rangle} [1 + s_i s_j v] \\ &= (\cosh \beta J)^{N_{nn}} \sum_{\{s_i\}} \left(1 + v \sum_{\langle i,j \rangle} s_i s_j + v^2 \sum_{\langle i,j \rangle \neq \langle k,l \rangle} s_i s_j s_k s_l + ... \right) \end{aligned} \tag{1.13}$$

where $v = \tanh(\beta J)$ and N_{nn} denotes the number of nearest neighbor pairs on the lattice.

Each product of a pair of spins, $s_i s_j$, can be associated with a bond on the lattice which connects those two spins. Each factor of v corresponds to a single bond. Terms of order v^2 correspond to two bonds which may, or may not touch and so on. Therefore each term of oder v^n is in correspondence with a graph with n edges on the square lattice. We do not have to consider *all* the possible bond configurations when evaluating the partition function. This becomes clear when we observe a single spin-product and remember that $s_i = \pm 1$ only :

$$\sum_{\{s_i\}} (s_i^{n_i} s_j^{n_j} s_k^{n_k} s_l^{n_l}, ...) = \begin{cases} 2^N, & \text{all } n_i \text{ even} \\ 0, & \text{otherwise} \end{cases} \tag{1.14}$$

Hence the only terms in 1.13 which contribute to Z are those, in whose graphical representation each spin is touched by an even number (including zero) of bonds. It is useful to introduce a new bond variable b_{ij} with value 1 ("present") when such a bond is present between sites i and j, and value 0 ("vacant") otherwise. Only configurations of bond variables in which all b_{ij} form closed loops will contribute to Z. We will denote such configurations as *allowed configurations.*

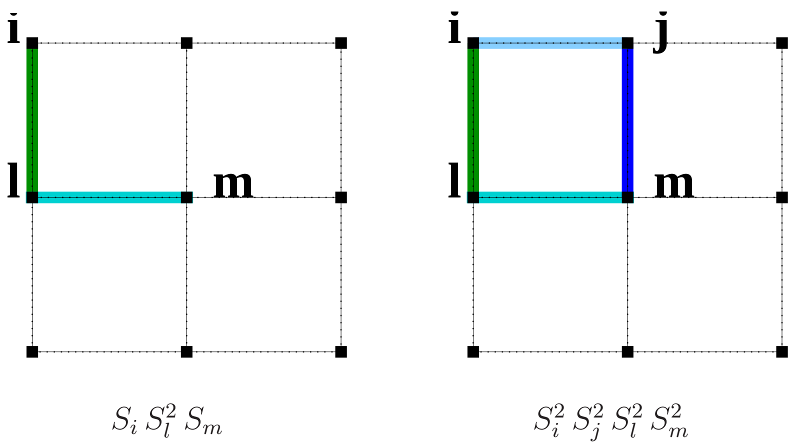

Figure 1.3: Graphs on the square lattice, each of which correspond to a product of spins in the sum of eq.1.13.

We now can rewrite the partition function Z_{Ising} as a sum over all allowed bond-configurations instead of a sum over all possible spin-configurations:

$$Z = (\cosh \beta J)^{N_{nn}} 2^N \sum_{\{b, \partial b = 0\}} v^{n_b} \tag{1.15}$$

Here $\sum_{\{b,\partial b=0\}}$ denotes the sum over all allowed bond-configurations on the lattice and n_b the number of bonds for which $b_{ij} = 1$.

The identity (1.15) is valid for the Ising model. More generally, one can write a High Temperature Representation for any model on a finite lattice by expanding $e^{\beta \mathcal{H}}$:

$$Z = \sum_{\{s_i\}} e^{-\beta\mathcal{H}(\mathbf{s})} = \sum_{\{s_i\}} \sum_{n=0}^{\infty} \frac{[-\beta\mathcal{H}(\mathbf{s})]^n}{n!}$$
$$= \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \sum_{\{s_i\}} [-\beta\mathcal{H}(\mathbf{s})]^n \tag{1.16}$$

The problem is to evaluate the trace of powers of the Hamiltonian. In many cases these are just traces of products of variables which it is helpful to identify with graphs on the lattice as before. However, in general, multiple bonds are allowed and the weights depend on the topology of the graphs.

If an exact evaluation is not possible, one can approximate the partition function by adding only a finite number of terms in the sum of either eq.1.15 or eq. 1.16. This yields to the high temperature *expansion*.

1.5 Joint representations of the Ising model

H.G. Evertz, H.M. Erkinger and W. von der Linden [3] introduced new joint representations that connect spins and Fourtuin-Kasteleyn bonds with the bonds of the exact "High-Temperature"-representation of the Ising model (see chapter 1.4). We start with the algebraic identity,

$$1 + Ov = \sum_{b=0,1} \delta_{b,0} + \delta_{b,1} O v$$
$$= \sum_{f=0,1} \sum_{b=0,1} [\delta_{f,1}(\delta_{b,0} + \delta_{b,1} O) v + \delta_{f,0}\delta_{b,0}(1 - v)] \tag{1.17}$$

with auxiliary variables b and f. By applying 1.17 with $O = s_i s_j$ and $v = \tanh \beta J$ to the High Temperature representation

$$Z = (\cosh \beta J)^{N_{nn}} \sum_{\{s_i\}} \prod_{\langle i,j \rangle} [1 + s_i s_j v] \tag{1.18}$$

we obtain a new representation for the partition function of the Ising model

$$Z = \sum_{\{s_i\}} \sum_{\{f_{ij}\}} \sum_{\{b_{ij}\}} \prod_{\langle i,j \rangle} (\cosh \beta J)[\delta_{f_{ij},1}(\delta_{b_{ij},0} + \delta_{b_{ij},1} s_i s_j) v$$
$$+ \delta_{f_{ij},0}\delta_{b_{ij},0}(1 - v)] \tag{1.19}$$

We obtain another new representation by summing over spins s_i in 1.19. This leads again to the constraint $\partial b_{ij} = 0$, which means that the b_{ij} have to form closed loops

$$Z = 2^N \sum_{\{f_{ij}\}} \sum_{\substack{\{b_{ij}\} \\ \partial b_{ij}=0}} \prod_{\langle i,j \rangle} (\cosh \beta J)[\delta_{f_{ij},1} v + \delta_{f_{ij},0} \delta_{b_{ij},0} (1-v)] \tag{1.20}$$

It turns out that eq. 1.20 gives a joint representation in terms of *Fortuin-Kasteleyn bonds* f_{ij}, which correspondent to the Swendsen-Wang bonds (see section 3.7.1) and *High-Temperature bonds* b_{ij}. This will be the basis for a new cluster method, introduced in chapter 3.7.5.

A similar representation can be formulated for Potts models [3].

Chapter 2

Phase Transitions

It is possible to find a direct connection between the partition function 1.1 and thermodynamic quantities. The free energy of a system can be determined from the partition function by

$$F = -k_B T \ln Z \tag{2.1}$$

and all other thermodynamic quantities can be calculated by appropriate derivatives of eq. 2.1. This provides the connection between statistical mechanics and thermodynamics. We consider a magnetic system: Z=$Z(T, H)$. The internal energy of a system can be obtained from the free energy, respectively from the partition function, via

$$U = \langle \mathcal{H} \rangle = -T^2 \frac{\partial (F/T)}{\partial T} = -\frac{\partial (\ln Z)}{\partial \beta} \tag{2.2}$$

where $\beta = 1/k_B T$ and the thermal average $\langle ... \rangle$ is discussed in chapter 3 . Another important quantity, the entropy, measures the amount of disorder in the system. The entropy can be determined from the free energy via

$$S = -(\frac{\partial F}{\partial T})_H \tag{2.3}$$

and the magnetization can be obtained by

$$M = -(\frac{\partial F}{\partial H})_T \tag{2.4}$$

A further derivative leads to the specific heat

$$C_X = T(\frac{\partial S}{\partial T})_X \tag{2.5}$$

with $X = H, M$ and to the isothermal susceptibility

$$\chi_T = (\frac{\partial M}{\partial H})_T \tag{2.6}$$

A phase transition occurs when there is a singularity in the free energy or one of its derivatives such as susceptibility or specific heat [1] . For a first-order phase transition the discontinuity occurs in a first derivative of the free energy. If the first derivatives are continuous but some higher derivatives are discontinuous the transition will be described as "higher order", or with the same meaning, "second order".

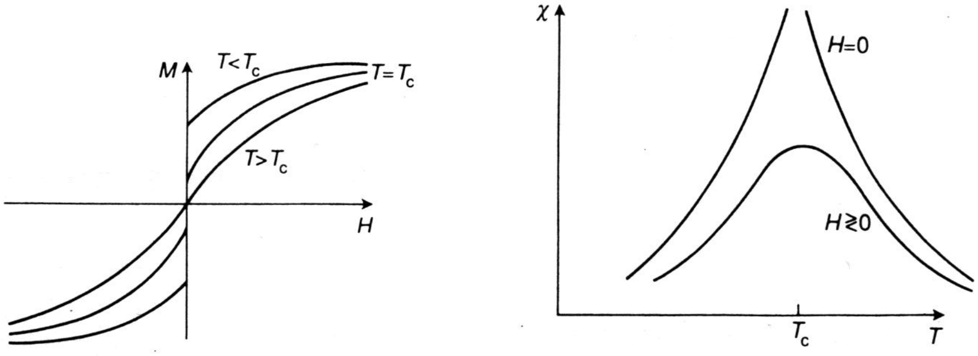

Figure 2.1: Phase Transitions of the Ising Model: The left picture shows the magnetization M as a function of an external magnetic field H and the temperature T. At the critical temperature T_c and zero magnetic field $H = 0$ the magnetizations shows an infinite slope. The right picture shows the susceptibility in dependence of T and H. At the critical temperature and zero magnetic field the susceptibility diverges.

At a phase transition a sharp change in the properties of a substance is often visible. The transition from liquid to gas, from a normal conductor to a superconductor, or from paramagnet to ferromagnet are common examples [1].

[1]This only holds in the thermodynamic limit $N \to \infty$. If the system is finite, the singularities are "rounded" (see section 3.6).

2.1 Order parameter

The distinguishing feature of most phase transitions is the appearance of a non-zero value of an 'order parameter', i.e. there is some property of the system whose thermal average is non-zero in the ordered phase but identically zero in the disordered phase[1]. The order parameter is different in various kinds of physical systems. It may be a scalar quantity or may be a multicomponent (or even complex) quantity [4].

In a ferromagnetic/paramagnetic transition, the magnetization M changes from a non-zero value at $T < T_c$ to zero at T_c. Thus the magnetization is a suitable order parameter for the ferromagnetic/paramagnetic transition.

In a ferromagnetic material it is energetically favorable for all spins to align. In an antiferromagnetic material, by contrast, the energetically favored configuration is one in which neighboring spins are anti-aligned. Even at low temperatures an antiferromagnet has no net magnetic moment. We can define a suitable order parameter on a bipartite lattice by dividing the lattice sites into two classes A and B such that in the lowest-energy configuration the spins on all sites in class A are parallel to another and anti-parallel to all spins on sites in class B. We then define the order parameter to be

$$m = \sum_{i \in \mathrm{A}} s_i - \sum_{i \in \mathrm{B}} s_i \tag{2.7}$$

Thus defined, the order parameter at $T = 0$ is everywhere equal to the spins on sites in class A, and has vanishing thermal average for $T > T_c$.

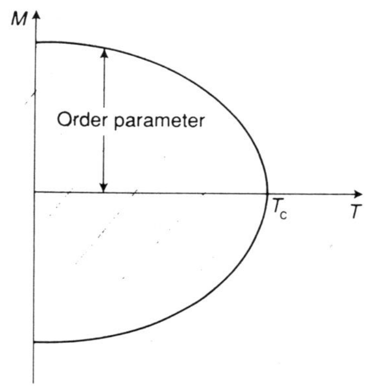

Figure 2.2: Paramagnetic/ferromagnetic phasetransition of the Ising model: While above the critical temperature T_c the spontanous magnetization M vanishes, it shows a finite value below T_c.

2.2 Correlation Function

Even if a system is not ordered, there will be microscopic regions in the material in which the characteristics of the material are correlated. For a fuller understanding of phase transitions, it is necessary to consider what is happening on a microscopic level [1]. The spin-spin correlation function, for example, which measures the correlation between the spins on sites i and j, is

$$\Gamma_{ij} = \langle (s_i - \langle s_j \rangle)(s_i - \langle s_j \rangle) \rangle \tag{2.8}$$

where $\mathbf{r_i}$ is the position vector of site i and $\langle ... \rangle$ denotes a thermal average (see eq. 3.2). If the system is translationally invariant, then $\langle s_i \rangle = \langle s_j \rangle$ and Γ depends only on $(\mathbf{r_i} - \mathbf{r_j})$

$$\Gamma(\mathbf{r_i} - \mathbf{r_j}) = \langle s_i s_j \rangle - \langle s_j \rangle^2 \tag{2.9}$$

Away from the critical point, the spins become uncorrelated as $r \to \infty$ and the correlation function decays to zero exponentially with the distance between the spins

$$\Gamma(\mathbf{r}) \sim r^{-\rho} \mathrm{exp}^{-r/\xi} \tag{2.10}$$

where ρ is some number and ξ the correlation length. The correlation length is the characteristic length scale of the system. According to eq. 2.10, the spins fluctuate in blocks of all sizes up to size ξ, but fluctuations that are significantly larger are exceedingly rare. At the critical point itself, long range order develops. The correlation length reaches infinity and the correlation function decays as a power law [1]

$$\Gamma(\mathbf{r}) \sim \frac{1}{r^{d-2+\eta}} \tag{2.11}$$

where η is a so called critical exponent (see section2.3) and d is the dimensionality of the system. We mention that it is possible to relate the spin-spin correlation function to the fluctuations in the magnetization and hence to the susceptibility. By writing the magnetization as a sum over spins, one obtains [1]:

$$\langle (M - \langle M \rangle)^2 \rangle = \frac{\chi}{\beta} = \langle \sum_i (s_i - \langle s_i \rangle) \sum_j (s_j - \langle s_j \rangle) \rangle = \langle \sum_{ij} \Gamma_{ij} \rangle \tag{2.12}$$

2.3 Critical point exponents and universality

We have argued that the critical point is marked by divergences in the specific heat and susceptibility. It turns out to be very important to understand more carefully the form of these divergences and the singular behavior of the other thermodynamic functions near the critical point [1]. For this purpose we define the reduced distance from the transition

$$t = (T - T_c)/Tc \tag{2.13}$$

Then the critical exponent associated with a function $F(t)$ is (assuming that this limit exists)

$$\lambda = \lim_{t \to 0} \frac{\ln \mid F(t) \mid}{\ln \mid t \mid} \tag{2.14}$$

or, as it is more usually written,

$$F(t) \sim\mid t \mid^{\lambda} \tag{2.15}$$

It is important to remember that 2.15 only represents the asymptotic behavior of the function $F(t)$ as $t \to 0$.

The zero-field susceptibility and the zero-field specific heat diverge at T_c as

$$\chi_T \sim\mid t \mid^{-\gamma} \tag{2.16}$$
$$C_H \sim\mid t \mid^{-\alpha} \tag{2.17}$$

The definitions of critical exponents are collected in Table 1.1 [1]. In particular, ν describes how the correlation length diverges as the critical temperature is reached.

It turns out that, whereas T_c depends sensitively on the details of the interatomic interactions, the critical exponents are to a large degree universal, depending only on a few fundamental parameters, such as the dimensionality of space, and the symmetry of the order parameter (for models with short range interactions). By making sure that one is working in the right dimension and that the symmetry of the order parameter is correctly represented by a model, such a model can be used to obtain critical exponents for all the systems within its universality class. This immediately illustrates the power of using simple models like the Ising model to describe critical behavior [1].

Striking evidence for this comes from a plot by Guggenheim shown in fig. 2.3, where the coexistence curve of eight different fluids are plotted in reduced units, T/T_c and ρ/ρ_c. Close to the critical point (and even far away from it!) all the data lie on the same curve and hence can be described by the same exponent β which corresponds to that of the Ising model.

Zero-field specific heat	$C_H \sim\mid t \mid^{-\alpha}$
Zero-field magnetization	$M \sim (-t)^{\beta}$
Zero-field isothermal susceptibility	$\chi_T \sim\mid t \mid^{-\gamma}$
Critical isotherm (t=0)	$H \sim\mid M \mid^{\delta} \mathrm{sgn}(M)$
Correlation length	$\xi \sim\mid t \mid^{-\nu}$
Pair correlation function at T_c	$G(\mathbf{r}) \sim \frac{1}{r^{d-2+\eta}}$

Table 2.1: Definitions of the critical exponents for a magnetic system

Universality class	α	β	γ	δ	ν	η
2-d Ising	0	1/8	7/4	15	1	1/4
3-d Ising	0.10	0.33	1.24	4.8	0.63	0.04

Table 2.2: Critical exponents of the two- and three dimensional Ising model. The exponents in $2d$ are exact.

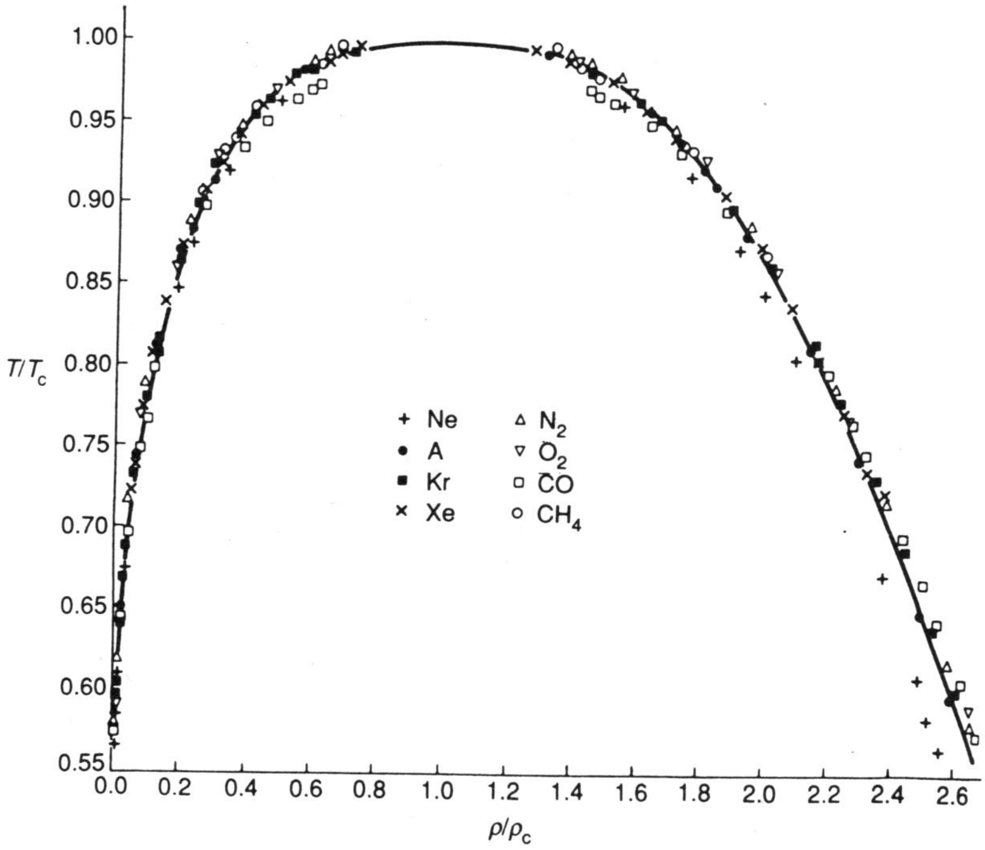

Figure 2.3: The coexistence curve of eight different fluids plotted in reduced variables. The fit assumes the critical exponent of the three-dimensional Ising-model $\beta = 1/3$. After [5].

Chapter 3

Monte Carlo methods

3.1 Evaluating thermal averages

In chapter 1.1 we introduced the partition function. The sum in eq. 1.1 is over all possible states of the system and thus depends upon the size of the system and the number of degrees of freedom for each particle. For systems consisting of only a few interacting particles the partition function can be written down exactly with the consequence that the properties of the system can be calculated in closed form. In a few other cases the interactions between particles are so simple that evaluating the partition function is possible, but in general the partition function cannot be evaluated exactly. Even enumerating the terms in the partition function on a computer can be a daunting task. If we have only 10 000 interacting particles, a very small fraction of Avogadro's number, with only two possible states per particle, the partition function would contain $2^{10000} \approx 10^{3000}$ terms [6].

One of the key questions in statistical mechanics is how to evaluate the thermal average of some quantity Q which depends on the system configuration $\mathbf{x}$. $\mathbf{x}$, in general, is a highly dimensional vector in phase space and stands for the considered degrees of freedom, for example spins in a system consisting of N particles $\mathbf{x} = (s_1, s_2, s_3, ..., s_N)$. The probability of any particular state of the system is determined by the Boltzmann factor

$$P(\mathbf{x}) = e^{-\beta\mathcal{H}(\mathbf{x})}/Z \tag{3.1}$$

where $\mathcal{H}(\mathbf{x})$ is the Hamiltonian when the system is in the state $\mathbf{x}$. The thermal average of an observable $Q(\mathbf{x})$ is

$$\langle Q \rangle = \frac{1}{Z} \sum_{\{\mathbf{x}\}} e^{-\beta\mathcal{H}(\{\mathbf{x}\})} Q(\mathbf{x}) \tag{3.2}$$

where Z is the partition function 1.1 and $\{\mathbf{x}\}$ labels the different configurations of the system. The problem of the evaluation of 3.2 lies, as mentioned before, in the extremely high dimensionality of the sum.

We now use the *central limit theorem*, which says that if random variables $x_1, x_2, x_3, ..., x_m$ are all *independent* of each other and drawn from the same distribution, the arithmetic average $\mathcal{Q} = \frac{1}{M}\sum_{m=1}^{M} Q(x_m)$ in the limit of $M \to \infty$ will always be distributed according to a Gaussian distribution, irrespective of the distribution from which the x_m were drawn. One can also show that the variance of $\mathcal{Q}$ equals the variance of Q divided by M

$$\text{var } \mathcal{Q} = \frac{\text{var } Q}{M} \tag{3.3}$$

Thus high dimensional sums of the type

$$\langle Q \rangle = \sum_{\{\mathbf{x}\}} Q(\mathbf{x}) P(\mathbf{x}) \tag{3.4}$$

can be approximated by the average of a sample $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_M\}$ of random variables distributed according to $P_{\mathbf{x}}$ because

$$\langle Q(\mathbf{x}) \rangle = \frac{1}{M} \sum_{m=1}^{M} Q(\mathbf{x}_m) \pm \sqrt{\frac{\text{var } (Q)}{M}} \tag{3.5}$$

holds. For more details see reference [7].

To summarize: All we have to do is to produce random variables $\mathbf{x}_\mu$ according to a given probability $P(\mathbf{x}_\mu)$, calculate the desired quantity $Q(\mathbf{x}_\mu)$ according to the given configurations $\mathbf{x}_\mu$. Finally we have to average over all observations $Q(\mathbf{x}_\mu)$ we have obtained during our sampling.

3.2 Markov Chain Monte Carlo

A way to produce random variables according to a given distribution P is the method of sampling Markov-Chains. Consider a situation where the probability of an event changes from trial to trial. If the probability of an event depends only upon the outcome of the preceding trial, the sequence of events is called a *Markov-Chain*. Denote a set of N mutually exclusive discrete states by $\{Z_1, ..., Z_N\}$. Suppose we want to calculate the probability of the event $L+1$ according to some law

$$P(x_{L+1} = Z_j \mid x_L, x_{L-1}, ..., x_1) \tag{3.6}$$

Then the sequence is called a Markov Chain if the above probability actually depends only on x_L, e.g.

$$P(x_{L+1} = Z_j \mid x_L = Z_i) =: P_{ij}^{(L)} \tag{3.7}$$

The quantities $P_{ij}^{(L)}$ are called the *stochastic matrix* of the Markov Chain. The parameter L labeling the events Z can be viewed as a discrete *time*. If the stochastic matrix does not depend on time, the Markov Chain is called *homogeneous*. This is assumed to be true in the sequel. The stochastic matrix P_{ij}, inducing transitions among different states Z, satisfies the following relations

$$P_{ij} \geq 0 \text{ and } \sum_{J=1}^{N} P_{ij} = 1 \tag{3.8}$$

It is important whether all states Z can be reached from a given initial state, e.g. whether all states have a nonzero probability to appear in the chain, respectively if the Markov Chain is *ergodic*. A Markov Chain is called ergodic if

$$\text{for all } i, j\ \exists M > 0 : (P_{ij}^{M}) > 0 \tag{3.9}$$

This is assumed to be true in the sequel.

We now consider an ensemble of many Markov chains, all with the same transition matrix P_{ij}. At a given time n, lat the states in this ensemble be distributed according to some distribution $\rho^n(Z_i)$. I.e. the probability that a system in the considered ensemble is in state Z_i at time n is

$$P(x_n = Z_i) =: \rho^n(Z_i) \tag{3.10}$$

We obtain the distribution at time $n+1$ by applying the stochastic matrix to the distribution at time n (for further details see [8]). Thus the evolution equation is given by

$$\rho^{n+1}(Z_i) = \sum_{j=1}^{N} P_{ij}\rho^n(Z_j) \tag{3.11}$$

A distribution is called *invariant distribution* $\rho^\infty(Z_i)$ if it does not change by application of the stochastic matrix

$$P_{ij}\rho^\infty = \rho^\infty \tag{3.12}$$

It can be proved [8] that by multiple application of the stochastic matrix to a distribution, this distribution converges to the (unique) invariant distribution although this convergence can be very slow.

How shall we choose the stochastic matrix P_{ij} to create a given distribution $P(\mathbf{x})$ as the invariant distribution of the Markov Chain? A sufficient condition is that the the transition probabilities satisfy the micro-reversibility or *detailed balance condition*

$$P(Z_i)P_{ij} = P(Z_j)P_{ji} \tag{3.13}$$

Provided this condition is satisfied (and the stochastic matrix P is ergodic), one can prove that applying the stochastic matrix leads to a invariant distribution equal to the given distribution $P(\mathbf{x})$ [8]. Furthermore, the the distribution of states within an *individual* Markov chain will, starting from an arbitrary initial state, also converge to $P(x)$, so that they can be used to calculate appropriate averages (3.5). See, however, chapter 3.5 for the intricacies of error analysis.

3.3 The Metropolis-Hastings algorithm

The most important and most frequently used algorithm for the Markov process is the Metropolis-Hastings [4] algorithm.

1. Start with an arbitrary start configuration.

2. First one creates a trial state $\mathbf{x}^T$ for time $n+1$ from the actual state $\mathbf{x}$ according to a proposal distribution $q(\mathbf{x}^T \mid \mathbf{x})$.

3. This trial state $\mathbf{x}^T$ is accepted with the probability

$$\alpha(\mathbf{x}^T, \mathbf{x}) = \min\left(1, \frac{P(\mathbf{x}^T)q(\mathbf{x} \mid \mathbf{x}^T)}{P(\mathbf{x})q(\mathbf{x}^T \mid \mathbf{x})}\right) \tag{3.14}$$

   The second factor in the numerator and in the denominator is due to Hastings. He improved the original idea of metropolis. This correction assures detailed balance [8].

   If the trial state is not accepted, the old state $\mathbf{x}$ is taken again.

4. Go to step 2.

3.4 The Gibbs Method

The Gibbs sampler, also known as the *heat bath method*, is a special case of Metropolis Hasting sampling, wherein trial states are chosen such that the acceptance probability $\alpha \equiv 1$.

Gibbs sampling is illustrated for a case with variables $(x_1, .., x_N) = \mathbf{x}$

1. Start with an arbitrary configuration
2. On each iteration, we start from the current state $\mathbf{x}$. Only one of the components of $\mathbf{x}$ will be changed, say x_i. The index i of this component is chosen at random. The possible changes of x_i are denoted with $\{x'_i\}$. The proposal probability for a new configuration $\mathbf{x}^T$, produced by a change of $x_i \rightarrow x_i^* \in \{x'_i\}$, is given by

$$q(\mathbf{x}^T) = \frac{P(\mathbf{x}^T)}{\sum_{\{x'_i\}} P(x'_i)} \tag{3.15}$$

3. Substituting 3.15 into eq. 3.14 leads to an acceptance probability $\alpha = 1$

Gibbs sampling in general leads to somewhat faster convergence of the Markov chain (see below). However, it can be difficult to generate trial configurations which are distributed to $q(\mathbf{x}^T)$.

3.5 Independence of configurations

It is clear that with any of our evolution algorithms, the probability of finding a particular state at step $(t + 1)$ is strongly dependent on the state of the system at the t^{th} step. Since our configurations $\mathbf{x}_i$ are thus *not* all statistically independent, the error estimation in eq. 3.5 is not valid in that form.

The correlation of configurations is illustrated in figure 3.1 which shows the time series of energy sampled during a Monte Carlo run. Since we start at an arbitrary configuration the system needs time to *thermalize*, i.e. to tend toward the equilibrium distribution corresponding to the given temperature. This is due to the fact that the configurations produced during the Markov Chain Monte Carlo are correlated.

We also see in fig. 3.1 that even after the system is thermalized, we have to wait some number of steps for the system to lose its "memory" of the previous value of the quantity, in order to obtain statistically independent measurements.

3.5.1 The autocorrelation function

One way of deciding when it is time to evaluate a quantity Q again is to calculate the autocorrelation function of Q along our Markov chain. The corresponding normalized

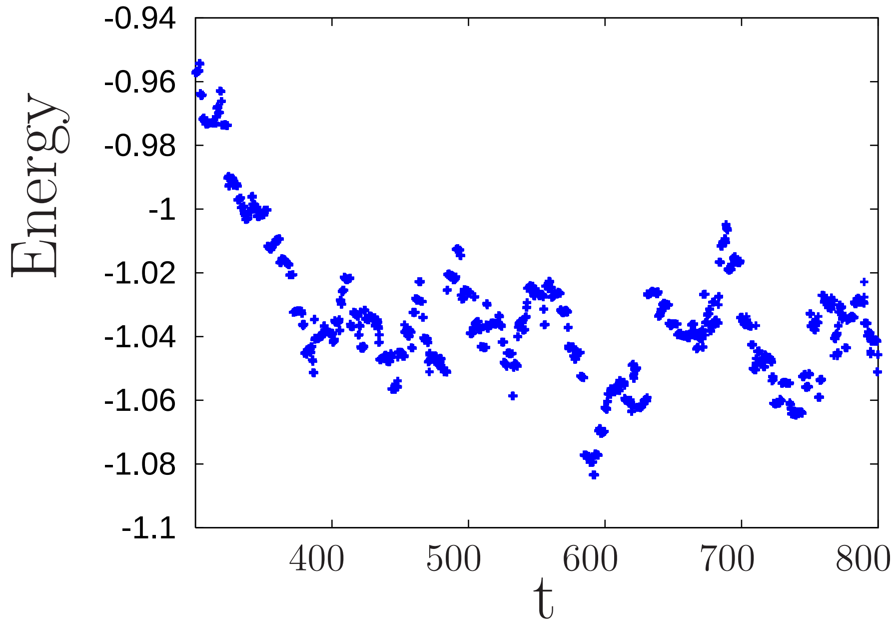

Figure 3.1: Illustration of the thermalization and autocorrelation effect. E denotes the Energy and t the CPU-time in units of sweeps. The data is taken from a Monte Carlo run of the Worm Algorithm ($112 \times 112 \times 28$ system, $H = 0.00348, T = 4.4665$, see chapter 9).

autocorrelation function C_Q is defined as [6].

$$C_Q(t) = \frac{\langle Q_i Q_{i+t} \rangle - \langle Q_i \rangle^2}{\langle Q_i^2 \rangle - \langle Q_i \rangle^2} \tag{3.16}$$

where Q_i is the value of Q at the i^{th} Monte Carlo step and the averages are taken over the Markov chain. C has been normalized so that it is unity at $t = 0$, where the values of Q are completely correlated, and it decays to zero when t is large enough to have become completely uncorrelated. One can often associate a time scale τ, the *correlation time* with this decay. The decay of the correlation function is usually a sum of exponentials

$$C_Q(t) \sim \sum_i c_i(Q) e^{-\frac{t}{\tau_i^{(Q)}}} \tag{3.17}$$

This behavior is shown in figure 3.2. We define the *exponential autocorrelation time* for the observable Q by the asymptotic decay [9]

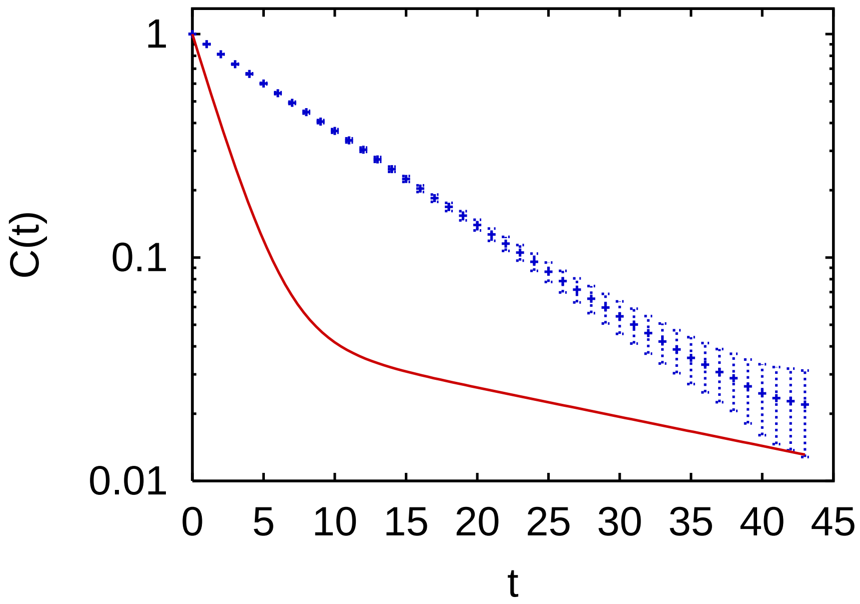

Figure 3.2: Decay of the autocorrelation function $C(t)$. The upper data points show a single-mode decay of an autocorrelation function $C(t) \sim c_1 \cdot e^{-t/\tau_1}$ while the lower curve shows the qualitative behavior of an autocorrelation function decaying with two modes: $C(t) \sim c_1 \cdot e^{-t/\tau_1} + c_2 \cdot e^{-t/\tau_2}$. The statistical errors of the upper curve (see chapter 5) are approximately constant on a non-logarithmic scale.

$$\tau_{exp}^{(Q)} := \lim_{t\to\infty} \sup \frac{t}{-\log C_Q(t)} \tag{3.18}$$

This is the relaxation time of the the slowest mode in the Monte Carlo updates which couples to Q. One can also define the slowest overall mode as

$$\tau_{exp} := \sup_Q \left(\tau_{exp}^{(Q)}\right) \tag{3.19}$$

We have mentioned above that if the observables Q_i were statistically independent, the error estimate of $\langle Q \rangle$ would be $\sqrt{\mathrm{var}(Q)}/\sqrt{M}$, where M means the number of measurements. Instead the actual statistical error of $\langle Q \rangle$ is controlled by the *integrated autocorrelation time*

$$\tau_{int}^{(Q)} := \frac{1}{2} + \sum_{t=1}^{\infty} C(t)_Q \tag{3.20}$$

and becomes $\mathrm{var}(Q)/\sqrt{M^*}$ with $M^* = M/(2\tau_{int}^{(Q)})$. Therefore a Monte Carlo run of M

measurements effectively contains only $M/(2\tau_{int}^{(Q)})$ independent samples for measuring $\langle Q \rangle$. If $C(t)$ shows a single exponential decay $e^{-t/\tau_{exp}^{(Q)}}$ then $\tau_{exp}^{(Q)} = \tau_{int}^{(Q)}$ otherwise $\tau_{exp}^{(Q)} > \tau_{int}^{(Q)}$ [9].

3.5.2 Critical slowing down

In chapter 2.2 we mentioned that near a critical point, long range order develops. As we approach the critical temperature, the characteristic "time" (more strictly the number of steps) τ for the autocorrelation to disappear gets longer and longer. This phenomenon is known as *critical slowing down.* As the transition is approached, the correlation length ξ becomes longer and longer and the system contains larger and larger "islands" in which all the spins are aligned. Empirically, it is found that decorrelation times τ are related to the correlation length ξ by a relation of the form

$$\tau = \xi^z, \tag{3.21}$$

where z are *dynamical critical exponents.* Note that eq. 3.21 holds for τ_{int}^{Q} and τ_{exp}^{Q} but with a priori different dynamical critical exponents z_{int}^{Q} and z_{exp}^{Q}, which depend on the quantity Q also. Analogous to the statical critical exponents, z_{exp} corresponds to the exponent of a correlation length and z_{int} to the exponent of sum of correlations at different temporal distances, i.e. a susceptibility. For the Ising model, z is found to be approximately 2 for the Metropolis algorithm and for any other Markov process in which the spins are flipped one at a time [1].

This has serious consequences for our ability to perform simulations for systems in the interesting regime close to the critical point [6].

3.6 Finite-size corrections

When simulating finite systems, finite-size effects must be taken into account. Away from the critical point, where the correlation length is small compared to the system size, this is usually not a major problem. Here the errors due to the limitation of the lattice are small compared to statistical errors.

The problem becomes much more acute as a phase transition is approached, because on a finite lattice the correlation length ξ is prevented from becoming infinite. As a result any singularities associated with the phase transition are shifted and rounded. A way to learn something about the infinite system is to obtain high quality data for lattices of different linear dimension L and extrapolate to $L = \infty$.

[1] This becomes plausible, since a single spin-flip algorithm can be compared with a random walk on the lattice which shows a dynamic critical exponent $z = 2$ [4]

Moreover, by simulating a number of large systems, one can indeed extract values of the critical exponents by plotting appropriate quantities for different values of L (the linear dimension of the lattice) and fitting these plots with functions that contain the critical exponent as fit parameter. Further information on this so-called *finite size scaling* can be found e.g. in reference [4].

3.7 Cluster Algorithms

One of the most severe limitations of Monte Carlo many-body simulations stems from critical slowing down. Long autocorrelation times cause restrictions on the maximum lattice size for which decent statistics can be achieved [8].

An efficient way to overcome critical slowing down is to use cluster flipping methods instead of simply attempting single spin-flips. Even when the Markov Chain has reached a representative spin configuration, moves by a single spin flip will definitively take a long time to uncorrelate. We expect that reasonable Monte Carlo moves are rather those which flip large portions of ferromagnetic domains at once. So we

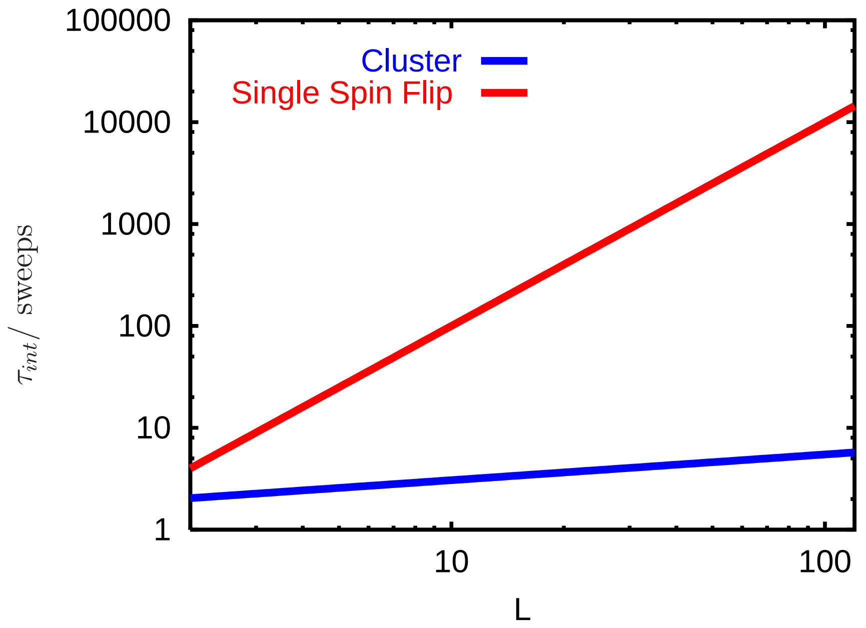

Figure 3.3: Comparison of the increase of the integrated autocorrelation time tau_{int} with the system's size L between Single Spin Flip algorithm (e.g. $z \approx 2$ for algorithms of the Metropolis type [10]) and Cluster Algorithm (Swendsen Wang: $z \approx 0.5$ [11]) in $d = 2$. For a qualitative comparison schematic curves $\tau = L^z$ are plotted.

glue ferromagnetically aligned spins together to form rigid *clusters* which have to be flipped simultaneously in the subsequent Markov step. Of course, flipping all spins of a ferromagnetic domain together would not be a valid procedure if those domains would never split apart. We need a stochastic ergodic process describing the formation of clusters in a given spin configuration $\mathbf{s}$. For example in the Swendsen-Wang algorithm this process is given by:

- Parallel neighbor spins are glued (from a bond) only with probability $p_b < 1$.
- There is still a finite chance $(1 - p_b)$ that ferromagnetic bonds brake apart.
- Unequal spins are never glued together.

Then clusters are described by a set of bonds b_{ij} between spin s_i and s_j [8].

There are two accesses to formulate cluster algorithms for the Ising model: via the "Edwards-Sokal-Fortuin-Kasteleyn representation" (see chapter 1.3) and via the "Joint representation"(see chapter1.5) of the partition function. The cluster-algorithms following from the Fortuin-Kasteleyn representation are described in the following sections 3.7.1-3.7.5.

Unfortunately the applicability of cluster algorithms is restricted to a fairly small class of models, such as Ising, Potts, O(N)-spin models, SOS. An exception is the "Geometric cluster algorithm", introduced in chapter 3.7.4, which is based on geometric transformations and provides a wider applicability.

3.7.1 The Swendsen-Wang algorithm

The first successful cluster algorithm was introduced by Swendsen and Wang [12]. The essential idea is to let the system itself decide the shape of the clusters of spins which are to be flipped simultaneously. The Swendsen-Wang cluster algorithm is implied immediately by eq.1.9 and thus follows directly from the Edwards-Sokal-Fortuin-Kasteleyn representation [13]:

Given a configuration $\{s\}$ of spins, one can generate bond variables f_{ij} with conditional probabilities following from eq. 1.9, namely $s_i = s_j$ implies that $f_{ij} = 1$ with probability p, and $s_i \neq s_j$ implies $f_{ij} = 0$, because then only the second term in eq.1.9 contributes to the partition function. Conversely, given a configuration of bond variables $\{f\}$, eq.1.9 implies that for $f_{ij} = 1$ we have $s_i = s_j$, i.e. all sites in a cluster have a common spin value, whereas $f_{ij} = 0$ does not imply any constraints on the product $s_i s_j$, i.e., the spin values of different clusters are independent of each other [3]. The Swendsen-Wang algorithm iterates these steps:

1. For each interaction pair $\langle i, j\rangle$ with equal spins $s_i = s_j$ a bond is inserted with probability $p = 1 - e^{-2\beta J}$.

2. All spins which are pairwise connected by bonds are glued together and form clusters. An isolated spin (not connected to any bond) is also called a cluster. The clusters are identified and enumerated $1, 2, ...N_c$. They constitute the objects to be manipulated by Monte Carlo moves.

3. In step three new spin values are generated for all clusters.

   The clusters are scanned sequentially and for each cluster a new spin value is chosen at random. There is a finite probability that spins in a cluster are not changed. In the Ising case the spins in a cluster are flipped with probability 1/2.

4. A measurement for the observable Q is carried out.

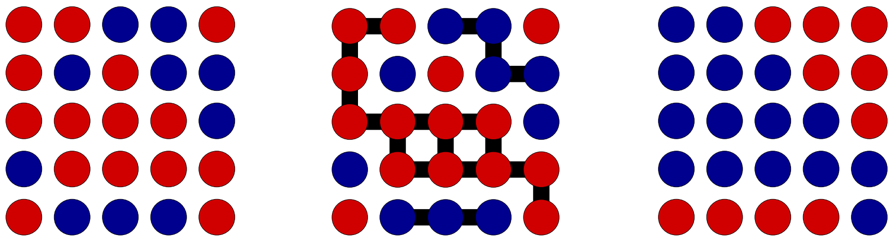

Figure 3.4: Schematic view of the Swendsen-Wang algorithm: original spin configuration → intermediate bond configuration → new spin configuration.

The key success factor of the Swendsen-Wang algorithm is that the bond creation is driven by the Hamiltonian itself. This leads to a proposal spin configuration which will *always* be accepted. Contrast this with the naive scheme where large modifications are proposed by an unphysical proposal distribution. Then the modifications have to be decided by the Metropolis-Hastings acceptance probability which leads to tiny acceptance rates.

The Swendsen-Wang algorithm shows a critical dynamical exponent $z^E_{int} = 0.25$ (or possibly $\tau^E_{int} \sim \ln L$) in two dimensions and $z^E_{int} = 0.50$ in three dimensions for the Ising model[14]. This algorithm can be formulated directly for Potts models and was generalized for O(N)- spin models.

3.7.2 Improved estimators

Instead of taking only $Q(\mathbf{s})$, *improved estimators* allow us to use averages of Q over all spin configurations which can be reached by cluster flips from a given cluster configuration. The improved estimator for Q is thus nothing but Q in bond-representation. Improved estimators of the Swendsen-Wang algorithm (and related algorithms) are introduced in the following sections.

Measurement of spatial correlation functions

In order to calculate the thermal average of spatial correlation functions $\langle s_i s_j \rangle$, given a certain bond configuration $\{f\}$ we distinguish two different cases:

- Site i and site j belong to different clusters. The average over the four possible spin values $s_i, s_j = \pm 1$ yields

$$\langle s_i s_j \rangle = \sum_{\{s\}} p(\{s\} \mid \{f\}) s_i s_j = \frac{1}{4} \sum_{s_i \pm 1} \sum_{s_j \pm 1} s_i s_j = 0 \tag{3.22}$$

- Site i and site j belong to the same cluster. Because all spins in the same cluster need to have the same spin value, $s_i = s_j$ and $s_i s_j = 1$. Now the average leads to

$$\langle s_i s_j \rangle = \sum_{\{s\}} p(\{s\} \mid \{f\}) s_i s_j = \frac{1}{2} \sum_{s_j = s_j = \pm 1} s_i s_j = 1 \tag{3.23}$$

By combining the results above we obtain the rule for measuring an improved estimator for the spatial correlation function:

$$\langle s_i s_j \rangle = \delta_{i \sim j} \tag{3.24}$$

where the index $i \sim j$ denotes that i and j belong to the same cluster.

Measurement of $\sum_{ij} s_i s_j$

By applying the improved estimator for measuring a spatial correlation function $\langle s_i s_j \rangle$, we find a rule for measuring the thermal average of $\sum_{ij} s_i s_j$:

$$\begin{aligned}
\Big\langle \sum_{ij} s_i s_j \Big\rangle &= \sum_{ij} \langle s_i s_j \rangle \\
&= \sum_{ij} \delta_{i\sim j} = \sum_{\substack{\text{all} \\ \text{clusters k}}} \sum_{i\in k} \underbrace{\sum_{j} \delta_{i\sim j}}_{s(k)} \\
&= \sum_{\substack{\text{all} \\ \text{clusters k}}} s(k) \underbrace{\sum_{i\in k} 1}_{s(k)} \\
&= \sum_{\substack{\text{all} \\ \text{clusters } k}} s(k)^2 \qquad (3.25)
\end{aligned}$$

where $s(k)$ denotes the size of cluster k.

Measurement of the magnetization

The magnetization M of a system is defined via

$$\frac{M}{N} = \lim_{H\to 0+} \lim_{N\to\infty} \frac{1}{N} \Big\langle \sum_i s_i \Big\rangle \qquad (3.26)$$

We mention that in this equation the limits must not be exchanged! In an infinite Ising system M equals the fraction of sites in the infinite cluster (it can be proved that there is only one infinite cluster) [15].

Therefore an improved estimator with the correct thermodynamic limit for measuring the magnetization is to measure the size of the *largest* cluster on the lattice, divided by the lattice volume.

Measurement of the susceptibility

By using eq. 2.6 we find

$$\chi_N = \Big\langle \sum_{ij}^{N} s_i s_j \Big\rangle - M^2 \qquad (3.27)$$

which approaches χ in the thermodynamic limit $N \to \infty$. Since we have found rules for measuring $\big\langle \sum_{ij}^{N} s_i s_j \big\rangle$ and M, χ can be obtained immediately.

3.7.3 The Wolff algorithm

Wolff [16] described an algorithm, similar to that of Swendsen and Wang. Only step 3 of the Swendsen-Wang algorithm is modified, the other steps remain the same:

- Step 3: One cluster is determined at random and for all the spins in the selected cluster a new common spin value is determined at random; the old value excluded. In case of the Ising model this means that the spins are flipped with probability 1

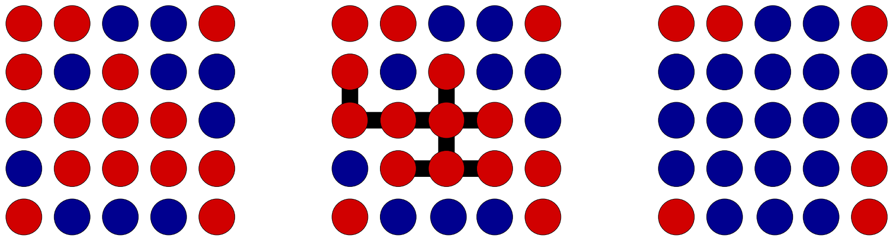

Figure 3.5: Schematic view of the Wolff algorithm: original spin configuration → intermediate bond configuration → new spin configuration.

Whereas in the Swendsen-Wang procedure all the clusters are possible subjects to a spin-flip at each step, in the Wolff method it is only the cluster containing the starting spin. This means that larger clusters are subjected to such flips more often in the Wolff procedure. This greater effort in the larger clusters makes the algorithm more effective. The Wolff algorithm shows a critical dynamical exponent $z^E_{int} = 0.25$ (resp. logarithmic scaling) in two dimensions and $z^E_{int} = 0.29$ in three dimensions [11] for the Ising model.

3.7.4 Geometric cluster Monte Carlo Simulation

J. R. Heringa and H.W. Blöte introduced a cluster algorithm based on geometric transformations [2]. This algorithm may employ reflections, inversions, translations and rotations, provided these are self-inverse and global symmetries of the Hamiltonian. It interchanges spins between two clusters which are related by the symmetry employed.

Let sites i, j and k map on i', j' and k' under the symmetry transformation. We denote the energy difference (divided by kT) when a neighbor k of i is interchanged with k' as Δ_{ik}. For the Ising model, $\Delta_{ik} = \beta J(s_i s_k + s_{i'} s_{k'} - s_i s_{k'} - s_{i'} s_k)$. The algorithm involves the following steps:

1. Choose a random lattice site i; i and i' belong to the same cluster .

[2] we follow reference [10]

2. Interchange s_i and $s_{i'}$.

3. For all neighboring sites k of i that do not (yet) belong to the cluster do the following:

   (a) If $\Delta_{ik} > 0$ do the following with probability $1 - e^{-\Delta_{ik}}$:

   - interchange s_k and $s_{k'}$ (k and k' included in the cluster)
   - write k in a list of addresses (the stack).

   (b) If $\Delta_{ik} \leq 0$, do nothing.

4. Read an address j from the stack.

5. Execute the steps listed under (3), substituting j for i.

6. Erase the address j from the stack.

7. Repeat steps (4)-(6) until the stack is empty.

When the stack is empty, the cluster is completed and moved. This algorithm satisfies detailed balance (see reference [10]), moreover one can show [10] that the cluster sizes scale similar as in the Wolff case. This method also shows a small critical dynamical exponent $z^E = 0.21$ in three dimensions [10] for the Ising model. This algorithm effectively suppresses critical slowing down at least for a number of Isinglike models [10]. It opens new possibilities for applications to models that were outside the reach of cluster simulations. Furthermore, one may investigate different ensembles, for instance, the constant magnetization ensemble in the case of the Ising model.

3.7.5 Joint Representation Cluster algorithm for the Ising model

H.G.Evertz, H.M. Erkinger and W. von der Linden [3] introduced an algorithm based on the joint representation eq.1.20. This representation leads to relations between the two kinds of bond clusters by considering the *conditional probabilities* between $\{f\}$ and $\{b\}$ which follow from eq.1.20:

- Given a configuration of FK-bonds $\{f\}$
  - if $f_{ij} = 0$, then $b_{ij} = 0$ with probability $\frac{1-v}{1-v} = 1$
  - if $f_{ij} = 1$, then any $\{b\}$ that satisfies the closed path constraint $\partial b = 0$ is equally probable

- Given a configuration of HT-bonds $\{b\}$
  - if $b_{ij} = 0$, then $f_{ij} = 1$ with probability $\frac{v}{v+(1-v))} = v$

– if $b_{ij} = 1$, then $f_{ij} = 1$ with probability $\frac{v}{v} = 1$

These relations imply that high-temperature-clusters are strictly contained within Fortuin-Kasteleyn clusters [3].

The algorithm is implemented as follows [3]

1. We start with a lattice where the dynamic variables consist of the bonds f_{ij}[4]. We are allowed to start with any bond-configuration we like. This bond configuration does not have to satisfy the High-Temperature constraint (=only closed loops). We will call this bond-pattern the *graph of deletions*.

2. The next task is to map the graph of deletions G to an allowed bond-configuration B in the High-Temperature representation in such a way that the resulting configuration consists of closed loops only. This construction is described in [17], [3] and[18].

3. Having constructed a valid High-Temperature configuration, we are now able to measure the energy according to eq.4.17, but not the quantities magnetization, susceptibility and spatial correlation functions.

4. We create a new graph of deletions G', according to the probabilities described above. Now we can measure the magnetization, susceptibility and spatial correlation functions by using the improved estimators of the Swendsen-Wang algorithm 3.7.1. Finally we jump to 2.

Similar to the Swendsen-Wang method, this algorithm can also be used in a single cluster version [16]. The algorithm can also be used with embeddings of the Ising model in other systems like the O(N) model. This algorithm shows a dynamical critical exponent $z^E_{exp} = 0.31 \pm 0.01$ in two dimensions and $z^E_{exp} = 0.65 \pm 0.04$ in three dimensions.

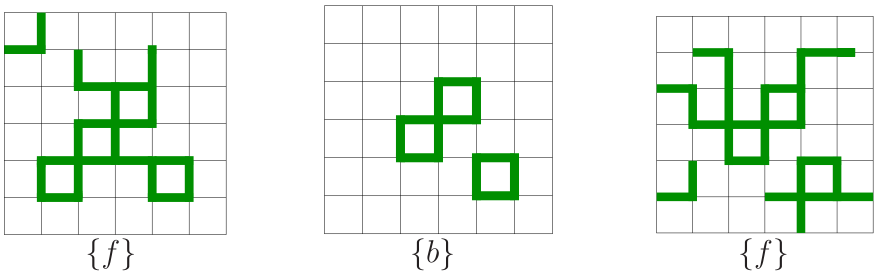

Figure 3.6: Schematic view of the Joint Representation algorithm: old FK-cluster configuration → HT-cluster configuration → new FK-cluster configuration

[3]we strictly follow [17]

[4]We will use the term “bond” both for the elements f_{ij} of F and b_{ij} of B.

3.8 Worm Algorithms for Classical Statistical Models

"Worm algorithms", introduced by N. Prokof'ev and B. Svistunov [19], utilize the idea of updating the closed-path configurations of a High Temperature representation (see chapter 1.4) through the motion of end points of a non closed path. This approach is based on principles which differ radically from cluster methods and has a different range of applicability.

In order to develop the algorithm, we start from the High-Temperature representation eq. 1.16 of the Ising-Model. First we rewrite the Hamiltonian (we omit factors k_b)

$$\frac{-\mathcal{H}}{T} = \sum_{b=\langle i,j\rangle} \mathcal{H}_{ij} = K s_i s_j \tag{3.28}$$

where $K = \beta J$ is the dimensionless coupling parameter between the spin variables and $b = \langle i, j\rangle$ refers to the lattice bonds between nearest neighbor-sites i, j. We will also use another (identical) notation to label bonds by specifying the lattice site and the direction toward its nearest neighbor[5], ν, i.e, $b = (i, \nu)$. Since $\mathcal{H}$ consists of a sum of bond Hamiltonians, the corresponding Boltzmann-weight $e^{-\mathcal{H}/T}$ factorizes in terms of exponentials for each bond. By expanding each exponential in a Taylor series in powers of K we arrive at

$$e^{-\mathcal{H}/T} = \prod_b \left(\sum_{N_b=0}^{\infty} \frac{K^{N_b}}{N_b!} (s_i s_j)^{N_b} \right) \tag{3.29}$$

In what follows we will consider summation indices $\{N_b\}$ as integer variables $N_b = 0, 1, 2, ...$ living on bonds and call them "bond configuration". Now we can write

$$\begin{aligned} Z &= \sum_{\{s_i\}} \prod_b \left(\sum_{N_b=0}^{\infty} \frac{K^{N_b}}{N_b!} (s_i s_j)^{N_b} \right) \\ &= \sum_{\{s_i\}} \sum_{\{N_b\}} \prod_b \frac{K^{N_b}}{N_b!} (s_i s_j)^{N_b} \end{aligned} \tag{3.30}$$

[5]With these notations we follow [19]

For each configuration of bonds, $\{N_b\}$, the spin summation of $e^{-\mathcal{H}/T}$ factorizes,

$$
\begin{aligned}
Z &= \sum_{\{s_i\}} \sum_{\{N_b\}} \prod_b \frac{K^{N_b}}{N_b!} (s_i s_j)^{N_b} \\
&= \sum_{\{N_b\}} \left[\left(\prod_b \frac{K^{Nb}}{N_b!} \right) \left(\sum_{\{s_i\}=\pm 1} \prod_b (s_i s_j)^{N_b} \right) \right] \\
&= \sum_{\{N_b\}} \left[\left(\prod_b \frac{K^{Nb}}{N_b!} \right) \left(\prod_i Q(k_i) \right) \right] \qquad (3.31)
\end{aligned}
$$

with

$$
Q(k_i) = \sum_{s_i=\pm 1} s_i^{k_i} \text{ where } k_i = \sum_{\nu} N_{i,\nu} \qquad (3.32)
$$

Graphically, each bond state can be represented by lines ascribed to bonds connecting next neighbor lattices sites. The number of lines on the bond is equal to N_b. By symmetry, $Q(k_i)$ is nonzero only if the number of bond lines k_i connected to the site i is *even*. Hence we can rewrite the partition function

$$
Z = \sum_{\{N_b\} \in CP} W_Z(\{N_B\}) \qquad (3.33)
$$

where CP denotes closed path configurations, i.e. configurations where the bonds on the lattice form closed loops. The bond configuration weight W_Z is then given by

$$
W_Z(\{N_B\}) = \left(\prod_b \frac{K^{N_b}}{N_b!} \right) \left(\prod_i Q(k_i) \right) \qquad (3.34)
$$

We define the function

$$
g(\mathbf{i}_1, \mathbf{i}_2) = \sum_{\{s_i\}} s_{i_1} s_{i_2} e^{-\mathcal{H}/T} \qquad (3.35)
$$

where $\mathbf{i}_1$ and $\mathbf{i}_2$ are vectors pointing at lattice sites i_1, i_2. This function is related to the spin-spin correlation function via

$$
G(\mathbf{i}_1, \mathbf{i}_2) = \langle s_{i_1} s_{i_2} \rangle = \frac{g(\mathbf{i}_1, \mathbf{i}_2)}{Z} \qquad (3.36)
$$

Using the same trick with the expansion of $e^{-\mathcal{H}/T}$ and summation over spin variables, we arrive at bond configurations for g which differ from those contributing to the partition function in only one respect: There are two special sites i_1 and i_2 with an odd number of bond lines connected to them if $i_1 \neq i_2$ (if $i_1 = i_2$ the bond configuration is again a closed path). These points are the only places were a path may start and end. We will abbreviate the configuration space for the g function by $\mathcal{G}$: it includes the set of all bond configurations $\{N_b\}$ that either consists of closed paths $\partial N_b = 0$ or have an even number of bonds incident on all but two sites $\partial N_b = \{i_1, i_2\}$. The $\mathcal{G}$-configuration weights are given by the same expression 3.34, provided

$$k_i = \delta_{i,i_1} + \delta_{i,i_2} + \sum_{\nu} N_{i,\nu} \tag{3.37}$$

.

The worm algorithm emerges as an idea to update $\mathcal{G}$-configurations through the space of motion of the end points i_1 and i_2, i.e. during our sampling we produce configurations that contribute either to $g(\mathbf{i}_1 - \mathbf{i}_2)$ (open path configurations) or to the partition function Z, respectively $g(0)$ (closed path configurations). One may consider the end point i_1 as if it were the tip of a “magic pen”: when i_1 is shifted to one of its neighboring sites, the pen “draws” a new or “erases” an existing bond line (increases or decreases the bond number by one). When $i_1 = i_2$, then we get a contribution to Z, i.e. we have performed an update of the previous CP-configuration to a new one by means of several intermediate steps. At this point one is free to start drawing/erasing bond lines from any lattice site. Thus the algorithm employs two elementary updates which are proposed in the context of a given configuration:

- If $i_1 = i_2$, then with probability p_0 we propose to “move” both end points to site i_0 selected at random among all L^{dim} lattice sites. With probability $(1 - p_0)$ we “shift” the endpoint i_1 to one of the neighboring sites by selecting the direction ν of the shift at random. Formally, $p_0 \in (0, 1)$ and arbitrary otherwise.

- If $i_1 \neq i_2$ we always propose to shift i_1

The probability for a move-update in general involves the ratio of Q functions,

$$P_{\text{move}}(i \to j) = \frac{Q(k_j + 2)}{Q(k_j)} \frac{Q(k_i - 2)}{Q(k_i)} \tag{3.38}$$

Note that for the Ising model ($Q(k) = 2$) moves are always accepted. The shift-updates change the position of the end point i_1 and simultaneously increase/decrease the bond state by 1. The probability of increasing/decreasing the bond state is 1/2. The “shift probabilities” are given by:

$$P_{\text{shift}}(i \to j, N_b \to N_b + 1) = \min\left(1, r\frac{K}{N_b+1}\frac{Q(k_j+2)}{Q(k_j)}\right) \tag{3.39}$$

$$P_{\text{shift}}(i \to j, N_b \to N_b - 1) = \min\left(1, r\frac{N_b}{K}\frac{Q(k_i-2)}{Q(k_i)}\right) \tag{3.40}$$

where

$$r = \begin{cases} 1/[2(1-p_0)], \text{ if } i_1 = i_2 = i \\ 2(1-p_0), \text{ if } i_1 = i, i_2 = j \wedge \langle i,j\rangle \\ 1, \text{ otherwise} \end{cases} \tag{3.41}$$

As usually, with $\langle i,j\rangle$ we denote that i and j are nearest neighbors. The practical implementation of the algorithm is as follows: One starts from an arbitrary bond configuration, e.g., from N_b=0 for each b, and then performs a set of procedures which forms a standard Metropolis update in a cycle:

1. We select an update: If $i_1 \neq i_2$ then shift i_1 according to the probabilities P_{shift} above . If $i_1 = i_2$ we either move i_1 and i_2 or shift i_1 as described above.

2. We calculate the acceptance probabilities using eq. 3.38-3.41. If the update is accepted we change the configuration accordingly, otherwise we go back to step 1.

3. Now we collect statistics for various quantities using improved estimators.

The move-update satisfies and the shift probabilities are solutions of the detailed balance equation 3.13 (we refer to [19]).

Measurement of spatial correlation functions

In eq. 3.36 we introduced the relation between the spatial correlation function $G(\mathbf{i})$ and the function $g(\mathbf{i}_1 - \mathbf{i}_2)$. We now define $g(i)$ as the average of $g(\mathbf{i}_1 - \mathbf{i}_2)$ over $\mathbf{i}_2$, with $|\mathbf{i}_1 - \mathbf{i}_2| = i$. We make use of the fact that for large $|\mathbf{i}_1 - \mathbf{i}_2|$, the correlation function depends only on the distance $|\mathbf{i}_1 - \mathbf{i}_2|$. Since we generate bond configurations $\{N_b\}$ by a Monte Carlo process with probabilities proportional to the weights of the configurations, the diagram with $|\mathbf{i}_1 - \mathbf{i}_2| = i$ contributes unity to the stochastic MC sum for $g(i)$. The MC estimator for $g(i)$ and (up to a n unknown common normalization) Z defined on $\mathcal{G}$ are (for details see [19])

$$g(i \mid \mathcal{G}) = \delta_{|\mathbf{i}_2 - \mathbf{i}_1|, i} \tag{3.42}$$

$$Z(\mathcal{G}) = \frac{Q(k_{i_1}-2)}{Q(k_{i_1})}\delta_{\mathbf{i}_2,\mathbf{i}_1} \tag{3.43}$$

The ratio $g(i \mid \mathcal{G})/Z(\mathcal{G})$ yields to the physical result for $G(i)$.

Measurement of the susceptibility

The susceptibility is defined as :

$$\chi = \langle (\sum_i s_i)^2 \rangle - \langle M \rangle^2 \tag{3.44}$$

Since $\langle M \rangle_{T=T_c} = 0$ we can write

$$\begin{aligned} \chi_{T=T_c} &= \left\langle \left(\sum_i s_i\right)^2 \right\rangle \\ &= \left\langle \sum_i \sum_R s_i s_{i+R} \right\rangle \\ &= N \sum_R \langle s_i s_{i+R} \rangle \end{aligned} \tag{3.45}$$

and thus we can rewrite the spin susceptibility:

$$\frac{\chi}{N}_{T=T_c} = \sum_R \frac{g(R)}{Z} \tag{3.46}$$

Measurement of the energy

The bond energy may be computed in two ways [19]:

- as a nearest neighbor sum

  $$\frac{E}{N} = \frac{K}{2} \frac{q \cdot g(1)}{Z} \tag{3.47}$$

  where q is the number of nearest neighbors of a lattice site.

- using E=e/Z, which is nonzero only for the CP configurations contributing to Z, with the Monte Carlo estimator:

  $$e(\mathcal{G}) = \frac{Q(k_{i_1} - 2)}{Q(k_{i_1})} \delta_{i_1,i_2} \sum_b N_b \tag{3.48}$$

  where the first factor accounts for the reweighting between $\mathcal{G}$ and CP, and the last factor follows from eq. 2.2

The Ising model has a special property which can be used to enhance the efficiency by truncating the configuration space. Namely by using identity 1.12 one can restrict the bond summation to $N_b = 0, 1$ only. The corresponding modification of the scheme is shown in chapter 4 and one of the main tasks of this diploma thesis.

While applicability of most cluster algorithms is restricted (see the preceding chapter), the Worm Algorithm is applicable to a wide class of systems. Moreover it shows very small autocorrelation times and in most cases (such as the Ising model and the XY-model in two and three dimensions, the Gaussian model in three dimensions) even a logarithmic dependence $\tau^E_{int} \sim \ln L$ (potential dependence $\tau^E_{int} \sim L^z$ for the q=3 Potts model with $z = 0.55$ [19]). A discussion of the specific implementations of the Worm Algorithm for $\mid \psi \mid^4, XY$ and $q = 3$ Potts models can be found in [19]. For the Ising model Prokof'ev and Svistunov found : $\tau_{int} = -7.2 + 6.2 \ln(L)$ and $z(L = 64) = 0.18$ in two dimensions and $\tau_{int} = -4.3 + 9.2 \ln(L)$ and $z(L = 512) = 0.25$ in three dimensions [19][6]. It remains to be mentioned that the Worm Algorithm can also be applied to quantum mechanical systems [20].

[6]A detailed comparison of the Worm Algorithm with cluster algorithms for the Ising model is shown in chapter 5.3

Chapter 4

The worm algorithm applied to the Ising model

The Worm algorithm can be applied to the Ising model in two different ways:

The first way is to use the High-Temperature representation 3.33 with configuration weight 3.34 setting $Q(k) = 2$ and then follow the algorithm described in chapter 3.8. The other way will be described in this chapter:

We start from the High-Temperature representation 1.15 using identity 1.12:

$$Z = (\cosh\beta)^{N_{nn}} 2^N \sum_{\{b_i\},\partial b=0} v^{n_b}$$

The configurations contributing to the partition function are only the closed-path configurations (see chapter 1.4). As already mentioned in chapter 3.8, in this representation either there is *one* bond connecting two lattice sites, or there is *no* bond, e.g. $b_{ij} = 0, 1$. The configuration-weight is now defined as:

$$W_Z(\{b\}) = v^{n_b} = (\tanh\beta J)^{n_b} \tag{4.1}$$

where n_b is the number of bond variables with value $b_{ij} = 1$.

Note that the application of the Worm Algorithm on frustrated antiferromagnetic systems leads to a sign problem, because negative configuration weights do appear.

4.1 The algorithm

The algorithm is analogous to that introduced in chapter3.8. Again we denote the two end points of an open configuration by i_1, i_2

1. We start at a random position $i_1 = i_2 = i_{start}$ and iterate the following steps:

2. (a) If $i_1 = i_2$ we transfer the obtained values of our observables $E, \chi, G(i)$ and M to the statistics routine.
      i. With probability

$$p_0 \in (0,1) \tag{4.2}$$

      we "move" both end points i_1, i_2 to a new random lattice site and go to step 2. One elementary step has been carried out.
      ii. with probability $1 - p_0$ we go to step 3
   (b) If $i_1 \neq i_2$ go to step 3

3. We choose a neighboring site j by random $P_{nn}(j \mid \langle i_1, j \rangle) = 1/N_{nn}$.

4. We suggest to shift the end i_1 to j. This shift will increase/decrease the bond state by one. Since $b_{i,j} = 0, 1$ we can only suggest to set a bond $b_{i,j} = 1$ if there was no bond (connecting i and j) before and vice versa. The acceptance-probability is given by

$$P_{shift}(i \rightarrow j, b_{ij} = 0 \rightarrow b_{ij} = 1) = r \tanh \beta = rv \tag{4.3}$$

$$P_{shift}(i \rightarrow j, b_{ij} = 1 \rightarrow b_{ij} = 0) = r \tag{4.4}$$

   with

$$r = \begin{cases} (1 - p_0) & \text{... if } j = i_2 \\ 1 & \text{... else} \end{cases} \tag{4.5}$$

   If the shift is not accepted, we keep the current configuration.

5. We add the contribution of the configuration now achieved to the susceptibility χ and to the corresponding correlation function $G(i)$ (see 4.2). An "elementary step" has been carried out.

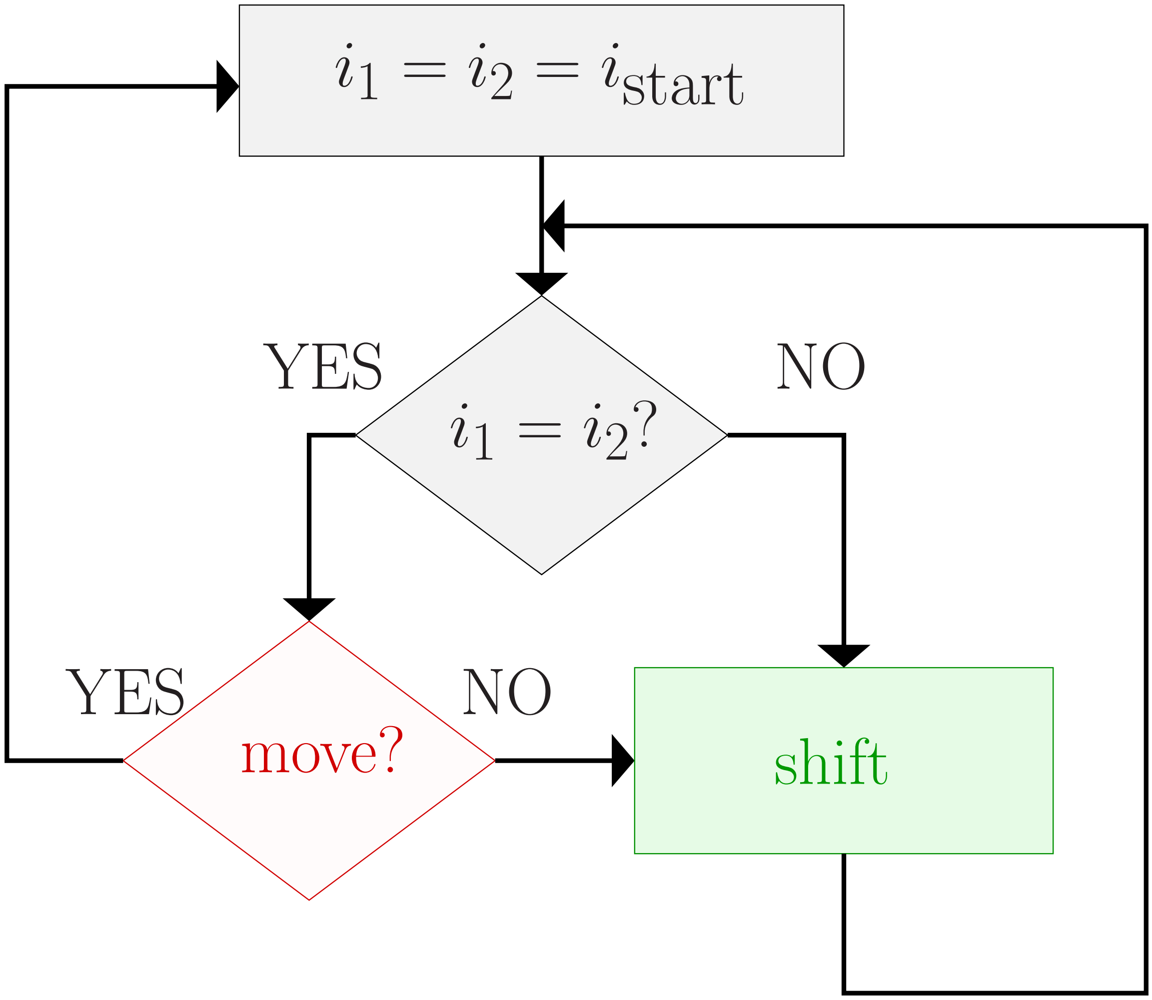

Figure 4.1: Flow chart of the Worm-Algorithm applied to the Ising model

4.1.1 Detailed balance

It is obvious that with the transition-probabilities introduced above every bond-configuration can be reached, thus our algorithm is ergodic. It remains to be shown that detailed balance is satisfied. In order to prove that eq. 3.13 is always satisfied by using the probabilities 4.2-4.4 we distinguish between four different cases:

- Starting from a closed configuration C_1 with $i_1 = i_2 = i$, shown in figure 4.2, we select with probability p_0 to *move* both end points i_1, i_2 to another lattice site chosen at random. The probability to choose $i_1 = i_2 = j$ is simply one divided by the lattice volume N. The weight 4.1 does not change since the number of bonds remains the same. Thus the detailed balance equation is given by:

$$\begin{aligned} P(C_1 \to C_2)W(C_1) &= P(C_2 \to C_1)W(C_2) \\ p_0 \frac{1}{N} v^{n_b} &= p_0 \frac{1}{N} v^{n_b} \end{aligned} \tag{4.6}$$

$$q.e.d.$$

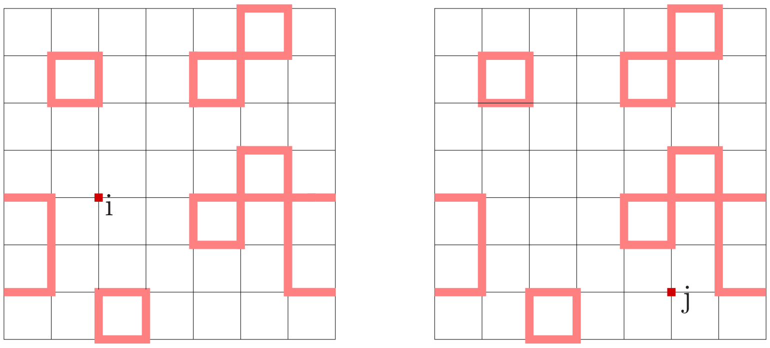

Figure 4.2: Configuration C_1 (left picture): $i_1 = i_2 = i$ and configuration C_2(right picture): $i_1 = i_2 = j$

- We start from an open configuration C_3 with $i_1 = j, i_2 = i$ and configuration weight $W(C_3) = v^{n_b}$ (shown in figure 4.3). With probability $P_{nn} = 1/N_{nn}$ we choose the neighboring lattice site $k \neq i$. The probability to set a bond connecting j with k is given by eq. 4.3 with $r = 1$.

  Now we consider this transition the opposite way: We start from configuration C_4 with $i_1 = k, i_2 = i$ and weight $W(C_4) = v^{n_b+1}$. In order to obtain configuration C_3 again, we have to choose the neighboring site j (again with probability

$P_{nn} = 1/N_{nn}$) and erase the existing bond b_{kj} according to eq. 4.4. Substituting these steps into eq. 3.13 shows that the detailed balance condition is satisfied

$$\begin{aligned} P(C_3 \to C_4)W(C_3) &= P(C_4 \to C_3)W(C_4) \\ \frac{1}{N_{nn}} v v^{nb} &= \frac{1}{N_{nn}} \cdot 1 \cdot v^{n_b+1} \\ & \qquad\qquad q.e.d. \end{aligned} \tag{4.7}$$

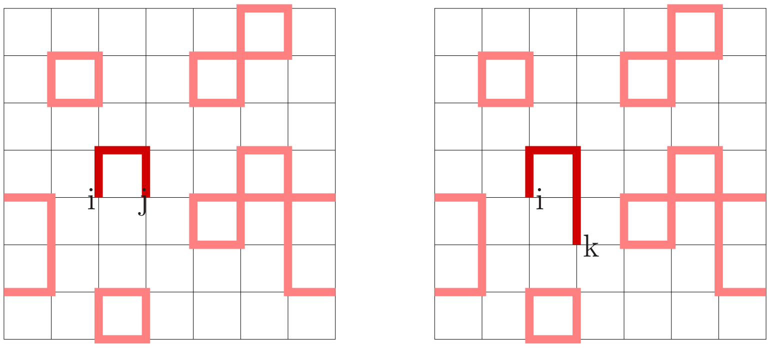

Figure 4.3: Configuration C_3 (left picture): $i_1 = j$, $i_2 = i$ and configuration C_4 (right picture): $i_1 = k$, $i_2 = i$

- Another situation is shown in figure 4.4. We start again with an open configuration C_5 with $i_2 = j, i_1 = i$ and configuration weight $W(C_5) = v^{n_b}$. With probability $P_{nn} = 1/N_{nn}$ we choose the neighboring site i of j and set a bond between the two sites according to the probability eq. 4.3 (with $r = (1 - p_0)$ since our random neighboring site equals the starting point $i_2 = i$).

  On the other hand we open the closed configuration C_6 with probability $(1-p_0)$ and erase the existing bond b_{ij} with probability $1/N_{nn}$ (for choosing j) times one (the probability to erase the existing bond equals one). Taking into account the weights $W(C_5) = v^{n_b}$ and $W(C_6) = v^{n_b+1}$, the detailed balance condition turns out to be satisfied:

$$\begin{aligned} P(C_5 \to C_6)W(C_5) &= P(C_6 \to C_5)W(C_6) \\ \frac{1}{N_{nn}} v \cdot (1-p_0) v^{nb} &= (1-p_0)\frac{1}{N_{nn}} \cdot 1 \cdot v^{n_b+1} \\ & \qquad\qquad q.e.d. \end{aligned} \tag{4.8}$$

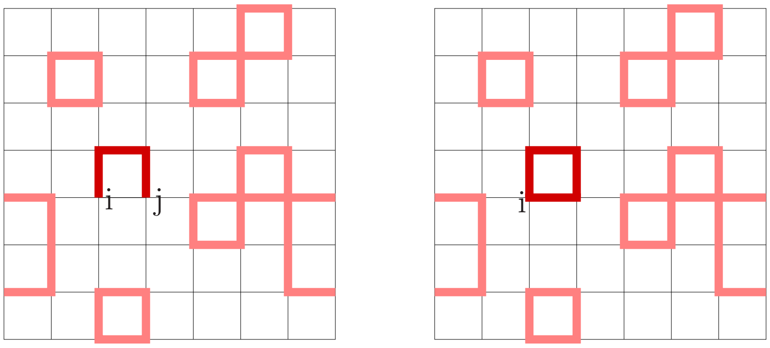

Figure 4.4: Configuration C_5: $i_1 = j$, $i_2 = i$ and configuration C_6: $i_1 = i_2 = i$

- Finally we consider the situation shown in fig. 4.5. In configuration C_7 we have $i_2 = j, i_1 = i$, where i, j are nearest neighbors. The configuration weight is $W(C_7) = v_b^n$. With probability $P_{nn} = 1/N_{nn}$ we choose the neighboring site i of j and remove the existing bond with probability $r = 1 - p_0$.

  On the other hand we open the closed configuration C_8 (configuration weight $W(C_8) = v^{n_b - 1}$) with probability $(1 - p_0)$ and and set bond b_{ij} with probability $1/N_{nn}$ (for choosing j) times $r \cdot v$ with $r = (1 - p_0)$. Thus the detailed balance equation is given by:

$$\begin{aligned} P(C_7 \to C_8)W(C_7) &= P(C_8 \to C_7)W(C_8) \\ \frac{1-p_0}{N_{nn}} v^{n_b} &= v \frac{1-p_0}{N_{nn}} v^{n_b - 1} \qquad\qquad q.e.d. \end{aligned}$$

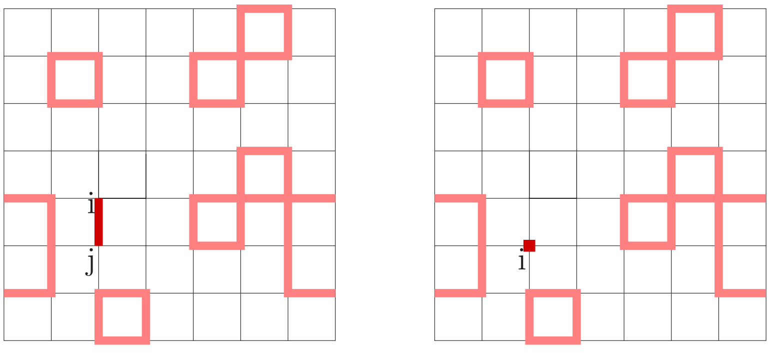

Figure 4.5: Configuration C_7 (left picture): $i_1 = i, i_2 = i$ and configuration C_8(right picture): $i_1 = i_2 = i$

4.2 Measurement of physical quantities

4.2.1 Measurement of the spatial correlation function

Again write $G(\mathbf{i}_1, \mathbf{i}_2) = g(\mathbf{i}_1, \mathbf{i}_2)/Z$ with

$$g(\mathbf{i}_1, \mathbf{i}_2) = \sum_{\{s_i\}} s_{i_1} s_{i_2} e^{-H/T} \tag{4.9}$$

Now we expand $\sum_{\{s_i\}} s_{i_1} s_{i_2} e^{-H/T}$ with expansion-parameter $v = \tanh \beta J$ analogous to eq. (1.13)-(1.14):

$$\begin{aligned}
g(\mathbf{i}_1, \mathbf{i}_2) &= \sum_{\{s_i\}} s_{i_1} s_{i_2} e^{-H/T} \\
&= \sum_{\{s_i\}} \prod_{\langle i,j \rangle} s_{i_1} s_{i_2} \cosh(\beta J)[1 + s_i s_j \tanh(\beta J)] \\
&= (\cosh \beta J)^{N_{nn}} \sum_{\{s_i\}} \prod_{\langle i,j \rangle} s_{i_1} s_{i_2} [1 + s_i s_j v] \\
&= (\cosh \beta J)^{N_{nn}} \sum_{\{s_i\}} \left(s_{i_1} s_{i_2} + v s_{i_1} s_{i_2} \sum_{\langle i,j \rangle} s_i s_j + v^2 s_{i_1} s_{i_2} \sum_{\langle i,j \rangle, \langle k,l \rangle} s_i s_j s_k s_l + ... \right)
\end{aligned} \tag{4.10}$$

Summation over the spins, and taking into account the two additional spins $s_{i_1} s_{i_2}$ that now appear at every power of v leads us (by using the same arguments as in 1.14) to

$$g(\mathbf{i}_1, \mathbf{i}_2) = (\cosh \beta)^{N_{nn}} 2^N \sum_{\{b_i\}, \partial b = i_1, i_2} v^{n_b} \tag{4.11}$$

This means that every "open" configuration with $\mid \mathbf{i}_1 - \mathbf{i}_2 \mid = i$ contributes to the stochastic sum for $g(i)$, i.e.

$$g(i \mid \mathcal{G}) = \delta_{i_1, i_2} \tag{4.12}$$

The measurement is carried out as follows: We start with $g(i) = 0$. During our sampling we add up the number of open configurations with $\mid i_1 - i_2 \mid = i$ to the $g(i)$[1], with $i = 0, 1, 2..L/2$. If $i_1 = i_2$, the closed-path configuration contributes unity to

[1] the real values of i were rounded to the corresponding integer values

$g(0)$. Since $G(i) = g(i)/Z$ and the improved estimator for Z defined on $\mathcal{G}$ is

$$Z(\mathcal{G}) = \delta_{i_1,i_2} \tag{4.13}$$

the measurement is carried out whenever a closed configuration is reached, i.e. the actual values of the $g(i)$'s are transferred to the statistics-routine and subsequently reset to zero.

4.2.2 Measurement of the susceptibility

As mentioned in eq.3.46, the susceptibility at $T = T_c$ can be measured via the sum over all correlation functions:

$$\frac{\chi}{N} = \sum_i G(i) \tag{4.14}$$

Thus the susceptibility is measured by counting the configurations (i.e. "elementary steps") that are sampled until a closed configuration is reached again.

4.2.3 Measurement of the energy

As described in chapter 3.8, there are two ways of measuring the internal energy:

- The first way is to express the energy $E/N = -J\sum_{\langle i,j\rangle}\langle s_i s_j\rangle$ via the expectation of the spin-spin correlation function $G(1)$ in axial directions:

$$\begin{aligned}\frac{E}{N} &= -J\frac{1}{N}\sum_{\langle i,j\rangle}\langle s_i s_j\rangle \\ &= -J\frac{1}{N}N_{nn}\langle s_i s_j\rangle \\ &= -J\cdot d\cdot G(1)\end{aligned} \tag{4.15}$$

  (In the last step we used $N_{nn} = Nd$ on a hypercubic lattice with periodic boundary conditions.)

- The representation 1.15 of the partition function allows us to derive an expression for the internal energy per site:

$$
\begin{aligned}
\frac{E}{N} &= -\frac{J}{N}\frac{\partial \ln Z}{\partial \beta} = -\frac{J}{N}\frac{1}{Z}\frac{\partial Z}{\partial \beta} \\
&= -\frac{J}{N}\frac{1}{Z}\Bigg(N_{nn}\sinh\beta(\cosh\beta)^{N_{nn}-1}2^N\sum_{\{b\}} v^{n_b} + \\
&\qquad\qquad (\cosh\beta)^{N_{nn}}2^N\sum_{\{b\}}\frac{n_b}{\cosh^2\beta}v^{n_b-1}\Bigg) \\
&= -\frac{J}{N}\left(N_{nn}v + \frac{1}{Z}(\cosh\beta)^{N_{nn}}2^N\sum_{\{b\}}\frac{n_b}{\sinh\beta\cosh\beta}v^{n_b}\right) \\
&= -\frac{J}{N}\left(N_{nn}v + \frac{\langle n_b\rangle}{\sinh\beta\cosh\beta}\right) \qquad (4.16)
\end{aligned}
$$

Since the number of nearest-neighbor pairs on a hypercubic lattice with periodic boundary conditions and of spatial dimension d is given by $N_{nn} = N \cdot d$, eq. 4.16 becomes

$$
\frac{E}{N} = -\left(Jdv + \frac{J\langle n_b\rangle}{N\sinh\beta\cosh\beta}\right) \qquad (4.17)
$$

Chapter 5

Results without magnetic fields

We used the Worm Algorithm to simulate an Ising system in two and three dimensions. In all simulations the lattice was taken to be square, resp. simple cubic and to have periodic boundary conditions (this allows the loops of the High Temperature representation to wind around the lattice). In our simulation we measured the energy, the susceptibility and the spatial correlation functions up to distance $L/2$ in radial- and axial- direction. The thermalization time was taken to be 1/10 of the measurement-Monte Carlo sweeps. The expectation values of all these quantities agreed within statistical errors with results obtained from Wolff simulations. Further results are summed up in the following sections.

5.1 Evaluating the correlation function and autocorrelation times

The correlation function was computed via:

$$C(t) = \frac{\frac{1}{M}\sum_j^M Q(j)Q(j+t) - \frac{1}{M^2}\sum_j^M Q(j)\sum_j^M Q(j+t)}{\frac{1}{M}\sum_j^M Q(j)Q(j) - \frac{1}{M^2}\sum_j^M Q(j)\sum_j^M Q(j)} \tag{5.1}$$

where M denotes the number of measurements.

In order to obtain the exponential autocorrelation time τ_{exp}^Q, the function

$$f(t) = c \cdot e^{-t/\tau_{exp}} \tag{5.2}$$

was fitted to $C(t)$ in the interval $t \in [0.3\tilde{t}, 0.6\tilde{t}]$ where $\tilde{t}$ is the smallest distance t for which [1]

[1] We denote the error of the correlation function $C(t)$ at time t by $\sigma_C(t)$.

$$C(t+1) < 2.5 \cdot \sigma_C(t+1) \tag{5.3}$$

The integrated autocorrelation time τ_{int}^{Q} was then calculated self-consistently via

$$\tau_{int}^{Q} = \sum_{t}^{t^*} C(t) \tag{5.4}$$

with $t^* = \min(\tilde{t}, 3\tau_{int}^{Q})$ in the interval $[0, t^*]$ and analytically via the fitted function $f(t)$ in remaining interval $[t^*, \infty]$. The error estimation was calculated via the “Jackknife” method which is, for example, described in [17].

5.2 General remarks

In order to compare the Worm Algorithm to other Monte Carlo algorithms we evaluated the integrate d autocorrelation time τ_{int} of the energy measured via eq. 4.17. Though the energy measured via the correlation function (eq. 4.15) showed smaller integrated autocorrelation times (cp. eq. 5.5), the relative error was bigger, since we got worse statistics for $G(0)$ during a Monte-Carlo run.

The quantities are measured in *sweeps* where

$$\text{one sweep} = d \cdot L^d \text{elementary steps} \tag{5.5}$$

and d denotes the dimension of the system. Since our measurements are taken only at closed configurations we have to find a relation between elementary steps and closed configurations. Remembering that the average sum of spatial correlations is proportional to the average number of elementary steps that are needed to reach a closed configuration (eq. 3.45 and eq. 4.14), one can write:

$$\#\text{elementary steps} = \frac{1}{N}\langle \sum_{ij} s_i s_j \rangle \cdot \#\text{closed configurations} \tag{5.6}$$

The susceptibility and correlation functions at all distances show very small integrated correlation times. They seem to be uncorrelated after a few elementary steps so that an upper bound for the integrated autocorrelation time can be estimated via :

$$\tau_{int}(\chi, G(i)) \leq \frac{1}{N}\left\langle \sum_{ij} s_i s_j \right\rangle \text{ elementary steps} \tag{5.7}$$

These estimations are listed in table 5.1.

	L	τ_{int}^{max}/sweeps
dim=2		
	4	0.38
	8	0.33
	16	0.27
	32	0.22
	64	0.20
	128	0.17
	256	0.17
dim=3		
	4	0.11
	8	0.06
	16	0.03
	32	0.02
	64	0.01
	128	0.01

Table 5.1: Estimates (eq. 5.7) of upper bounds for the integrated autocorrelation times τ_{int}^{max} of the susceptibility and spatial correlation functions as a function of the lattice size L at T_c.

5.3 Autocorrelation time results

We did an autocorrelation time analysis for two- and three dimensional Ising systems. All measurements were taken at the critical temperature $T = T_c$. We analyzed the dependence of integrated autocorrelation time of the energy on the parameter p_0, eq.4.2. We found that for $p_0 \to 1$ the autocorrelation time τ_{int} of the energy is a monotone function which seems to take its maximum value at $p_0 = 1$. For very small values $p_0 \to 0$, τ_{int} increases again (see figure 5.1). At $p_0 = 0$ the Worm Algorithm becomes an infinite cluster simulation similar to [21]. It will be applicable to more models, however. Detailed properties of this approach remain to be investigated. For our simulations we chose $p_0 = 0.5$.

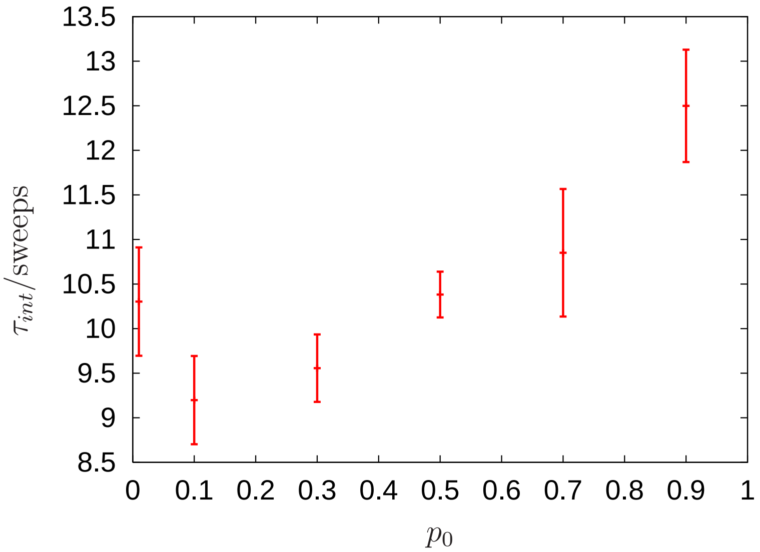

Figure 5.1: Integrated autocorrelation time τ_{int} of the energy as a function of the choice of the parameter p_0 for the 32×32 Ising model at T_c.

5.3.1 2d-Ising model

Energy

We found, that for the energy the integrated autocorrelation time τ_{int}^E agrees with the exponential autocorrelation time within their errors, i.e. $\tau_{int}^E \approx \tau_{exp}^E$. This is also visible in fig.5.2 where the autocorrelation function of energy of a 256×256 system is depicted. This correlation function (and also the correlation functions of the smaller systems, see appendix A) shows a single mode decay.

Our results for the integrated autocorrelation time of energy for the two dimensional Ising model are shown in table 5.2. They are slightly lower than those presented

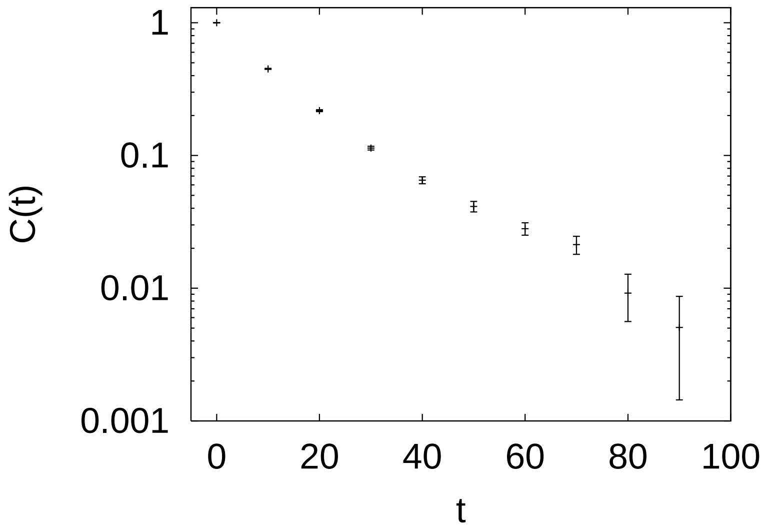

Figure 5.2: Autocorrelation function $C(t)$ of energy of a 256×256 system. The statistics of the Monte Carlo run can be extracted from table 5.2.

in [19]. In order to describe the dependence $\tau_{int}(L)$ we fitted different functions to our data. For the fit we used an implementation of the nonlinear least-square Marquardt-Levenberg algorithm.

- In the first picture of fig.5.3 our data is fitted to $\tau = a + b \cdot \ln(L)$. The values obtained are $a = -10.5 \pm 0.8, b = 6.0 \pm 0.2$ and $\chi_r^2 = 1.45$ [2]. This is the functional dependence claimed in the original worm paper [19].

- In the second picture of fig.5.3 the data points are assumed to show a behavior $\tau = a \cdot L^z$. We obtained $a = -2.7 \pm 0.2, z = 0.39 \pm 0.2$, and $\chi_r^2 = 2.3$.

- In the last picture of fig.5.3 we chose $\tau = a * (\ln L)^b$ behavior and obtained : $a = -1.215 \pm 0.008, b = 1.714 \pm 0.005$ and $\chi_r^2 = 0.01$

Thus any of these functions fit the data well. The best agreement is found for a logarithmic dependence $\tau = a * (\ln L)^b$ of the integrated autocorrelation time on the lattice size.

In figure 5.4 we compare the autocorrelation results of the energy using the Worm Algorithm with those of other cluster algorithms introduced in chapter 3.7. We point out that a comparison to the published exponential autocorrelation times of the Joint Representation Cluster algorithm (JCA) is possible since they agree with the integrated times within their statistical errors [17]. We found that the Worm algorithm

[2] $\chi_r^2 = \chi^2/f$ (f denotes the number of degrees of freedom)

shows higher autocorrelation times in two-dimensional systems than the Swendsen-Wang, the Wolff or the JCA method.

In a further step we made a detailed comparison between the Wolff- and the Worm Algorithm. In fig.5.5 we depicted the three possible functions that fit to the autocorrelation time data of the Worm Algorithm (shown in fig. 5.3) together with those of the Wolff Algorithm [3]. We found that even for very large lattice extensions the Wolff algorithm shows lower autocorrelation times which are about one order of magnitude lower than those of the Worm Algorithm. It thus works more efficiently.

However, to get a good estimation about the efficiency of an algorithm the amount of CPU time per sweep has to be taken into account. We tested an implementation of the Wolff Algorithm and the Worm Algorithm with regard to their amount of CPU-time. Table 5.3 shows this comparison. The CPU time for $1/V$ sweeps turned out to be about $0.005 - 0.006$ ms for the Worm Algorithm, while it is $0.002 - 0.003$ ms for the Wolff algorithm in our implementations[4]. Thus the Wolff algorithm gains about another factor of 2 with respect to the Worm algorithm.

[3] the corresponding data of the integrated autocorrelation function of energy was taken from [22]
[4] the Wolff data was obtained from Hans Gerd Evertz

L	τ_{int}/sweeps	$\Delta\tau_{int}$/sweeps	M/sweeps	t/s
4	3.66	0.10	50000	5
8	5.47	0.16	50000	20
16	7.71	0.12	820000	1000
32	10.24	0.15	250000	1300
64	13.96	0.29	500000	10250
128	18.19	0.37	500000	41000
256	22.93	0.39	270000	885000

Table 5.2: Integrated correlation time τ_{int} of the energy as a function of the lattice size L. M denotes the number of measurement sweeps that contributed to the statistics and t the CPU-time of the MC-run. The calculations were done on 100000 kFLOPS machines .

$L = 16$	$CPU/M[ms/\text{sweep}]$	$CPU/(M \cdot V)[ms/sweep]$
Worm	1.5	0.006
Wolff	0.8	0.002
$L = 32$		
Worm	5	0.005
Wolff	2.7	0.003

Table 5.3: Amount of CPU-time per sweep calculated for the Worm- and and the Wolff Algorithm (data taken on a 2d lattice at T_c). For the Wolff-Algorithm a sweep is defined per: one sweep = L^d elementary steps. M denotes the number of sweeps, V is the volume of the system. The right column us thus the time needed for 1 elementary step in the Wolff algorithm, resp. d elementary steps in the Worm

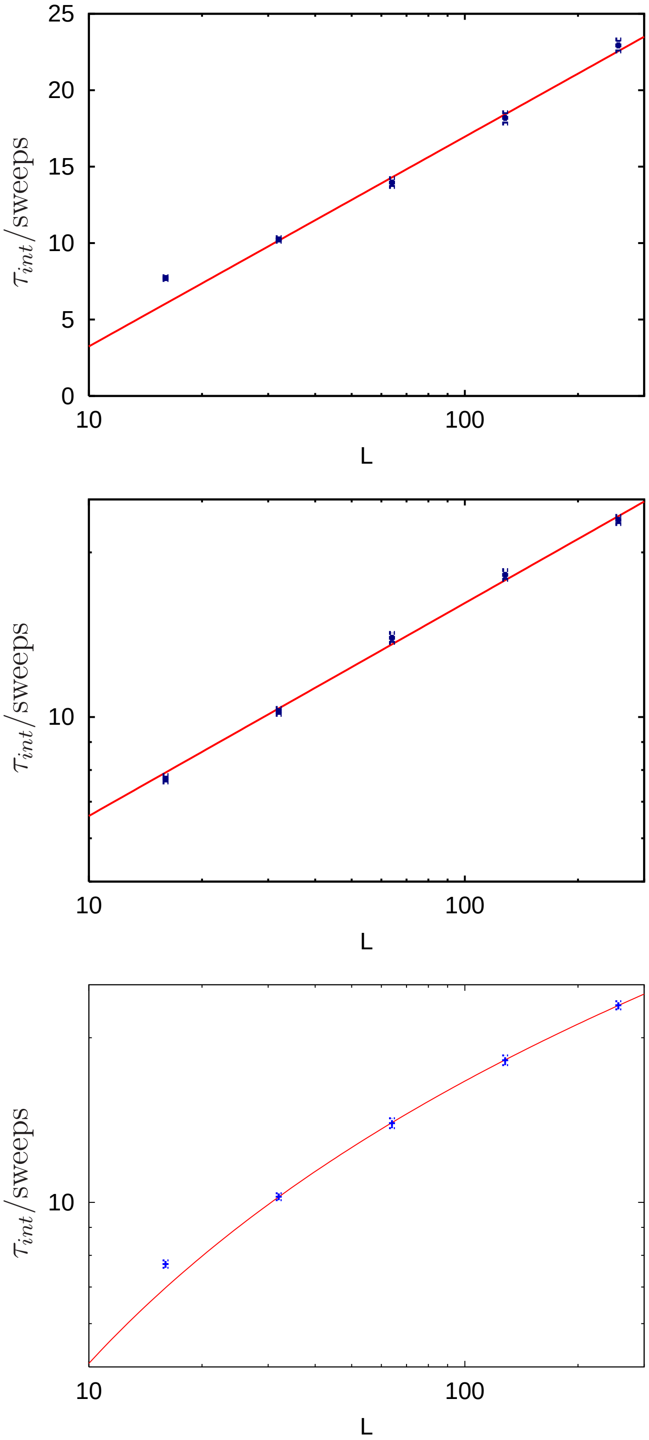

Figure 5.3: Dependence of the integrated autocorrelation time τ_{int} on the lattice size L in two dimensions. The data is fitted to $f(x) = -10.5 + 6\log(x)$ (top), $f(x) = -2.7x^{0.39}$ (center) and to $f(x) = -1.22\log(x)^{1.71}$ (bottom).

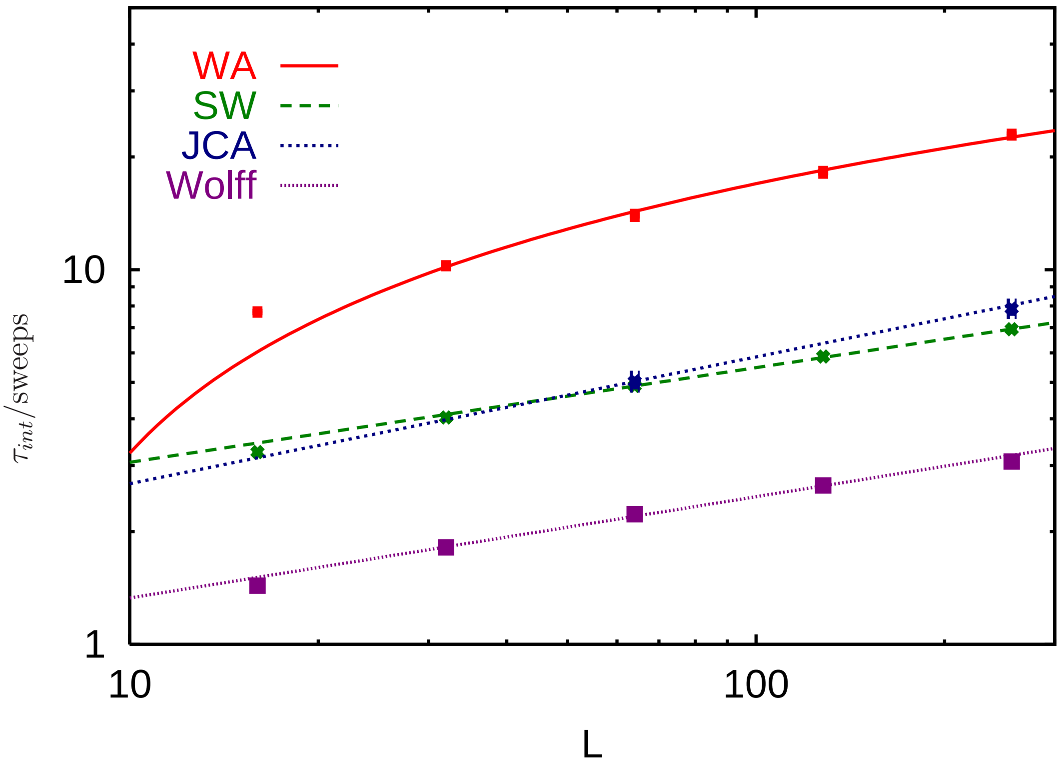

Figure 5.4: Comparison of the integrated autocorrelation times of the Worm-Algorithm with the Swendsen Wang- and the Wolff algorithm and the exponential autocorrelation time of the Joint Representation Algorithm (see text). The SW-data was obtained from [14], the Wolff-data from [22] and the data for the Joint Representation Algorithm can be found in reference [17].

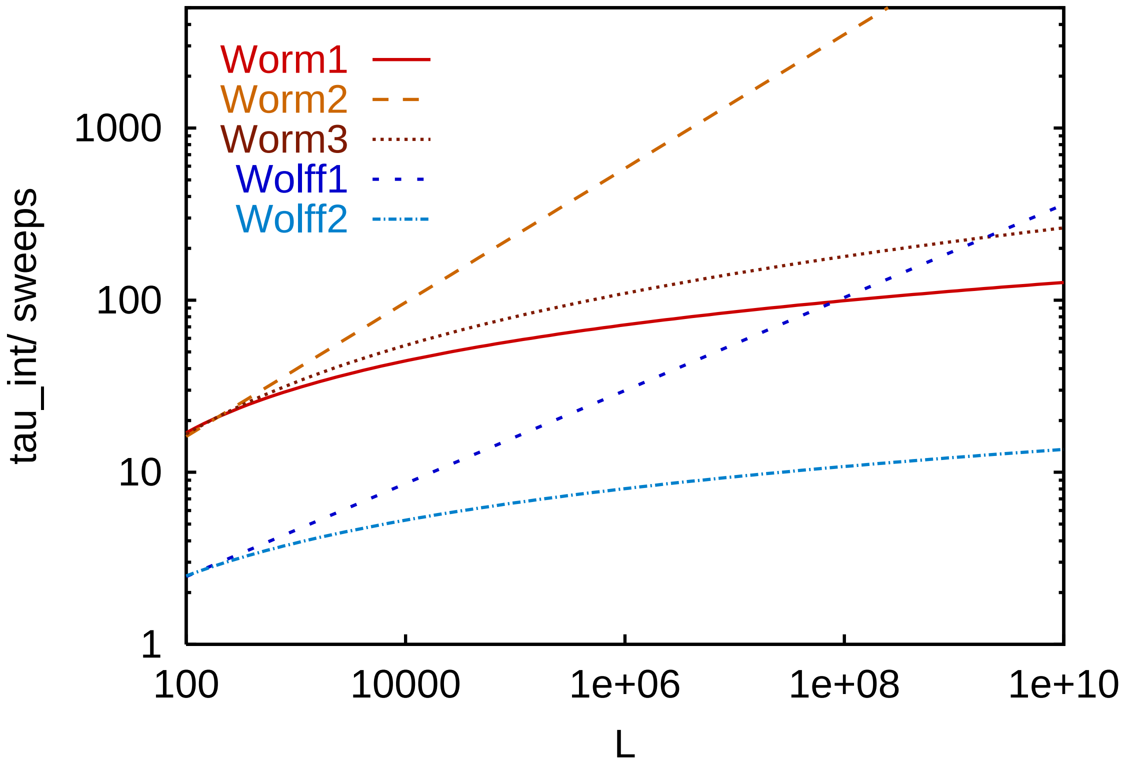

Figure 5.5: Extrapolated functions that were fitted to the integrated autocorrelation time results of energy of the Worm- and the Wolff algorithm. The Worm data is shown in table 5.2, the Wolff autocorrelation time data can be found in reference [22]. Worm1 $= a + b \cdot \ln(L)$, Worm2 $= a \cdot L^b$, Worm3 $\tau_{int} = a \cdot \ln(L)^b$. The Wolff data was fitted to exponential Wolff1$= 1.7 \cdot L^{0.25}$ and to logarithmic behavior Wolff2$= -1.1 + 1.45 \cdot \ln(L)$.

Correlation functions and susceptibility

While the autocorrelation function of energy showed a single mode decay, we found two modes in some autocorrelation functions of the susceptibility and correlation functions (see fig 5.6 and appendix A). Thus we can determine two different exponential autocorrelation times $\tau_{exp,1}$ and $\tau_{exp,2}$ corresponding to the decay of the modes (in this case τ_{int} and τ_{exp} do not agree). In figure 5.7 one can see that the autocor-

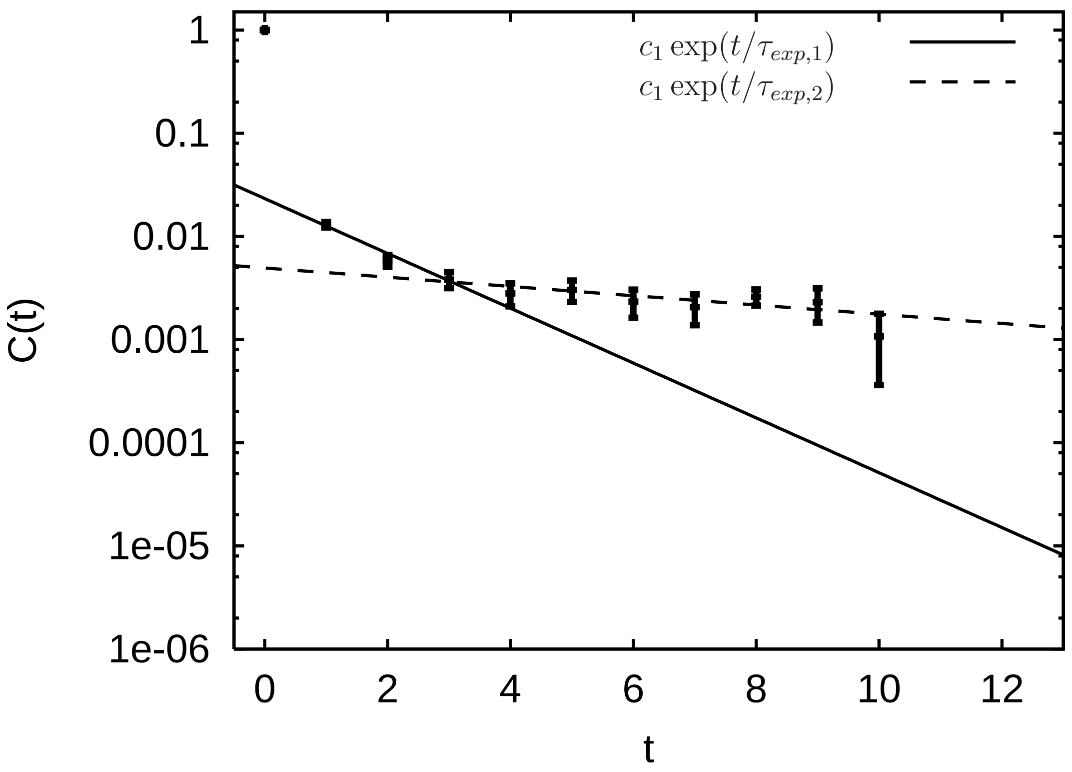

Figure 5.6: Correlation function of susceptibility of a 64×64 system.

relation function of the correlation function $G(L/2)$ and the susceptibility show very similar behavior and thus we will not treat them separately in the following.
The autocorrelation times $\tau_{exp,1}$ and $\tau_{exp,2}$ were determined via fits. In figure 5.8 we show the dependence of $\tau_{exp,1}$ (which characterizes the faster mode, see also fig. 5.6) on the lattice size. Its values are very small and barely increase with the lattice size L. The slower-mode autocorrelation times $\tau_{exp,2}(L)$ are clearly visible only at $L = 16, L = 32$ and $L = 64$ (see appendix A. They are in the range of 50 ± 30 sweeps. It is possible that this mode reflects the same mode of the autocorrelation function of energy or even a slower mode. Its contribution to the integrated autocorrelation time is however less than 1.

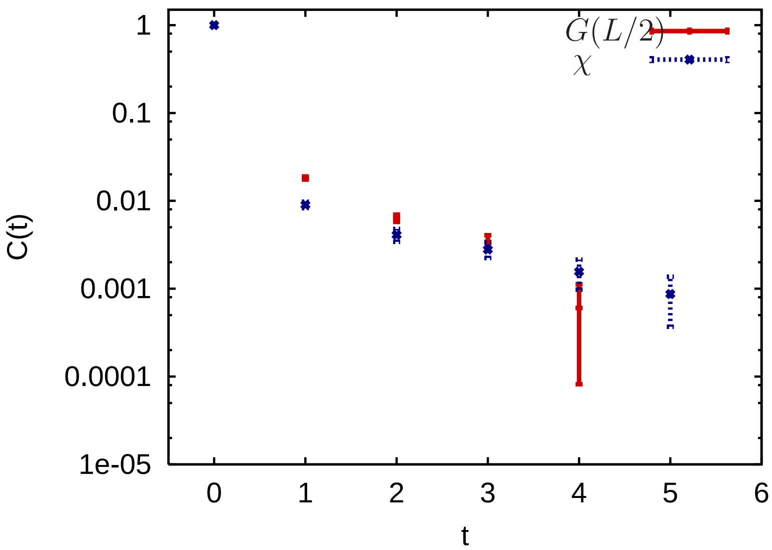

Figure 5.7: Correlation function of the susceptibility χ and the correlation function $G(L/2)$ of a 64×64 system at $T = T_c$.

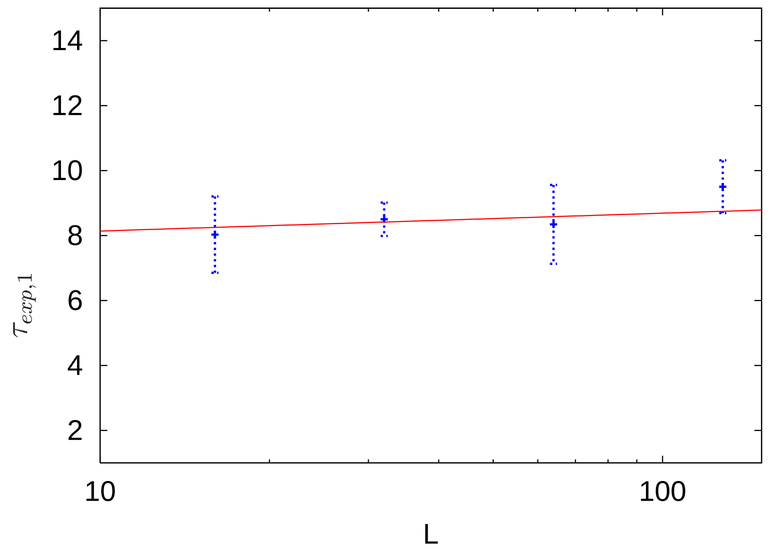

Figure 5.8: First mode exponential autocorrelation time $\tau_{exp,1}$ (see text) of susceptibility as a function of the lattice size. The fit function is given by $f(x) = a + b\ln(x)$ with $a = 6.1 \pm 0.8$ and $b = 0.7 \pm 0.2$

5.3.2 Autocorrelation vs. temperature for the 2d Ising model

We analyzed the dependence of τ_{int} of the energy on the temperature T. We measured this dependence for a 16×16 system. The result is depicted in fig.5.9. The maximum occurs at the critical temperature $T_c \approx 0.44$, as expected. There is no apparent problem with autocorrelation times at low or high temperature. Note that at very low temperatures, our algorithm becomes similar to a random walk since $\tanh(\beta J) \to 1$.

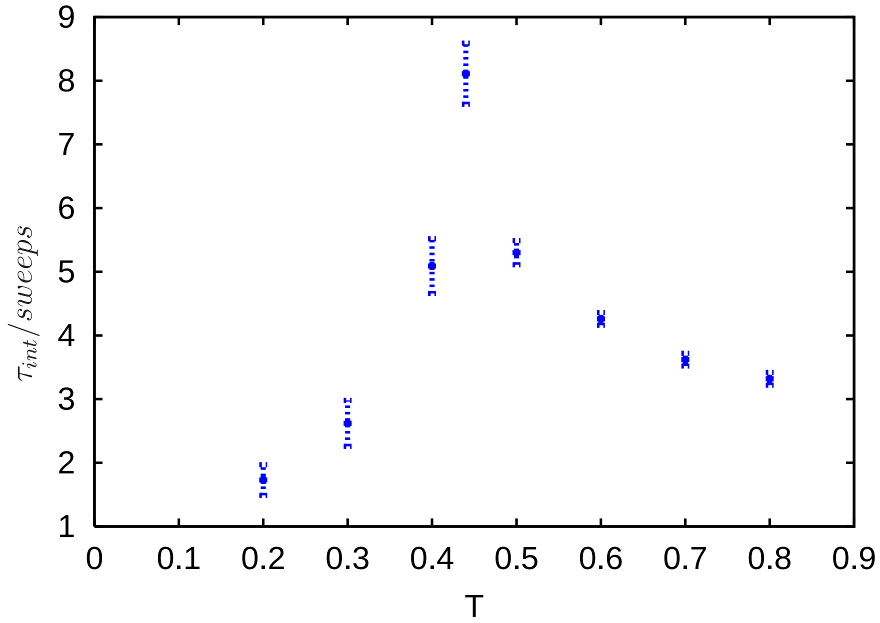

Figure 5.9: Dependence of the integrated autocorrelation time τ_{int} of the energy on the temperature T.

5.3.3 3d-Ising model

Energy

We analyzed the autocorrelation functions of the energy and found a single mode decay, i.e. $\tau_{int} = \tau_{exp}$, for all simulated lattices. Fig.5.10 shows the autocorrelation function of the energy for the biggest simulated lattice. The correlation functions for the smaller systems are depicted in appendix A.

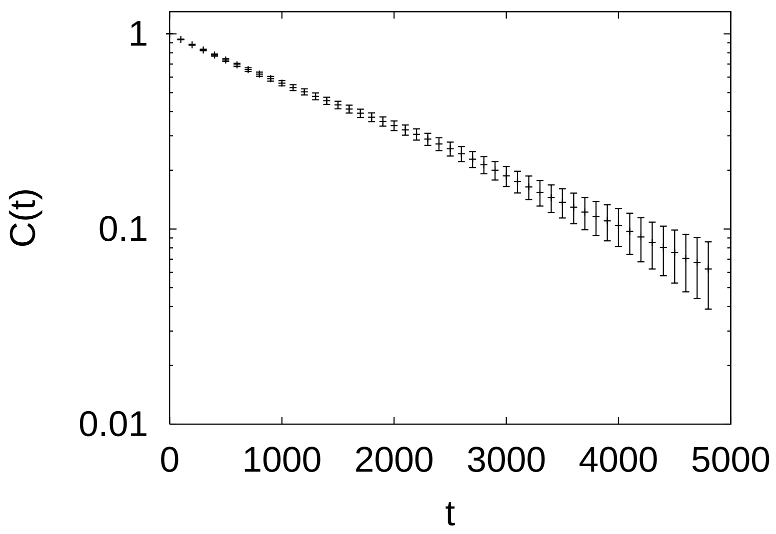

Figure 5.10: Autocorrelation function $C(t)$ of a $128 \times 128 \times 128$ system. The statistics of the Monte Carlo run can be extracted from table 5.4.

Again we made an autocorrelation time analysis in order to find out if the Worm Algorithm can compete with other cluster algorithms in three dimensions. The integrated autocorrelation times of the energy for the three dimensional Ising model are listed in table 5.4.

In order to describe the $\tau_{int}(L)$ behavior we fitted several functions to the data points obtained. Again we used an implementation of the nonlinear least-square Marquardt-Levenberg algorithm to execute the fits. The results are shown in fig. 5.11:

- In the first picture of fig.5.11 our data is fitted to $\tau = a + b \cdot \ln(L)$. The values obtained are $a = -2.1 \pm 1.6, b = 6.2 \pm 0.4$ and $\chi_r^2 = 0.32$.
- In the second picture of fig.5.11 the data points are assumed to show a behavior $\tau = a \cdot L^z$. We obtained $a = 7.59 \pm 0.09, z = 0.273 \pm 0.003$ and $\chi_r^2 = 0.01$.
- In the last picture of fig.5.11 we chose a $\tau = a + b \cdot \ln(L)^c$ behavior and obtained: $a = 14.0 \pm 0.6, b = 0.09 \pm 0.05, c = 3.2 \pm 0.03$, and $\chi_r^2 = 0.01$.

Thus any of these functions fits the data well. It is therefore possible that τ_{int} shows only a logarithmic dependence on the lattice size, while other cluster algorithms scale with a power law $\tau_{int} \sim L^z$ in three dimensions.

In fig. 5.12 we show a comparison of the autocorrelation time results of the energy of the Worm Algorithm and several cluster algorithms. We depicted $\tau_{int}(L)$ of the Worm Algorithm, the Swendsen-Wang Algorithm and the Wolff algorithm together with $\tau_{exp}(L)$ of the Joint Representation Cluster Algorithm. A comparison to the published exponential autocorrelation times of the Joint Representation Cluster algorithm (JCA) is possible since they agree with the integrated times within their statistical errors [17]. It turns out that from the linear lattice extension $L \approx 80$ on the Worm-Algorithm becomes more efficient than the Swendsen-Wang and the Joint Representation Algorithm, and thus is most suitable for simulating large systems.

However the Wolff algorithm shows smaller autocorrelation times in the range of our simulated lattice sizes and, as shown in figure 5.13, in the whole rangle of realistic system sizes.

Finally we show via a CPU-time comparison (see table 5.5) that the Wolff Algorithm remains more efficient in three dimensions. We found that, while the CPU time per sweep and volume $CPU/(M \cdot V)$ is roughly constant for the Wolff algorithm $CPU/(M \cdot V) \approx 0.5$, it varies stronger for the Worm Algorithm (see table 5.5). We expect that $CPU/(M \cdot V)$ becomes constant for sufficiently large systems.

L	τ_{int}/sweeps	$\Delta\tau_{int}$/sweeps	M/sweeps	t/s
4	11.13	0.29	30000	150
8	13.31	0.43	30000	1200
16	17.22	0.34	90000	30000
32	19.53	0.36	50000	130000
64	23.53	0.65	20000	420000
128	28.55	1.18	20000	3000000

Table 5.4: Integrated correlation time τ_{int} of the energy as function of the lattice size L of a three dimensional system. M denotes the number of measurements that contributed to the statistics and t the CPU-time of the MC-run. The calculations were done on 100000 kFLOPS machines.

$L = 8$	CPU/M[ms/sweep]	$CPU/(M \cdot V)$ [ms/sweep]
Worm	5	0.010
Wolff	2.5	0.005
$L = 16$		
Worm	35	0.009
Wolff	17	0.004
$L = 32$		
Worm	245	0.007
Wolff	193	0.006

Table 5.5: Amount of CPU-time per sweep calculated for the Worm- and and the Wolff Algorithm (data taken on a 3d lattice at T_c). For the Wolff-Algorithm a sweep is defined per: one sweep = L^d elementary steps. M denotes the number of sweeps, V is the volume of the system. The right column us thus the time needed for 1 elementary step in the Wolff algorithm, resp. d elementary steps in the Worm.

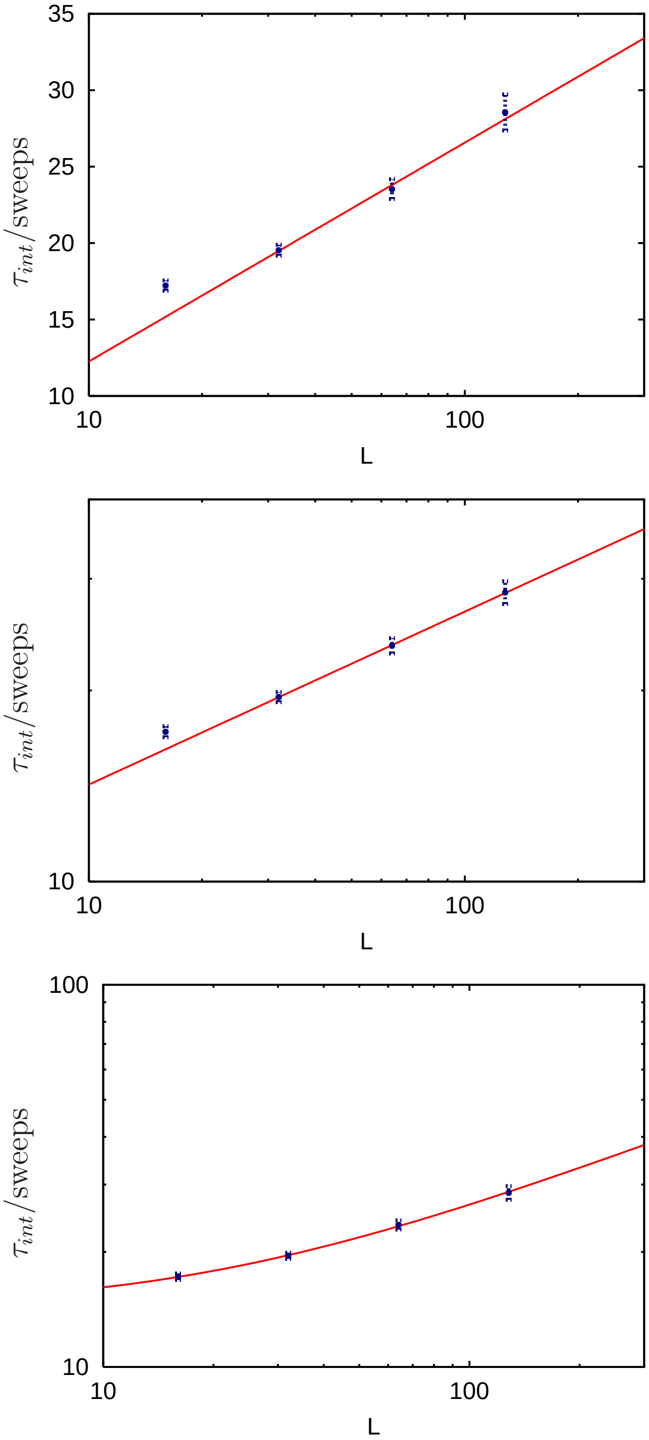

Figure 5.11: Dependence of the integrated autocorrelation time τ_{int} on the lattice size L in three dimensions. In the first picture the data is fitted to $f(x) = -2 + 6\log(x)$ in the second picture to $f(x) = 7.6x^{0.27}$ and in the last picture to $f(x) = 14 + 0.1\log(x)^{3.2}$.

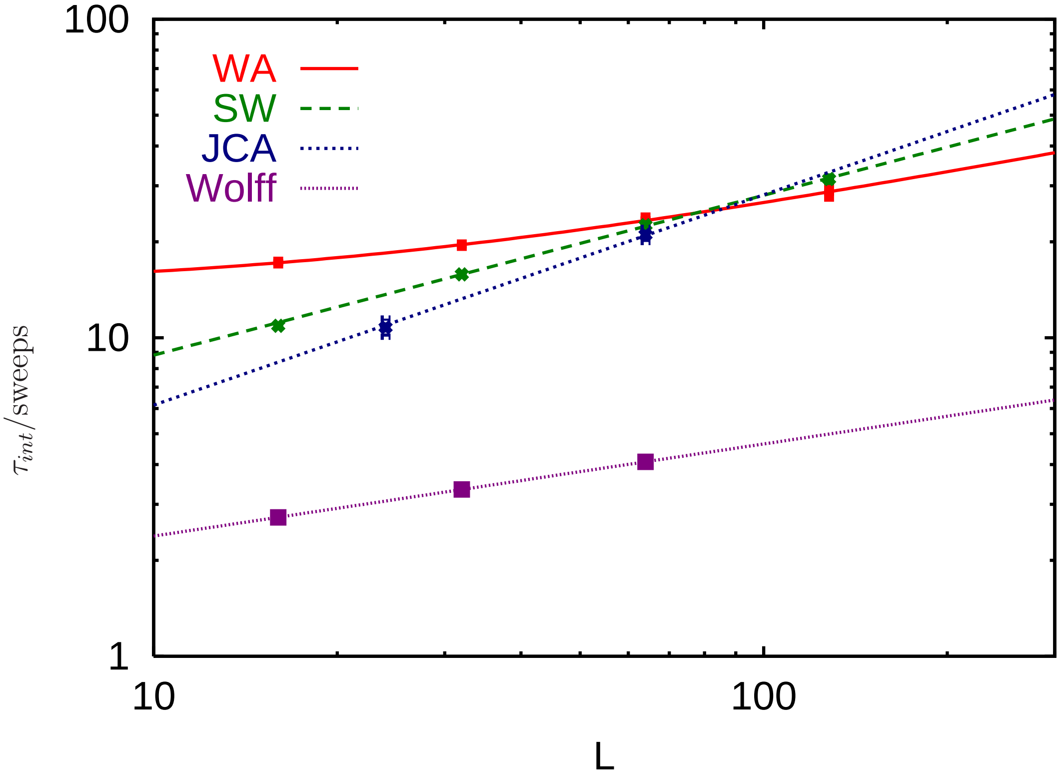

Figure 5.12: Comparison of the integrated-autocorrelation times of the energy of the Worm-Algorithm with the Swendsen Wang- and the Wolff Algorithm and the exponential autocorrelation time of the Joint Representation Algorithm in three dimensions. The SW-data and the Wolff-data were obtained from Prof. Jian-Sheng Wang. The data from the Joint Representation Algorithm can be found in reference [17].

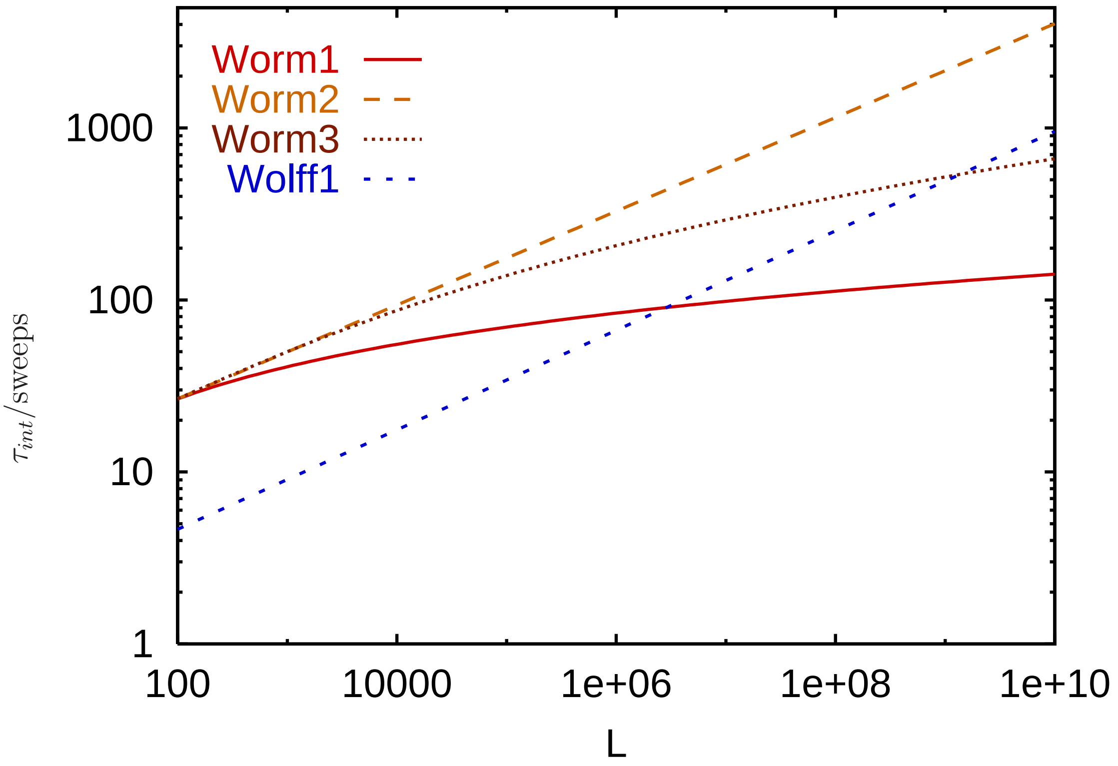

Figure 5.13: Extrapolated functions that were fitted to the integrated autocorrelation time results of the energy of the Worm- and the Wolff Algorithm. The Worm data is shown in table 5.2, the Wolff autocorrelation time was obtained from Prof. Jian-Sheng Wang. Worm1 $= a + b \cdot \ln(x)$, Worm2 $= a \cdot x^b$, Worm3 $= a \cdot \ln(x)^b$. The Wolff data was fitted to exponential Wolff1 $= 1.22 \cdot x^{0.29}$ behavior .

Correlation functions and suceptibility

We found also in three dimensions, that the decay of the autocorrelation function of the susceptibility χ and of the correlation functions $G(L/2)$ consists of two modes. An example is shown in fig.5.14, further functions can be found in the appendix A.

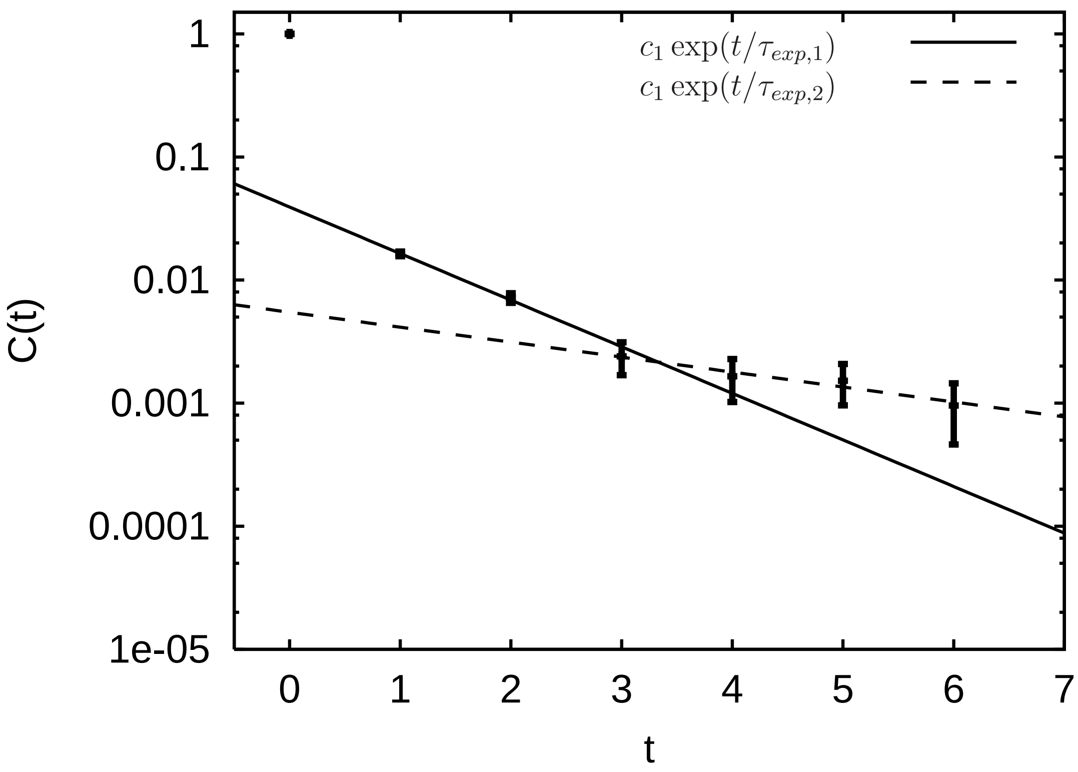

Figure 5.14: Correlation function of the susceptibility for a $16 \times 16 \times 16$ system.

The behavior of the autocorrelation functions of the susceptibility and the correlation function $G(L/2)$ is nearly identical (see figure 5.15) and thus those functions will not be treated separately in the following.

Again we analyzed the dependence of the two autocorrelation times $\tau_{exp,1}, \tau_{exp,2}$, corresponding to the two mode decay of the autocorrelation functions. The autocorrelation times $\tau_{exp,1}, \tau_{exp,2}$ were determined via fits. The "faster mode" autocorrelation time $\tau_{epx,1}$ (for explanation see fig.5.14) was located to be of the order of $1.5 - 2.5$ sweeps for lattice sizes from $L = 8$ up to $L = 128$. The "slower mode" autocorrelation time $\tau_{epx,2}$ showed values $\tau_{epx,2} < 10$ sweeps for our lattice sizes, and thus should not be a problem.

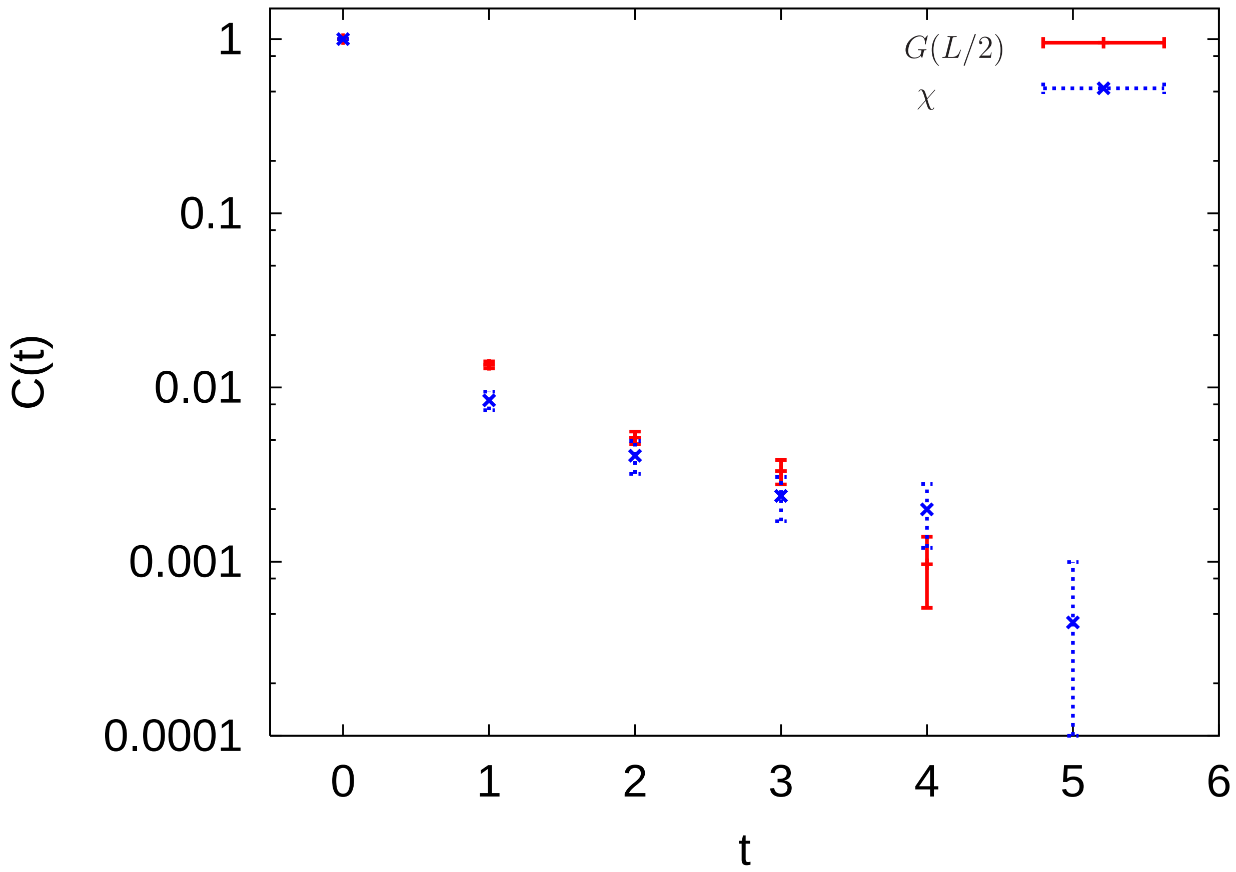

Figure 5.15: Autocorrelation function of the susceptibility χ and the correlation function $G(L/2)$ of a $64 \times 64 \times 64$ system.

5.4 Histograms

In order to get more information about the contribution of the sampled values of susceptibility we compared histograms produced by the Worm-Algorithm with those of a simple random walk. We found that the distributions, produced via the Worm Algorithm, do qualitatively not show extraordinary deviations from those of the random walk and thus should not cause large autocorrelation times. The histograms are shown in fig. 5.16 and fig. 5.17.

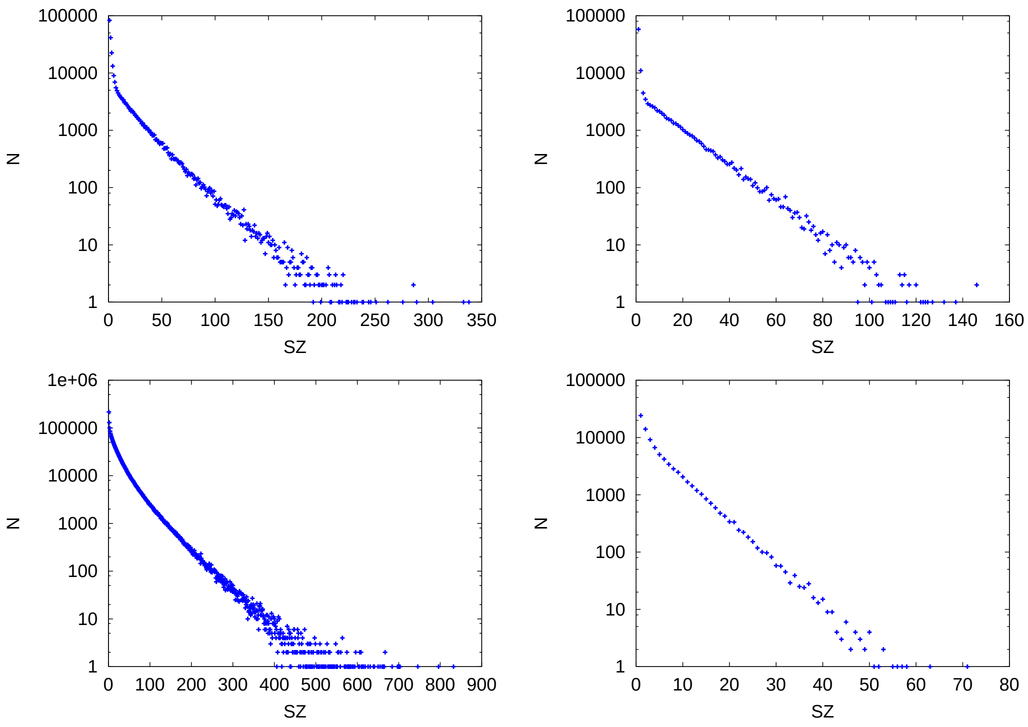

Figure 5.16: Histograms of the sampled values of susceptibility of systems with $L = 16$ and dim=2 (top), dim=3 (bottom) in semi-logarithmic scaling. The left pictures result from sampling the Worm Algorithm, in the right pictures we sampled a random walk. N denotes the number of entries for a certain χ value.

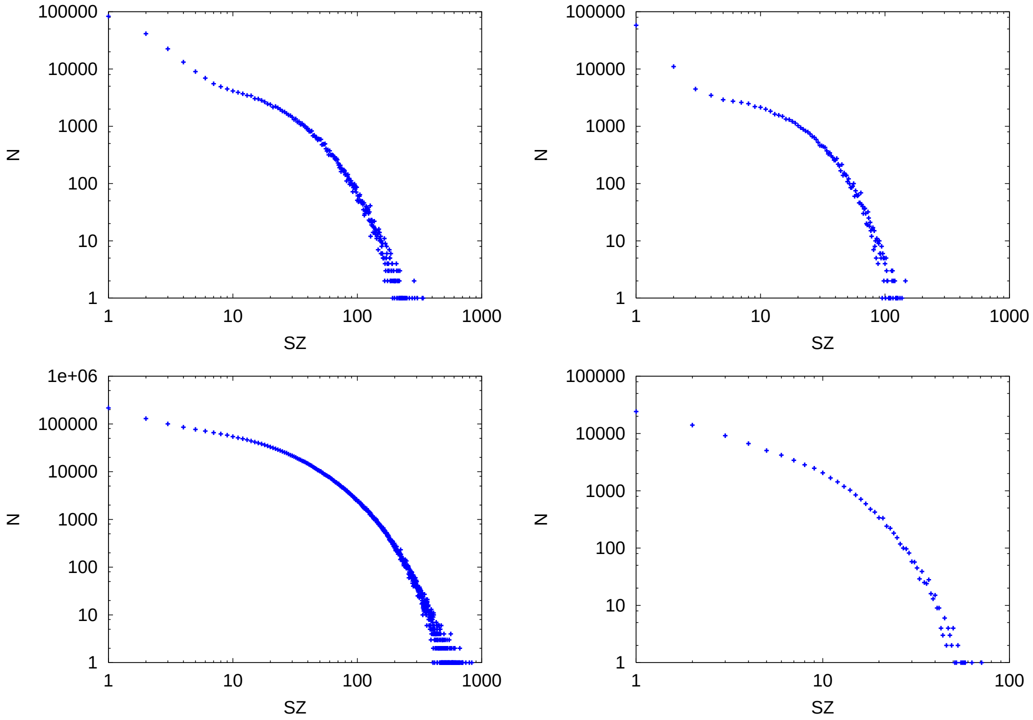

Figure 5.17: Histograms of the sampled values of susceptibility of systems with $L = 16$ and dim=2 (top), dim=3 (bottom) in double logarithmic scaling. The left pictures result from sampling the Worm Algorithm, in the right pictures we sampled a random walk. N denotes the number of entries for a certain χ value.

5.5 Summary

We now sum up the results of the preceding chapters:

- The energy showed the biggest integrated autocorrelation times τ_{int} of the physical quantities considered.
- The integrated autocorrelation times τ_{int} of the spatial correlation function $G(L/2)$ and the susceptibility are very small (< 1 sweep, see table 5.1).
- We found that in $d = 2$ and $d = 3$ a logarithmic scaling of τ_{int} of the energy is compatible with our data obtained, i.e. $\tau_{int} \sim \ln L$.
- A comparison with cluster algorithms showed that from $d = 3, L \approx 80$ on the Worm Algorithm becomes efficient than the Swendsen Wang and the Joint Representation algorithm. However, the Wolff algorithm showed smaller integrated

autocorrelation times in the whole range of realistic lattice sizes in $d = 2$ and $d = 3$.

Chapter 6

Including magnetic fields

In chapter 3.7 we introduced cluster-methods that effectively reduce critical slowing down for a broad class of spin models. However we did not include external magnetic fields in our algorithms. In this chapter we will describe several methods to simulate the Ising model with nonzero fields. In the following chapter we will extend our Worm-Algorithm to including magnetic fields and finally we will introduce a special application, the so-called “capillary condensation”

6.1 The Swendsen Wang Algorithm with Magnetic field

The Swendsen Wang Algorithm with zero field was described in section 3.7.1. When there is a magnetic field, two versions of the algorithm exist [23]: with or without a ghost spin.

We first state the algorithm without a ghost spin. The modification to the algorithm with zero field is in assigning spins to clusters (step(3)). Each cluster is like a noninteracting Ising spin experiencing a field of strength $H \cdot \eta_\alpha$, where η_α is the size of cluster α. Spin s_α for the cluster α is sampled proportional to $e^{H\eta_\alpha s_\alpha}$. A disadvantage of this algorithm is that the clusters constructed are the same as for zero field and thus do not reflect the correct physics. At large field, most spins point in the direction of the field and clusters will rarely be flipped. This leads to exponentional increase of the autocorrelation time.

The second version utilizes *ghost spin*. The modification to the zero field case is in generating bonds (see also [9]). We formally write the field term $H\sum_i s_i$ as $|H|\sum_i s_i s_g$ where s_g is the so-called ghost spin which takes the value $\mathrm{sign}(H)$ and does not change. Now the field term looks just the same as nearest-neighbor interactions. It is treated in a similar fashion: bonds are put between the ghost spin and every lattice spin with probability $p_g = 1 - e^{-\beta|H|}$ if $s_i s_g = 1$. After generating the bond configuration we carry out steps (2) and (3) (described in chapter 3.7.1), except we do not flip the cluster connected with the ghost spin.

However, if $H = h/k_bT$ is larger than order unity, then p_g is of order unity and this implies that huge clusters containing a large fraction of the entire simulation volume will be created most of the time but will not be flipped. It is clear that under such circumstances the algorithm will be very inefficient, again leading to exponatially large autocorrelation times. As a consequence, one needs $h/k_bT \lesssim 1$.

6.2 Joint Representation Cluster algorithm including a magnetic field

We introduced the Joint Representation Cluster algorithm in section 3.7.5. Magnetic fields H are immediately included in this formalism, by adding a fictitious ghost spin $s_0 \equiv 1$ to the lattice, to which all other spins couple with strength $J_0 := H$ (we follow reference [3]). I.e. we write

$$e^{\beta H \sum_i h_i s_i} = e^{\beta J_0 \sum_{\langle i,j \rangle} s_i s_0} \tag{6.1}$$

Everything else remains the same, including the Monte-Carlo procedure. After summing over spin degrees of freedom there has to be an even number of bond variables $b_{i,0} = 1$, with the constraint $\partial\{b_{i,j}\} = \{b_{i,0}\}$. Thus the additional "ghost-bond"-variables $b_{i,0} = 1$ and $f_{i,0} = 1$ can also be viewed as *sources* of the high-temperature representation (respectively the combined representation) with fields.

If one sums over sources $b_{i,0}$, one recovers the Swendsen Wang algorithm with ghost spin, which performs well only in a very small magnetic field (see chapter 6.1). However, in the JCA method there is no freezing of f clusters, since spins do not appear in eq. 1.20 and sources f_{i0} and b_{i0} can fluctuate even in a strong field. It might therefore perform better in strong fields. This has not been tested yet.

6.3 Two Replica Cluster Algorithm

The Two Replica Cluster algorithm is similar to the Swendsen Wang Algorithm except that clusters are grown in a two replica system rather than in a single replica and only active clusters, where there is no net field, are flipped.

The idea of this algorithm is as follows [24]: Two independent replicas of the system are simulated simultaneously. Each site of the lattice is therefore associated with two spins, one from each replica, and can be in one of four spin states, $(++), (--), (+-)$ and $(-+)$. Sites where the replicas disagree,$(+-)$ and $(-+)$ are called *active* sites. Clusters of active sites are constructed and flipped. Allowed cluster flips interchange $(+-)$ with $(-+)$. Additional updating is applied independently to each replica to ensure ergodicity . Since there is no net field on active clusters these flip freely.

We start from the Ising model Hamiltonian[1]:

$$\mathcal{H} = \sum_{\langle ij \rangle} s_i s_j - H \sum_i s_i \tag{6.2}$$

We now define a joint distribution of two sets of Ising spin variables, $\{s_i\}$ and $\{t_i\}$ and a bond variable $\{\eta_{ij}\}$. The bond variable is defined for each bond $\langle i,j \rangle$ and takes values 0 and 1. We say that $\langle i,j \rangle$ is *occupied* if $\eta_{ij} = 1$. The statistical weight $W(s,t,\eta)$ for the joint distribution is

$$W(s,t,\eta) = e^{-\beta \sum_{\langle i,j \rangle} s_i t_i s_j t_j + \beta H \sum_i (s_i + t_i)} \Delta(s,t,\eta) B_p(\eta) \tag{6.3}$$

where B is the Bernoulli factor

$$B_p(\eta) = p^{N_b} (1-p)^{N_{nn} - N_b} \tag{6.4}$$

N_b is the number of occupied bonds and N_{nn} is the number of nearest-neighbor pairs on the lattice. The factor Δ is defined via : if for every nearest-neighbor pair $\langle i,j \rangle$ such that $\eta_{i,j} = 1$ the spins agree in both replicas ($s_i = s_j$ and $t_i = t_j$) then $\Delta(s,t,\eta) = 1$; otherwise $\Delta(s,t,\eta) = 0$. One can verify that integrating $\Delta(s,t,\eta)$ over η yields the statistical weight for two independent Ising models in the same field

$$e^{-\beta \mathcal{H}(\{s\}) - \beta \mathcal{H}(\{t\})} = \text{const} \sum_{\{\eta\}} W(s,t,\eta) \tag{6.5}$$

if the identification is made that $p = 1 - e^{-4\beta}$.

The Two Replica Cluster algorithm simulates two independent Ising models, s and t, on the same lattice and with the same field . Bonds $\langle i,j \rangle$ connecting like spins in both replicas ($s_i = s_j$ and $t_i = t_j$) are called *satisfied* bonds. The algorithm is:

1. Satisfied bonds connecting active sites are occupied with probability $p = 1 - e^{-4\beta}$.

2. Clusters of active sites connected by occupied bonds (including single active sites) are identified. The k^{th} cluster is independently assigned a spin value, $\sigma_k = \pm 1$ with probability 1/2. If site i is in cluster k, then the new spin values are $s_i = \sigma_k$ and $\tau_i = -\sigma_k$. In this way all active sites are updated.

[1] we follow [24]

3. Each replica is independently updated in a way that preserves detailed balance and insures ergodicity. This completes one Monte Carlo step.

Without Step 3, the algorithm is not ergodic since the product $s_i t_i$ is locally conserved. Step 3 can be implemented in many ways. For example, each replica can be separately updated with local spin flips using the Metropolis algorithm. The validity of the algorithm is proved by showing that it is ergodic and that the joint distribution, $W(s, t, \eta)$, defined in eq. 6.3, is the stationary distribution of the algorithm. This can be found in reference [24].

For Ising models with staggered field this algorithm showed a critical dynamic exponent less than 0.5 in two dimensions [24].

6.4 Geometric Cluster algorithm in the fixed magnetization ensemble

In chapter 3.7.4 we introduced the Geometric Cluster algorithm. While the efficiency of this algorithm can be comparable with the Wolff Algorithm (see chapter 3.7.3), its applicability is different because it is based on (e.g.) geometric symmetries of a system and therefore in principle available for many more systems. In order to simulate a system with a magnetic field, one also introduces a second replica, with opposite field , and employs the symmetry of switching spins between the replicas. Then the algorithm samples the *constant magnetization ensemble*. It resembles the Two Replica algorithm of chapter 6.3. The Geometric Cluster algorithm in the fixed magnetization ensemble was introduced by H.W.J. Blöte and J.R. Heringa [25]. We could not find any comments on the performance of this algorithm.

Chapter 7

Worm Algorithm including a magnetic field

Within the framework of this diploma-thesis we extended the Worm Algorithm for Ising models, introduced in chapter 4, to including magnetic fields. In order to develop the algorithm, we make use of the "ghostspin"- formalism and rewrite the partition function with non-zero field H_i:

$$Z = \sum_{\{s_i\}} e^{\beta J \sum_{\langle i,j \rangle} s_i s_j + \beta \sum_{\langle i,0 \rangle} H_i s_i s_0} \tag{7.1}$$

with $s_0 \equiv 1$ and $\sum_{\langle i,0 \rangle} = \sum_i$ since the ghostspin s_0 is formally nearest neighbor to every lattice site i. Thus we reduced the partition function including external fields to the same formalism as in eq. 1.11 with an additional nearest neighbor pair $\langle i, 0 \rangle$ with coupling constant $J_{0,1} = H_i$. The high-temperature expansion is analogous to eq. 1.12-1.13:

$$
\begin{aligned}
Z &= \sum_{\{s_i\}} \left[\prod_{\langle i,j \rangle} e^{\beta J s_i s_j} \prod_i e^{\beta H_i s_i s_0} \right] \\
&= \cosh(\beta J)^{N_{nn}} \left[\prod_i \cosh(\beta H_i) \right] \sum_{\{s_i\}} \left[\prod_{\langle i,j \rangle} (1 + v s_i s_j) \prod_i (1 + w_i s_i s_0) \right] \\
&= \cosh(\beta J)^{N_{nn}} \left[\prod_i \cosh(\beta H_i) \right] \sum_{\{s_i\}} \Bigg[\Big(1 + v \sum_{\langle i,j \rangle} s_i s_j + v^2 \sum_{\substack{\langle i,j \rangle \\ \langle k,l \rangle}} s_i s_j s_k s_l + ... \Big) \\
&\qquad \cdot \Big(1 + \sum_{\langle i,0 \rangle} w_i s_i s_0 + \sum_{\substack{\langle i,0 \rangle \\ \langle j,0 \rangle}} w_i w_j s_i s_j s_0^2 + ... \Big) \Bigg] \\
&= \cosh(\beta J)^{N_{nn}} 2^N \left[\prod_i \cosh(\beta H_i \right] \sum_{\substack{\{b\},\{\tilde{b}\} \\ \partial b = \partial \tilde{b}}} \left[v^{n_b} \prod_i w_i^{n_{\tilde{b}}(i)} \right]
\end{aligned} \tag{7.2}
$$

with $v = \tanh(\beta J_i), w_i = \tanh(\beta H_i), n_b$ denotes the number of bonds with $b_{i,j} = 1$, $n_{\tilde{b}}(i)$ denotes the number of ghost bonds at lattice site i ($n_{\tilde{b}}(i) = 0, 1$) and $\{\tilde{b}\}$ denotes the ghost-bond configurations. In the last step we made use of eq. 1.14 which leads us again to the closed configuration constraint: only configurations $\{b\}$ whose boundary ∂b coincide with the boundary $\partial\tilde{b}$ of the ghostbond configuration $\{\tilde{b}\}$, i.e. $\partial b = \partial\tilde{b}$, contribute to the partition function.

Thus the high-temperature expansion including external fields differs from eq. 1.15 only, that we now have, in addition to bonds contributing $\tanh(\beta J_i)$ to the partition function, "ghost-bonds" that contribute $\tanh(\beta H_i)$.

If there is a homogeneous external field $H_i \equiv H$ the partition function can be written as:

$$
Z = \cosh(\beta J)^{N_{nn}} \cosh(\beta H)^N 2^N \sum_{\substack{\{b\},\{\tilde{b}\} \\ \partial b = \partial \tilde{b}}} v^{n_b} \cdot w^{n_{\tilde{b}}} \tag{7.3}
$$

with $w = \tanh(\beta H)$ and $n_{\tilde{b}} = \sum_i n_{\tilde{b}}(i)$.

We mention that if the sign of H_i is not constant, than also the sign of the configuration weight is not constant. This can lead to a "sign-problem" for the simulation due to the demand of positive configuration weights. Thus our algorithm is restricted to simulate ferromagnetic systems in a positive external field (or antiferromagnet on a bipartite lattice in a staggered field, respectively).

7.1 The algorithm

The algorithm including a non-zero field H_i differs from the old algorithm, (described in chapter 4.1) in the fact that we now set/erase ghost bonds, which contribute $w = \tanh\beta H_i$ to the partition function, in addition to our "normal" bonds which contribute $v = \tanh\beta J$. We consider the case of a homogeneous external field $H_i \equiv H$. The weight of a certain configuration $\{b\}, \{\tilde{b}\}$, consisting of bonds b and ghost-bonds $\tilde{b}$ is given by the partition function:

$$W_Z(\{\tilde{b}\}, \{\tilde{b}\}) = v^{n_b} w^{n_{\tilde{b}}} = (\tanh\beta J)^{n_b} (\tanh\beta H)^{n_{\tilde{b}}} \tag{7.4}$$

The algorithm is now:

1. We start at a random position i_{start} and iterate the following steps:
2. Choose a random number $x \in [0, 1)$
   (a) If $x < 0.5$ do one elementary step of the field-less Worm Algorithm (see chapter 4.1).
   (b) If $x \geq 0.5$: If $i_1 = i_2$ transfer the current values of the observables $E, \chi, G(i)$ and M to the statistics routine.
   Do the *heatbath-algorithm for ghost spins* described below. Add the contribution of the gained configuration to the susceptibility and to the corresponding correlation function. One *elementary step* has been carried out.
3. Go to step 2.

Heatbath algorithm for ghost-spins

Using the Metropolis algorithm for the ghost-spin updates would lead to acceptance rates that are proportional to $1/N$ and thus become very small for large lattices [1]. Thus we decided to use a heatbath-algorithm for the ghost-spin updates sampling two steps at once: One step, where the worm end is shifted to the ghostspin s_0 and one step that shifts the worm end back to the lattice. If we have the possibility to set/erase ghost spins we proceed in the following way:

Let the worm end to be updated be at site i, i.e. $i_1 = i$, and let N denote the number of lattice sites.

[1] The ghost spin is nearest neighbor of all lattice sites, and thus the probability to shift the "worm-end" from the ghostspin-site back to a special lattice site i is proportional to $1/N$. Since the microreversibility condition has to be satisfied the acceptance probability for setting a ghost-bond is proportional to $1/N$.

1. If there is no ghostbond connecting the lattice site i with the ghost-spin site i_0, i.e $\tilde{b}_{i,0} = 0$, the following updates are possible:

   - Set a ghostbond $b_{i,0} = 1$. This increases the configuration weight by a factor$w$.Move back to one of the $(N - n_{\tilde{b}} - 1)$ lattice sites where $b_{j,0} = 0$ setting a second ghost-bond $b_{0,j} = 1$.This again increases the configuration weight by a factor w . There are $(N - n_{\tilde{b}} - 1)$ possibilities for this update.
   - Set a ghostbond $b_{i,0} = 1$. This increases the configuration weight by a factor w. Move back to one of the $(n_{\tilde{b}} + 1)$ lattice sites where $b_{j,0} = 1$ erasing a ghost-bond $b_{0,j} = 0$. This reduces the configuration weight by a factor $1/w$. There are $(n_{\tilde{b}} + 1)$ possibilities for this update.

   Then the total weight of all configurations which can be reached is given by:

   $$W_{total}(n_{\tilde{b}}|\tilde{b}_{i,0} = 0) = [(N - n_{\tilde{b}} - 1)w^2 + (n_{\tilde{b}} + 1)] \cdot W_{old} \tag{7.5}$$

   where W_{old} denotes the weight and $n_{\tilde{b}}$ the number of ghost-bonds of the configuration before the update. We proceed as follows:

   (a) With probability

   $$p(i \to j, \tilde{b}_{i,0} = 0 \to 1, \tilde{b}_{0,j} = 0 \to 1) = \frac{(N - n_{\tilde{b}} - 1)w^2 \cdot W_{old}}{W_{total}} \tag{7.6}$$

   we shift i_1 from i to lattice site j setting ghost-bonds $\tilde{b}_{i,0} = 1, \tilde{b}_{0,j} = 1$ (note that j must be chosen randomly among those lattice sites k where $\tilde{b}_{0,k} = 0$ before this update). After carrying out this step the number of ghost-bonds has increased: $n_{\tilde{b}} \to n_{\tilde{b}} + 2$

   (b) With probability

   $$p(i \to j, \tilde{b}_{i,0} = 0 \to 1, \tilde{b}_{0,j} = 1 \to 0) = \frac{(n_{\tilde{b}} + 1) \cdot W_{old}}{W_{total}} \tag{7.7}$$

   we shift i_1 from i to lattice site j setting a ghost-bond $\tilde{b}_{i,0} = 1$ and erasing a ghost-bond $\tilde{b}_{0,j,} = 0$ (in this case j must be chosen randomly among those lattice sites k where $\tilde{b}_{0,k} = 1$ before this update). The number of ghost-bonds remains the same $n_{\tilde{b}} \to n_{\tilde{b}}$.

2. If there is a ghost-bond connecting the lattice site i with the ghost-spin i_0, i.e $\tilde{b}_{i,0} = 1$, the total weight of all configurations which can be reached is given by

   $$W_{total}(n_{\tilde{b}}|\tilde{b}_{i,0} = 1) = [(N - n_{\tilde{b}} + 1) + \frac{n_{\tilde{b}} - 1}{w^2}] \cdot W_{old} \tag{7.8}$$

where W_{old} denotes the weight of the configuration and $n_{\tilde{b}}$ the number of ghost-bonds before the update.

(a) With probability

$$p(i \to j, \tilde{b}_{i,1} = 0 \to 1, \tilde{b}_{0,j} = 0 \to 1) = \frac{(N - n_{\tilde{b}} + 1) \cdot W_{old}}{W_{total}} \tag{7.9}$$

we shift i_1 from i to lattice site j erasing a ghost-bond $\tilde{b}_{i,0} = 0$ and setting a ghost-bond $\tilde{b}_{0,j} = 1$ (j must be chosen randomly among those lattice sites k where $\tilde{b}_{0,k} = 0$ before this update). $n_{\tilde{b}}$ remains unchanged.

(b) With probability

$$p(i \to j, \tilde{b}_{i,0} = 1 \to 0, \tilde{b}_{0,j} = 1 \to 0) = \frac{(n_{\tilde{b}} - 1) \cdot W_{old}}{w^2 W_{total}} \tag{7.10}$$

we shift i_1 from i to lattice site j erasing two ghost-bonds $\tilde{b}_{i,0} = 0$ and $\tilde{b}_{0,j,} = 0$ (j must be chosen randomly among those k lattice sites where $\tilde{b}_{0,k} = 1$ before this update). $n_{\tilde{b}}$ decreases to $n_{\tilde{b}} - 2$.

7.1.1 Detailed balance

In section 4.1.1 we showed that the Worm Algorithm without magnetic field satisfies the detailed-balance condition. In our extended Worm Algorithm we now choose with probability 1/2 to update the ghost-bond configuration via the heatbath algorithm introduced in the previous section. It remains to be shown that our heatbath-probabilities defined in eq. 7.6-7.7 and 7.9-7.10 satisfy the micro-reversibility-condition. We therefore distinguish the following cases :

- **Forward**: We start from the left configuration C_1 shown in fig 7.1: $i_1 = i$, $\tilde{b}_{i0} = 0$, $\tilde{b}_{0j} = 0$, number of ghost-bonds: $n_{\tilde{b}}$, with weight $W(C_1) = const \cdot w^{n_{\tilde{b}}}$ and move with probability

  $$p(C_1 \to C_2) = p(j|i, \{\tilde{b}_{ij}\}) \cdot p(i \to j, \tilde{b}_{i,0} = 0 \to 1, \tilde{b}_{0,j} = 0 \to 1)$$

  (where $p(j|i, \{\tilde{b}_{ij}\}) = 1/(N - n_{\tilde{b}} - 1))$ to the lattice site j while increasing the number of ghost-bonds $n_{\tilde{b}} \to n_{\tilde{b}} + 2$.

  Backward: We start from the right configuration C_2 shown in fig 7.1: $i_1 = j$, $\tilde{b}_{j0} = 1$, $\tilde{b}_{0i} = 1$, number of ghost-bonds: $n_{\tilde{b}} + 2$, with weight $W(C_2) = const \cdot w^{n_{\tilde{b}}+2}$ and move with probability

  $$p(C_2 \to C_1) = p(i|j, \{\tilde{b}_{ij}\}) \cdot p(j \to i, \tilde{b}_{j,0} = 1 \to 0, \tilde{b}_{0,i} = 1 \to 0)$$

(where $p(i|j,\{\tilde{b}_{ij}\}) = 1/(n_{\tilde{b}}+1)$) back to the lattice site i while decreasing the number of ghostbonds $n_{\tilde{b}}+2 \rightarrow n_{\tilde{b}}$.

The detailed balance condition (eq. 3.13) is satisfied with the transition probabilities 7.6, 7.10 and weights 7.4, 7.5, 7.8:

$$W(C_1)\cdot p(C_1 \rightarrow C_2) = \frac{const\cdot w^{n_{\tilde{b}}}}{(N-n_{\tilde{b}}-1)}\cdot\frac{(N-n_{\tilde{b}}-1)w^2}{[(N-n_{\tilde{b}}-1)w^2+(n_{\tilde{b}}+1)]} \tag{7.11}$$

$$W(C_2)\cdot p(C_2 \rightarrow C_1) = \frac{const\cdot w^{n_{\tilde{b}}+2}}{(n_{\tilde{b}}+1)}\cdot\frac{(n_{\tilde{b}}+1)}{w^2\cdot[(N-n_{\tilde{b}}-1)+\frac{n_{\tilde{b}}+1}{w^2}]}. \tag{7.12}$$

- **Forward**: We start from the left configuration C_3 shown in fig 7.2: $i_1 = i$, $\tilde{b}_{i0} = 0.\tilde{b}_{j0} = 1$,number of ghostbonds: $n_{\tilde{b}}$, with weight $W_(C_1) = const\cdot w^{n_{\tilde{b}}}$ and move with probability

$$p(C_3 \rightarrow C_4) = p(j|i,\{\tilde{b}_{ij}\})\cdot p(i \rightarrow j, \tilde{b}_{i,0} = 0 \rightarrow 1, \tilde{b}_{0,j} = 1 \rightarrow 0)$$

(where $p(j|i,\{\tilde{b}_{ij}\}) = 1/(n_{\tilde{b}}+1)$) to the lattice site j. The number of ghostbonds remains the same $n_{\tilde{b}} \rightarrow n_{\tilde{b}}$ since we have erased and sequentially set one ghost-bond during this update.

Backward: Taking a look at configuration C_3, depicted in the right picture of fig. 7.2, it becomes clear that we have the identical situation as described above. Thus detailed-balance is satisfied trivially. Using the transition probability 7.6, and weights 7.4, 7.5 we end up with:

$$W(C_3)\cdot p(C_3 \rightarrow C_4) = \frac{const\cdot w^{n_{\tilde{b}}}}{(n_{\tilde{b}}+1)}\cdot\frac{(n_{\tilde{b}}+1)}{[(N-n_{\tilde{b}}-1)w^2+(n_{\tilde{b}}+1)]} \tag{7.13}$$

$$W(C_4)\cdot p(C_4 \rightarrow C_3) = \frac{const\cdot w^{n_{\tilde{b}}}}{(n_{\tilde{b}}+1)}\cdot\frac{(n_{\tilde{b}}+1)}{[(N-n_{\tilde{b}}-1)w^2+(n_{\tilde{b}}+1)]} \tag{7.14}$$

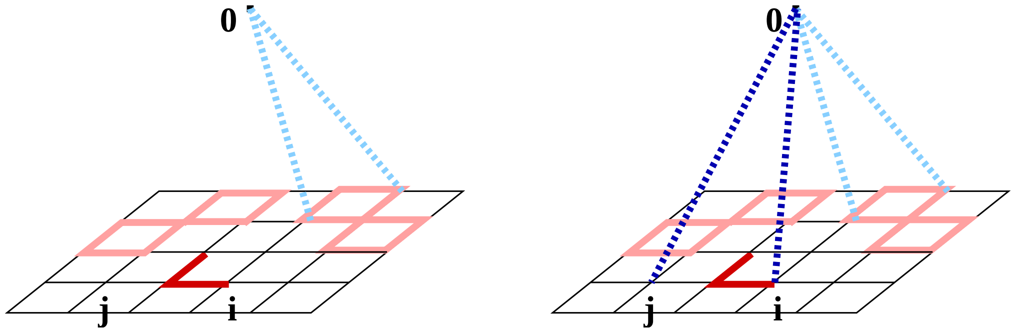

Figure 7.1: Configuration C_1 (left) and configuration C_2 (right).The dashed lines symbolize ghost-bonds.

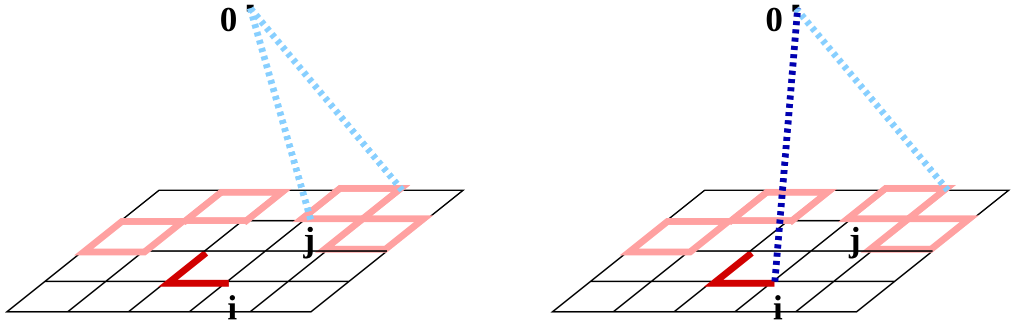

Figure 7.2: Configuration C_3 (left) and configuration C_4 (right). The dashed lines symbolize ghost-bonds.

7.2 Measurement of physical quantities including a magnetic field

7.2.1 Measurement of the energy

Inserting our partition function 7.3 including a homogeneous magnetic field $H_i \equiv H$ into the definition of the internal energy per lattice site:

$$\frac{E}{N} = -\frac{1}{N}\frac{\partial \ln Z}{\partial \beta} = -\frac{1}{N}\frac{1}{Z}\frac{\partial Z}{\partial \beta} \tag{7.15}$$

we get:

$$\begin{aligned}\frac{E}{N} = -\frac{1}{N}\frac{2^N}{Z}\Bigg(\bigg[&N_{nn} J \sinh(\beta J)\cosh(\beta J)^{N_{nn}-1}\cosh(\beta H)^N \\ &+ NH\sinh(\beta H)\cosh(\beta H)^{N-1}\cosh(\beta J)^{N_{nn}}\bigg]\cdot \sum_{\substack{\{b\},\{\tilde{b}\}\\ \partial b=\partial\tilde{b}}} v^{n_b}\cdot w^{n_{\tilde{b}}} \\ + \cosh(\beta J)^{N_{nn}}\cosh(\beta H)^N\cdot &\Bigg[\sum_{\substack{\{b\},\{\tilde{b}\}\\ \partial b=\partial\tilde{b}}} v^{n_b}\cdot w^{n_{\tilde{b}}}\frac{J n_b v^{n_b} w^{n_{\tilde{b}}}}{\cosh(\beta J)\sinh(\beta J)} + \frac{H n_{\tilde{b}} v^{n_b} w^{n_{\tilde{b}}}}{\cosh(\beta H)\sinh(\beta H)}\Bigg]\Bigg) \\ = -\frac{1}{N}\Bigg(J N_{nn} v + \frac{J\langle n_b\rangle}{\sinh(\beta J)\cosh(\beta J)} &+ HNw + \frac{H\langle n_{\tilde{b}}\rangle}{\sinh(\beta H)\cosh(\beta H)}\Bigg)\end{aligned} \tag{7.16}$$

Remembering that the number of nearest neighbors in a d-dimensional single cubic lattice with periodic boundary conditions is given by $N_{nn} = N\cdot d$ we end up with

$$\frac{E}{N} = -\left(\left[dv + \frac{\langle n_b\rangle}{N\sinh\beta J\cosh\beta J}\right] + \left[Hw + \frac{\langle n_{\tilde{b}}\rangle}{N\sinh(\beta H)\cosh(\beta H)}\right]\right) \tag{7.17}$$

Comparing eq. 7.17 with eq. 4.17, one sees that there now occurs an additional term due to the external magnetic field H.

7.2.2 Measurement of the magnetization

The magnetization M is defined via

$$\frac{M}{N} = \frac{1}{N}\frac{1}{\beta}\frac{\partial \ln Z}{\partial H} \tag{7.18}$$

Inserting the partition function 7.3 leads us to:

$$\frac{M}{N} = \frac{1}{N}\frac{1}{\beta}\frac{2^N}{Z}\cosh(\beta J)^N_{nn}\Bigg(N\beta\cosh(\beta H)^{N-1}\sinh(\beta H)\sum_{\substack{\{b\},\{\tilde{b}\}\\ \partial b=\partial\tilde{b}}} v^{n_b}\cdot w^{n_{\tilde{b}}}v^{n_b}w^{n_{\tilde{b}}}$$
$$+\cosh(\beta H)^N\sum_{\substack{\{b\},\{\tilde{b}\}\\ \partial b=\partial\tilde{b}}} v^{n_b}\cdot w^{n_{\tilde{b}}}\frac{\beta\cdot n_{\tilde{b}}\cdot v^{n_b}\cdot w^{n_{\tilde{b}}}}{\cosh(\beta H)\sinh(\beta H)}$$
$$= w + \frac{\langle n_{\tilde{b}}\rangle}{N\cosh(\beta H)\sinh(\beta H)} \tag{7.19}$$

7.2.3 Measurement of correlation functions and susceptibility including a magnetic field

The rule of measuring spatial correlation functions $G(i)$ remains unchanged.[2] (see eq. 4.2.1) Considering the external magnetic field, the susceptibility is given by:

$$\frac{\chi}{N} = \sum_i G(i) - M^2 \tag{7.20}$$

Since we have given rules both for measuring $\sum_i G_i$ (see chapter 4.2.2, respectively chapter 4.2.1) and M, χ can be obtained immediately.

[2]Note that there is no measurable correlation function $\langle s_i s_0\rangle$ and $\langle s_i s_0\rangle$ does not contribute to the susceptibility χ.

Chapter 8

Results including a magnetic field

Results for the physics of the ferromagnetic model in a homogeneous magnetic field were compared with those of a simple heatbath algorithm and agreed within their statistical errors.

8.1 Autocorrelation time results

Figures 8.1 and 8.2 show the integrated autocorrelation time τ_{int} of the energy and the magnetization. Since there is no Ising phase-transition at $H \neq 0$, the integrated autocorrelation time τ_{int} does not show any sizable dependence on the lattice size for $H = 1$ and even decreases for $H = 4$.

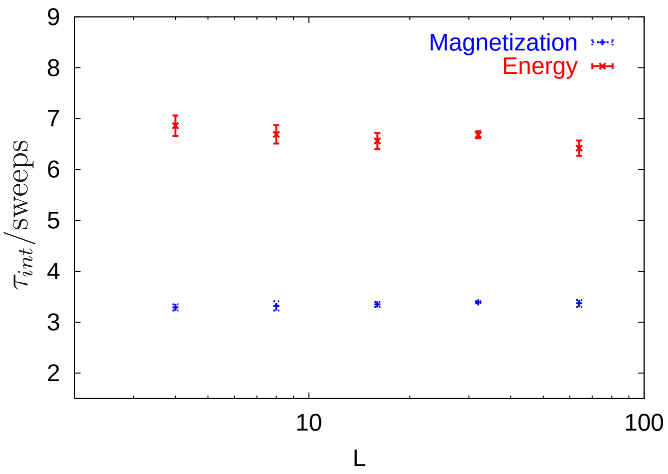

Figure 8.1: Integrated autocorrelation time τ_{int} of the magnetization (lower data-points) and energy (upper data-points) for a two-dimensional system with $H = 1$ and $\beta = 0.44$. L denotes the lattice size.

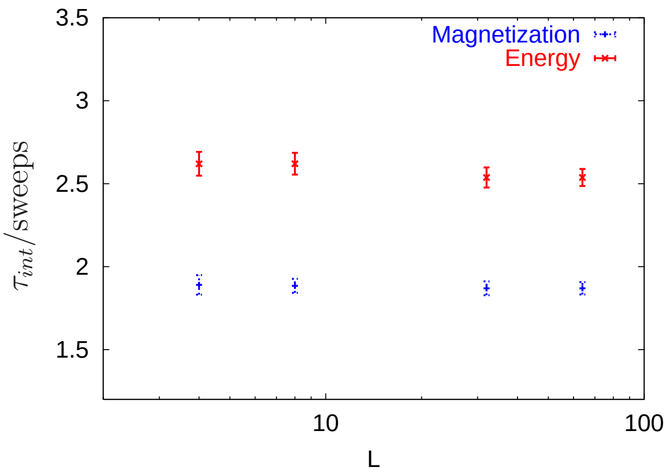

Figure 8.2: Integrated autocorrelation time τ_{int} of the magnetization (lower data-points) and energy (upper data-points) for a two-dimensional system with $H = 4$ and $\beta = 0.2$. L denotes the lattice size.

8.2 Dependence of the magnetization M on the magnetic field H

We verified the magnetization of a 16x16 system at $\beta = 0.2$ by varying the magnetic field from $H = 0.1$ until $H = 1.1$. The result is shown in figure 8.3. It agrees with the result of an implementation of the heatbath algorithm .

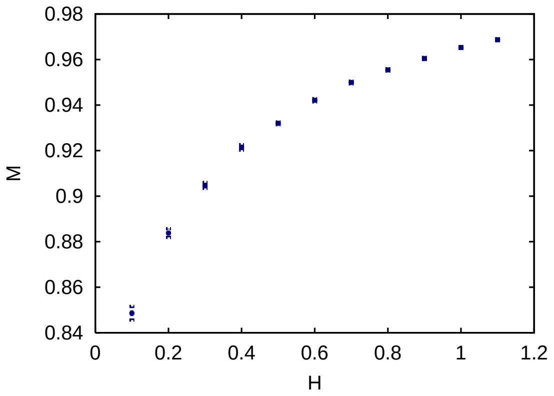

Figure 8.3: Dependence of the Magnetization M on an external magnetic field H for a 16x16 system.

Chapter 9

Capillary Condensation

The application prospects of nanoscale technology have created a fresh interest in the behavior of both simple fluids and complex fluids in pores or in a thin film geometry in layers confined by parallel walls [1]. By "capillary condensation" one means the finding, already discovered in the 19th century, that in a capillary the condensation of a gas occurs already at a lower pressure p than the coexistence pressure p_{coex} necessary to induce condensation in the bulk. Qualitatively, this shift of the transition can be attributed to the interaction of the fluid molecules with the attractive walls of the capillary.

9.1 The model

Invoking the well known isomorphism between the lattice gas model of fluids and the Ising model if magnetism (see chapter 1.2.2), we study the Ising model on the simple cubic lattice in presence of a bulk field H and a surface field H_1 in an $L \times L \times D$ geometry

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} s_i s_j - H \sum_i s_i - H_1 \sum_{i \in \text{surfaces}} s_i \tag{9.1}$$

The *surface field* H_1 corresponds to a chemical potential increment acting only close to the walls, on the scale of a lattice constant. Thus H_1 corresponds to a short range or "contact" interaction between the walls and the molecules in the film. In this system the values $T_c(D, H_1)$ and $H_c(D, H_1)$ at the critical point are both distinct from their bulk values T_c^∞ and $H_c^\infty (\equiv 0)$ [27]. (Values of $T_c(D, H_1)$ and $H_c(D, H_1)$ have been taken from reference [26] for our calculations and shown in table 9.1.)

The surface fields H_1 have the effect of shifting the transition between the coexisting phases away from the bulk location (see figure 9.1). Thus, for a fluid that

[1] we follow reference [26]

tends to wet the walls, the pressure p at which the condensation occurs in a thin film is reduced relative to its bulk coexistence value p_{coex}. In the magnetic terminology of the Ising model, capillary condensation means a change of of the sign of the total magnetization as a function of the magnetic field [28].

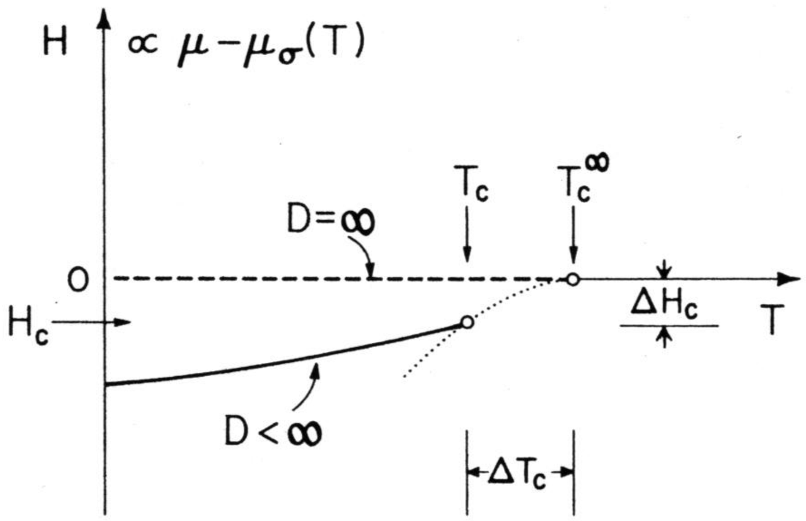

Figure 9.1: Phase diagram of a gas between plates at separation D showing the displaced phase boundary and shifted critical point [29].

We sampled simple cubic lattices in an $L \times L \times D$ geometry, where periodic boundary conditions act in the x and y direction, while we chose open boundary conditions in the z-direction. We used the Worm Algorithm extended to magnetic fields, introduced in the chapter 7 with $H_b = H$ in the bulk and $H_s =\mid H + H_1 \mid$ on the surfaces. Inserting the Hamiltonian 9.1 into the partition function 7.2 leads us to :

$$Z = \cosh(\beta J)^{N_{nn}} \cosh(\beta H_b)^{N_b} \cosh(\beta H_s)^{N_s} \sum_{\substack{\{b\},\{\tilde{b}\} \\ \partial b = \partial \tilde{b}}} v^{n_b} w_b^{n_{\tilde{b},b}} w_s^{n_{\tilde{b},s}} \tag{9.2}$$

where $v = \tanh \beta J, w_b = \tanh \beta H_b, w_s = \tanh \beta H_s$, $n_{\tilde{b},b}$ is the number of ghostbonds in the bulk and $n_{\tilde{b},s}$ the number of ghostbonds on the surfaces. We denote the number of lattice sites on the surface by N_s and the number of sites in the bulk by N_b.

9.2 The algorithm

The algorithm remains identical to that introduced in chapter 7.1, except for the heathbath-part. Since ghost-bonds set on the surfaces have a different weight from those in the bulk we have to distinguish between these two cases. The weight of a configuration is thus given by:

$$\begin{aligned} W_Z(\{b\},\{\tilde{b}\}) &= (\tanh \beta J)^{n_b})(\tanh \beta H_b)^{n_{\tilde{b},b}}(\tanh \beta H_s)^{n_{\tilde{b},s}} \\ &= v^{n_b} w_b^{n_{\tilde{b},b}} w_b^{n_{\tilde{b},s}} \end{aligned} \tag{9.3}$$

Modified Heathbath algorithm for Capillary Condensations

Step (3b) of the algorithm described in chapter 7.1 is modified in the following way:

Again we assume that $i_1 = i$, and N_b denotes the number of lattice sites in the bulk, N_s the number of lattice sites on the surfaces, respectively .

1. If there is no ghostbond connecting the lattice site i with the ghost-spin i_0, i.e $\tilde{b}_{i,0} = 0$, the total weight of all configurations which can be reached is given by:

   (a) If i is a lattice site on the surface $i \in \mathcal{S}$:

   $$\begin{aligned} W_{total}(n_{\tilde{b}}|\tilde{b}_{i,0} = 0, i \in \mathcal{S}) = [(N_b - n_{\tilde{b},b})w_s w_b + n_{\tilde{b},b}\frac{w_s}{w_b} \\ +(N_s - n_{\tilde{b},s} - 1)w_s^2 + (n_{\tilde{b},s} + 1)] \cdot W_{old} \end{aligned} \tag{9.4}$$

   where W_{old} denotes the weight of the configuration before the update. The update probabilities are now given by

   $$\begin{aligned} p(i \to j, \tilde{b}_{i,0} = 0 \to 1, \tilde{b}_{j,0} = 0 \to 1 \mid j \in \mathcal{S}) &= \frac{(N_s - n_{\tilde{b},s} - 1) \cdot w_s^2 \cdot W_{old}}{W_{total}} \\ p(i \to j, \tilde{b}_{i,0} = 0 \to 1, \tilde{b}_{j,0} = 0 \to 1 \mid j \in \mathcal{B}) &= \frac{(N_b - n_{\tilde{b},b}) \cdot w_s \cdot w_b \cdot W_{old}}{W_{total}} \\ p(i \to j, \tilde{b}_{i,0} = 0 \to 1, \tilde{b}_{j,0} = 1 \to 0 \mid j \in \mathcal{S}) &= \frac{(n_{\tilde{b},s} + 1) \cdot W_{old}}{W_{total}} \\ p(i \to j, \tilde{b}_{i,0} = 0 \to 1, \tilde{b}_{j,0} = 1 \to 0 \mid j \in \mathcal{B}) &= n_{\tilde{b},b}\frac{w_s \cdot W_{old}}{w_b \cdot W_{total}} \end{aligned} \tag{9.5}$$

   By construction for a heatbath method, they add up to unity.

   (b) If i is a lattice site in the bulk $i \in \mathcal{B}$:

   $$\begin{aligned} W_{total}(n_{\tilde{b}}|\tilde{b}_{i,0} = 0, i \in \mathcal{B}) = [(N_b - n_{\tilde{b},b} - 1)w_b^2 + (n_{\tilde{b},b} + 1) \\ +(N_s - n_{\tilde{b},s})w_s w_b + n_{\tilde{b},s}\frac{w_b}{w_s}] \cdot W_{old} \end{aligned} \tag{9.6}$$

The update probabilities are given by:

$$\begin{aligned}
p(i \to j, \tilde{b}_{i,0} = 0 \to 1, \tilde{b}_{j,0} = 0 \to 1 \mid j \in \mathcal{B}) &= \frac{(N_b - n_{\tilde{b},b} - 1) \cdot w_b^2 \cdot W_{old}}{W_{total}} \\
p(i \to j, \tilde{b}_{i,0} = 0 \to 1, \tilde{b}_{j,0} = 0 \to 1 \mid j \in \mathcal{S}) &= \frac{(N_s - n_{\tilde{b},s} \cdot) w_s \cdot w_b \cdot W_{old}}{W_{total}} \\
p(i \to j, \tilde{b}_{i,0} = 0 \to 1, \tilde{b}_{j,0} = 1 \to 0 \mid j \in \mathcal{B}) &= \frac{(n_{\tilde{b},b} + 1) \cdot W_{old}}{W_{total}} \\
p(i \to j, \tilde{b}_{i,0} = 0 \to 1, \tilde{b}_{j,0} = 1 \to 0 \mid j \in \mathcal{S}) &= n_{\tilde{b},s} \frac{w_b \cdot W_{old}}{w_s \cdot W_{total}}
\end{aligned} \tag{9.7}$$

2. If there is a ghostbond connecting the lattice site i with the ghost-spin i_0, i.e $\tilde{b}_{i,0} = 1$, the total weight of all configurations that can be reached is given by:

   (a) If i is a lattice site on the surface,$i \in \mathcal{S}$

$$\begin{aligned}
W_{total}(n_{\tilde{b}} | \tilde{b}_{i,0} = 1, i \in \mathcal{S}) = [&(N_b - n_{\tilde{b},b}) \frac{w_b}{w_s} + \frac{n_{\tilde{b},b}}{w_s w_b} \\
&+ (N_s - n_{\tilde{b},s} + 1) + \frac{n_{\tilde{b},s} - 1}{w_s^2}] \cdot W_{old}
\end{aligned} \tag{9.8}$$

   where W_{old} denotes the weight of the configuration before the update. The probabilities for changing the configuration are:

$$\begin{aligned}
p(i \to j, \tilde{b}_{i,0} = 1 \to 0, \tilde{b}_{j,0} = 0 \to 1 \mid j \in \mathcal{S}) &= \frac{(N_s - n_{\tilde{b},s} + 1) \cdot W_{old}}{W_{total}} \\
p(i \to j, \tilde{b}_{i,0} = 1 \to 0, \tilde{b}_{j,0} = 0 \to 1 \mid j \in \mathcal{B}) &= \frac{(N_b - n_{\tilde{b},b}) \cdot w_b \cdot W_{old}}{w_s W_{total}} \\
p(i \to j, \tilde{b}_{i,0} = 1 \to 0, \tilde{b}_{j,0} = 1 \to 0 \mid j \in \mathcal{S}) &= \frac{(n_{\tilde{b},s} - 1) \cdot W_{old}}{w_s^2 \cdot W_{total}} \\
p(i \to j, \tilde{b}_{i,0} = 1 \to 0, \tilde{b}_{j,0} = 1 \to 0 \mid j \in \mathcal{B}) &= n_{\tilde{b},b} \frac{W_{old}}{w_b \cdot w_s \cdot W_{total}}
\end{aligned} \tag{9.9}$$

   (b) If i is a lattice site in the bulk,$i \in \mathcal{B}$:

$$\begin{aligned}
W_{total}(n_{\tilde{b}} | \tilde{b}_{i,0} = 1, i \in \mathcal{B}) = [&(N_b - n_{\tilde{b},b} + 1) + \frac{n_{\tilde{b},s}}{w_s w_b} \\
&+ (N_s - n_{\tilde{b},s}) \frac{w_s}{w_b} + \frac{n_{\tilde{b},b} - 1}{w_b^2}] \cdot W_{old}
\end{aligned} \tag{9.10}$$

$$
\begin{aligned}
p(i \to j, \tilde{b}_{i,0} = 1 \to 0, \tilde{b}_{j,0} = 0 \to 1 \mid j \in \mathcal{B}) &= \frac{(N_b - n_{\tilde{b},b} + 1) \cdot W_{old}}{W_{total}} \\
p(i \to j, \tilde{b}_{i,0} = 1 \to 0, \tilde{b}_{j,0} = 0 \to 1 \mid j \in \mathcal{S}) &= \frac{(N_s - n_{\tilde{b},s}) \cdot w_s \cdot W_{old}}{w_b \cdot W_{total}} \\
p(i \to j, \tilde{b}_{i,0} = 1 \to 0, \tilde{b}_{j,0} = 1 \to 0 \mid j \in \mathcal{B}) &= \frac{(n_{\tilde{b},b} - 1) \cdot W_{old}}{w_b^2 \cdot W_{total}} \\
p(i \to j, \tilde{b}_{i,0} = 1 \to 0, \tilde{b}_{j,0} = 1 \to 0 \mid j \in \mathcal{S}) &= n_{\tilde{b},s} \frac{W_{old}}{w_b \cdot w_s \cdot W_{total}}
\end{aligned}
\tag{9.11}
$$

By construction the probabilities 9.4-9.11 satisfy the detailed-balance equation 3.13, in the same way as in chapter 7.1.1 .

9.3 Measurement of physical quantities

Due to the inhomogeneous external field, we distinguish between surface-magnetization M_s, from which we can derive a surface susceptibility $\chi_s = \frac{\partial M_s}{\partial H_s}$ and a bulk-magnetization M_b from which we define a bulk susceptibility $\chi_s = \frac{\partial M_b}{\partial H_b}$. Finally we define a total susceptibility :

$$
\chi_{\text{total}} = \frac{\partial M}{\partial H} \tag{9.12}
$$

with $M = M_b + M_s$. The way of measuring the total susceptibility remains the same as in chapter 7.2.3.

From equation 9.2 we derive the estimators for energy and magnetization.

9.3.1 Energy

Starting from eq. 9.2, executing operations analogous to eq. 7.16 - 7.17, we end up with:

$$
\begin{aligned}
\frac{E}{N} = -\Bigg(& \left[\frac{JN_{nn}v}{N} + \frac{\langle n_b \rangle}{N \sinh\beta \cosh\beta}\right] \\
& + \left[\frac{H_b N_b w_b}{N} + \frac{\langle n_{\tilde{b},b} \rangle}{N \sinh(\beta H_b) \cosh(\beta H_b)}\right] \\
& + \left[\frac{H_s N_s w_s}{N} + \frac{\langle n_{\tilde{b},s} \rangle}{N \sinh(\beta H_s) \cosh(\beta H_s)}\right] \qquad (9.13)
\end{aligned}
$$

9.3.2 Magnetization

In the same way as in section 7.2.2 we write the magnetization as:

$$
\begin{aligned}
\frac{M}{N} &= \left(\frac{N_b w_b}{N} + \frac{\langle n_{\tilde{b},b} \rangle}{N \cosh(\beta H_b) \sinh(\beta H_b)}\right) \\
&+ \left(\frac{N_s w_s}{N} + \frac{\langle n_{\tilde{b},s} \rangle}{N \cosh(\beta H_s) \sinh(\beta H_s)}\right) \\
&= M_b + M_s \qquad (9.14)
\end{aligned}
$$

9.4 Results

Note added: The calculations for capillary condensation use the same sign for the magnetic field at the boundary and in the bulk. Therefore the systems simulated are NOT near a critical point and are NOT the systems simulated by Binder et al.

We considered film thicknesses $D = 4, 8, 12, 16, 24, 28, 32$ for an $L \times L \times D$ geometry with $L = 4D$ ($L >> D$ should hold!). All measurements were taken at the critical point, with a fixed surface field $H_1 = -0.015$, the temperature T and the bulk-field H_b were adjusted to their critical values $T_c(D)$ and $H_c(D)$ (see table 9.1).

9.4.1 Autocorrelation times

We measured the integrated autocorrelation times of the magnetization and the energy. The results are shown in table 9.2 and 9.3.

The integrated autocorrelation times of the magnetization were fitted to a power law $\tau_{int}(D) = aD^z$ with $a = 12.4 \pm 1.0$, $z = -0.52 \pm 0.03$ and $\chi_r = 0.9$ (see figure 9.2,

D	$T_c(D)$	$H_c(T)$
4	3.8705(3)	0.006644(32)
8	4.2409(3)	0.002528(14)
12	4.3561(3)	0.001367(10)
16	4.4084(3)	0.000867(6)
24	4.4549(5)	0.000448(3)
28	4.4665(4)	0.000348(3)
32	4.4749(5)	0.000279(3)

Table 9.1: Critical temperatures T_c and fields H_c as a function of the film thickness D. (The values are taken from reference [26].)

top). Surprisingly, they become smaller as the lattice size increases. The integrated autocorrelations of the energy are nearly constant for all lattice sizes (see figure 9.3, top).

One possible reason would be that we are not actually precisely at the critical point.

Another subtlety concerns the definition of a *sweep*. Our system is no longer translationally invariant, and the worm might spend most of its time close to the surfaces. In this case it might be more appropriate to normalize autocorrelation times by the surface volume $L\times Ł$ instead of the bulk volume $L\times L\times D = L\times L\times L/4$. Doing so results in fig. 9.2 (bottom) and fig. 9.3 (bottom), in which the autocorrelation times increase with lattice size. For the magnetization we found $\tau_{int} = a \cdot L^z$ with $a = 0.015 \pm 0.004$ and $b = 1.23 \pm 0.005$, and for energy $\tau_{int} = a \cdot L^z$ with $a = 11 \pm 3$ and $b = 0.83 \pm 0.006$.

D	τ_{int}/sweeps	$\Delta\tau_{int}$/sweeps	M/sweeps
4	5.08	0.09	274000
8	4.28	0.1	137000
12	3.29	0.1	91000
16	2.89	0.1	82000
24	2.14	0.2	45000
28	2.23	0.11	37000
32	2.00	0.17	32000

Table 9.2: Integrated correlation time τ_{int} of magnetization in dependence of the film thickness D. M denotes the number of measurements that contributed to the statistics. The measurements are taken at the critical points with values of $T_c(D)$ and $H_c(D)$ shown in table 9.1.

D	τ_{int}/sweeps	$\Delta\tau_{int}$/sweeps	M/sweeps
4	13.9	0.3	274000
8	14.0	0.4	137000
12	14.9	0.6	91000
16	18.7	1.4	82000
24	12.5	1.7	2100
28	2.23	0.11	37000
32	2.00	0.17	32000

Table 9.3: Integrated correlation time τ_{int} of energy in dependence of the film thickness D. M denotes the number of measurements that contributed to the statistics.The measurements are taken at the critical points with values of $T_c(D)$ and $H_c(D)$ shown in table 9.1

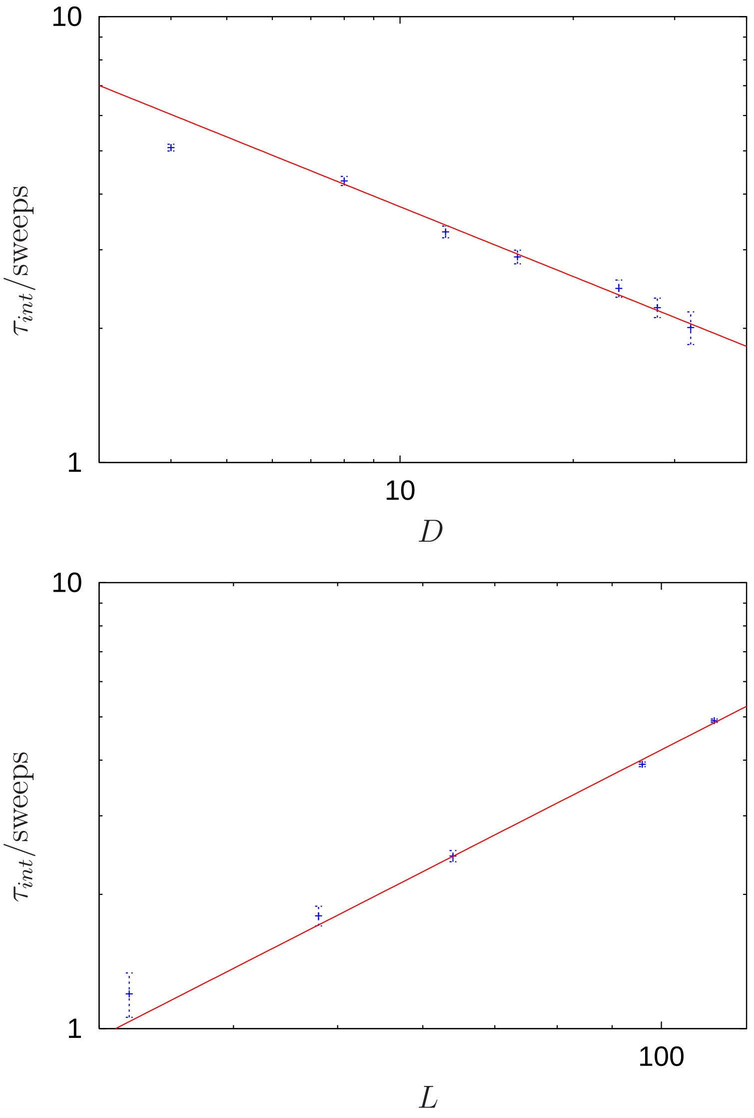

Figure 9.2: Integrated autocorrelation times τ_{int} of magnetization of thin Isings films of $L \times L \times D$ geometry in dependence of the film thickness D and $L = 4D$. All systems were simulated at the critical point with $H_1 = -0.015$ and $T_c(D), H_c(D)$ shown in table 9.1. In the upper figure we assumed the algorithm to be dominated by three dimensional behavior (thus 1 sweep= $3 \cdot L^3$ elementary steps). The autocorrelation time results were fitted to $\tau_{int}(D) = aD^z$ with $a = 12.4 \pm 1.0$, $z = -0.52 \pm 0.03$ and $\chi_r = 0.9$. The lower figure shows the same data, but now we assumed the algorithm to be dominated by two dimensional behavior (thus 1 sweep= $2 \cdot L^2$ elementary steps). This data was fitted to $\tau_{int}(L) = aL^z$ with $a = 0.015 \pm 0.004$, $z = 1.23 \pm 0.005$.

.

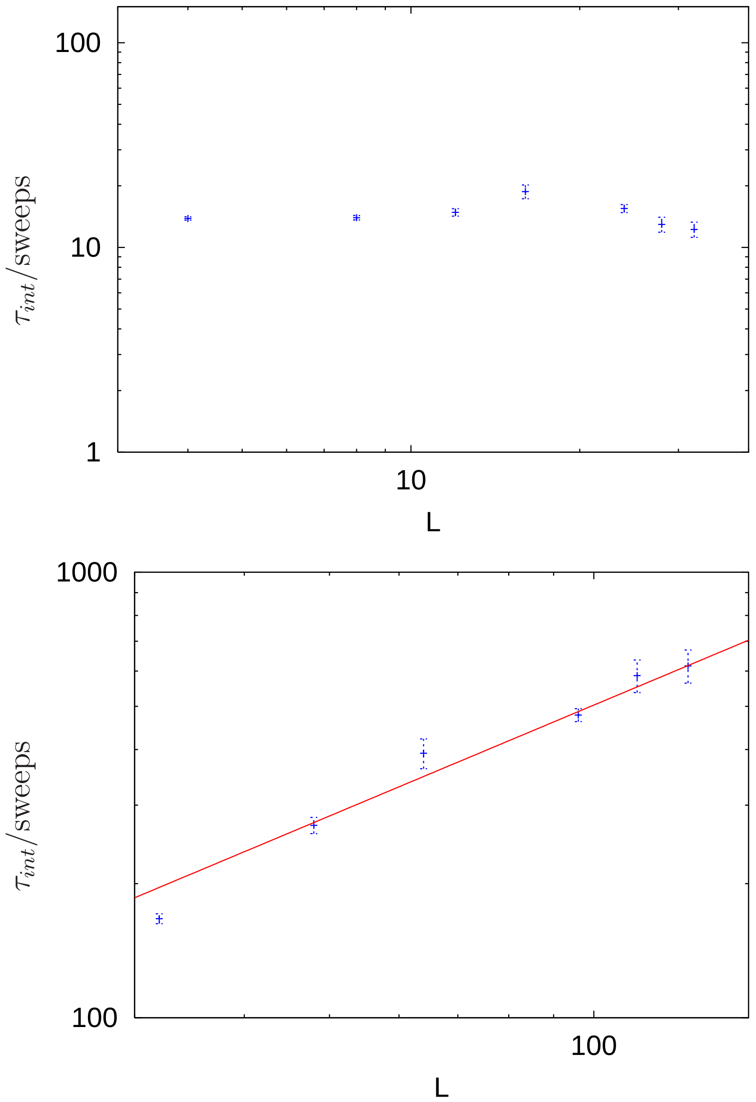

Figure 9.3: Integrated autocorrelation times τ_{int} of energy of thin Isings films of $L \times L \times D$ geometry in dependence of the film thickness D and $L = 4D$. All systems were simulated at the critical point with $H_1 = -0.015$ and $T_c(D), H_c(D)$ shown in table 9.1 .In the upper figure we assumed the algorithm to be dominated by three dimensional behavior (thus 1 sweep= $3 \cdot L^3$ elementary steps).The lower figure shows the same data, but now we assumed the algorithm to be dominated by two dimensional behavior (thus 1 sweep= $2 \cdot L^2$ elementary steps). This data was fitted to $\tau_{int}(L) = aL^z$ with $a = 11 \pm 3$, $z = 0.83 \pm 0.006$.

The autocorrelation functions of the magnetization and of the energy for $D = 4, 8, 12, 16, 24, 28, 32$ are shown in appendix B.

9.4.2 Finite-size scaling of the total susceptibility $\chi_{\mathbf{total}}$

We studied the critical behavior of the total susceptibility χ_{total} via a finite-size scaling analysis, fitting the values to $\chi(D) = aD^b$ with $a = 0.89 \pm 0.03$, $b = 2.05 \pm 0.01$ and $\chi_r = 0.2$.

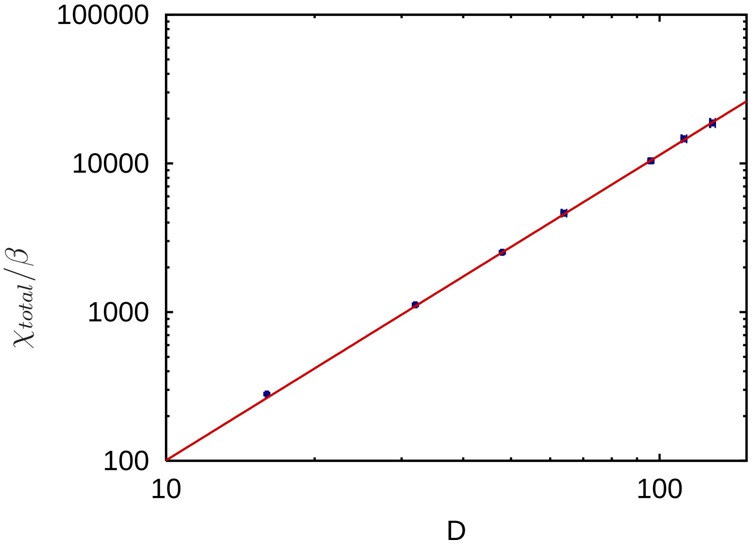

Figure 9.4: Total susceptibility χ_{total} divided through β measured at the critical points (see table 9.1) .

More finite-size scaling on the susceptibility can be found in [26].

Chapter 10

Conclusion

In this thesis we applied the Worm algorithm to the Ising model and compared it to alternative cluster algorithms. We started with the high-temperature expansion of the model's partition function, in which only those configurations consisting of closed loops of bonds yield to a contribution to Z.

We constructed the Monte Carlo algorithm and derived the improved estimators for the physical quantities energy, susceptibility and spatial correlation functions. We reviewed existing Monte Carlo algorithms and compared them to the Worm Algorithm in terms of autocorrelation behavior.

We found very small integrated autocorrelation times for the spatial correlation functions $G(i)$ and susceptibility χ which can be bounded by $\tau_{int}(G(i), \chi) < \langle \sum_{ij} s_i s_j \rangle$ elementary steps.

For the energy we found a behavior compatible with logarithmic dependence of the integrated autocorrelation time on the lattice size L, respectively for the critical dynamical exponent $z \approx 0$ in two and three dimensions. However, a detailed comparison of our Worm Algorithm with other algorithms, such as the Swendsen Wang, Wolff and Joint Representation algorithm made clear that the advantage of this logarithmic scaling makes our Worm Algorithm more effective than to SW and JCA only when lattice sizes $L > 80$ in three dimensions are reached. The Wolff algorithm turns out to have smaller integrated autocorrelation times for only of two and three-dimensional lattices with $L \leq 10^8$.

In a further step we applied algorithms for the Ising Model including an external magnetic field. We reviewed several methods, pointing out their advantages and disadvantages. We extended the Worm Algorithm for Ising Models to including external field H via introducing a ghost spin and found very small autocorrelation times even in a strong magnetic field.

Finally we applied the extended Worm Algorithm to Capillary Condensation. We performed an autocorrelation time analysis at the critical point and found a negative critical exponent $z \approx -0.5$ for the magnetization while the integrated autocorrelation time of energy did not show any dependence on the lattice size.

Appendix A

Autocorrelation functions in d=2, d=3

In the following pages we depict the autocorrelation functions $C(t)$ of the energy, the susceptibility and the correlation function[1] $G(L/2)$ with their errors. The evaluation of $C(t)$ was treated in chapter 5.1. The time t is given in units of $(1/V)\langle\sum_{ij} s_i s_j\rangle$ elementary steps. The statistics of the several MC-runs can be extracted from table 5.2 for $d = 2$ and table 5.4 for $d = 3$.

Energy in d=2, d=3

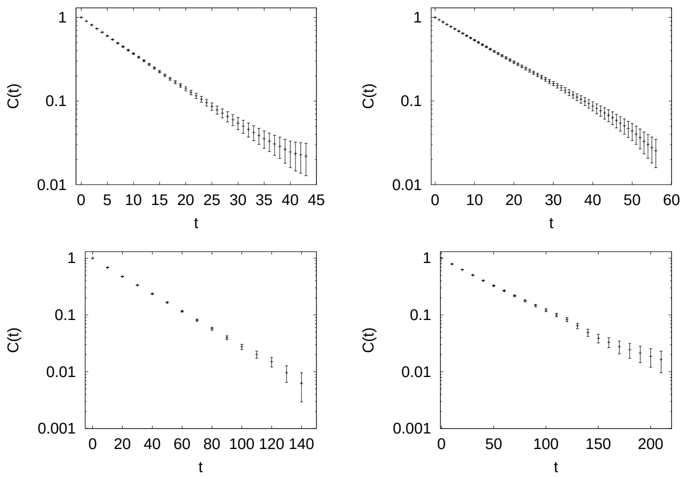

Figure A.1: Autocorrelation function $C(t)$ of the energy on a 16×16 (top left), a 32×32 (top right) system, a 32×32 (bottom left), and a 64×64 (bottom right) system.

[1]The data was produced via the field-less Worm Algorithm

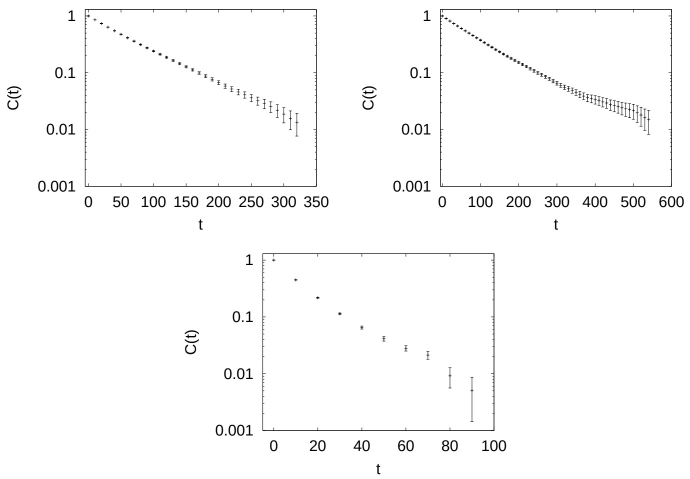

Figure A.2: Autocorrelation function $C(t)$ of the energy on a 64×64 (top left), 128×128 (top right), and 256×256 (bottom) system.

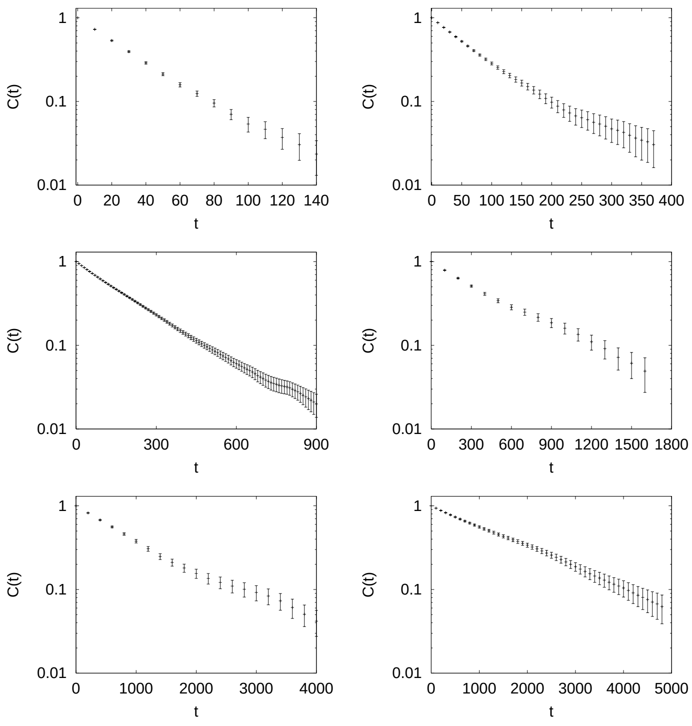

Figure A.3: Autocorrelation function $C(t)$ of the energy on a $4 \times 4 \times 4$ (top left), a $8 \times 8 \times 8$ (top right) system, a $16 \times 16 \times 16$ (center left), a $32 \times 32 \times 32$ (center right) system, a $64 \times 64 \times 64$ (bottom left), and a $128 \times 128 \times 128$ (botttom right) system.

Susceptibility in d=2

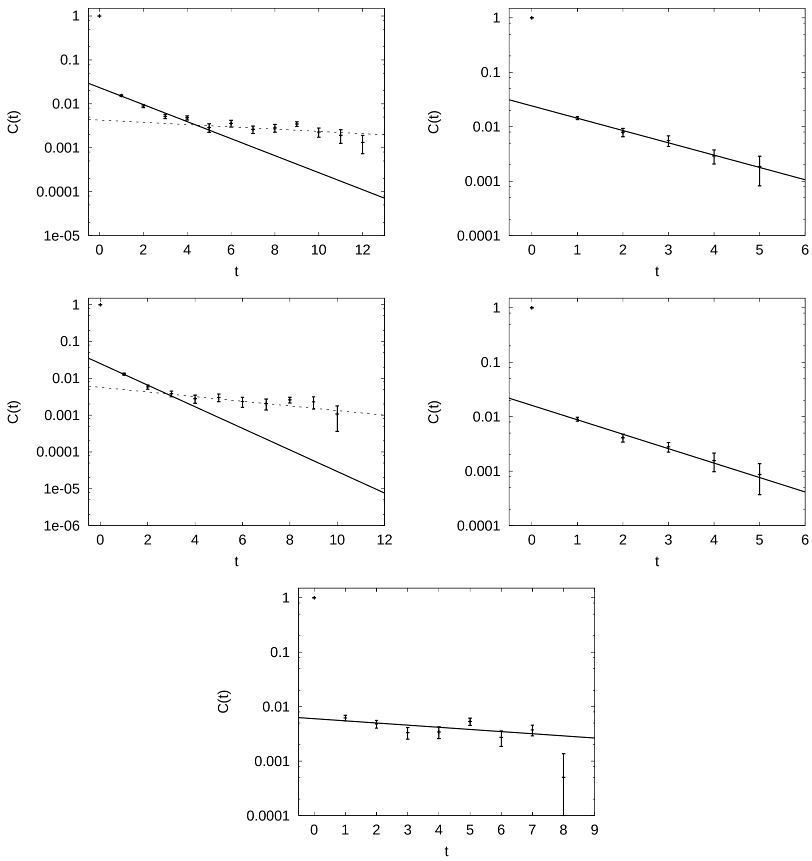

Figure A.4: Autocorrelation function of the susceptibility on a 16×16 system (top left) with $\tau_{exp,1} = 8 \pm 1$ sweeps, $\tau_{exp,2} = 50 \pm 30$ sweeps, a 32×32 system (top right) with $\tau_{exp,1} = 8.5 \pm 0.5$ sweeps, a 64×64 system (center left) with $\tau_{exp,1} = 8 \pm 1$ sweeps, $\tau_{exp,2} = 50 \pm 30$, a 128×128 system (center right) with $\tau_{exp,1} = 9.5 \pm 0.8$ sweeps, and a 256×256 system.

Susceptibility in d=3

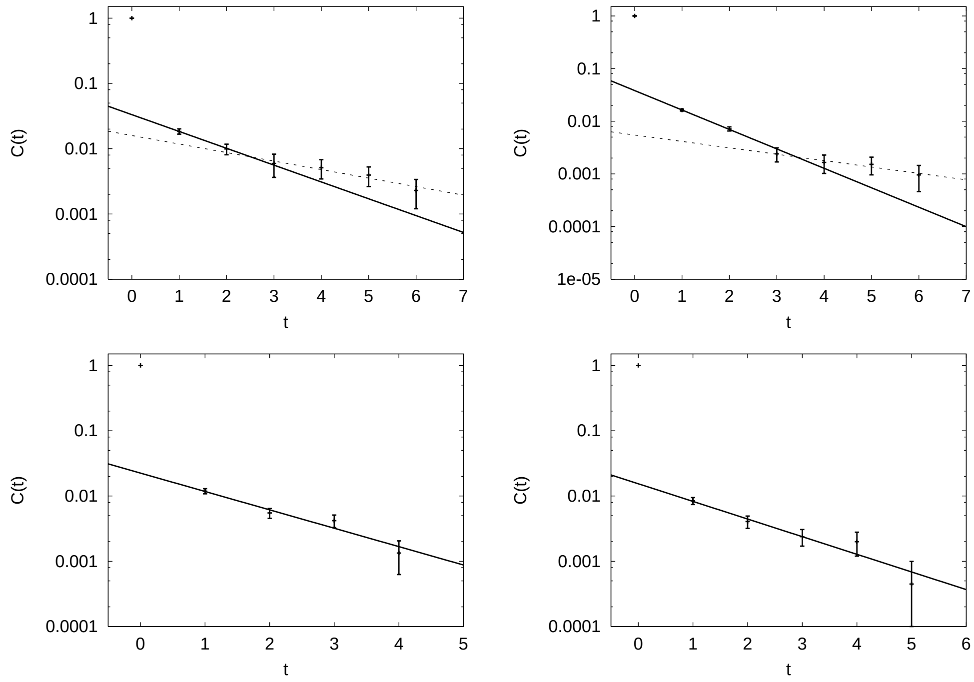

Figure A.5: Autocorrelation function of the susceptibility on a $8 \times 8 \times 8$ system (top left) with $\tau_{exp,1} = 2.4 \pm 0.1$ sweeps, $\tau_{exp,2} = 4.8 \pm 1$ sweeps and a $16 \times 16 \times 16$ system (top right) with $\tau_{exp,1} = 1.7 \pm 0.17$ sweeps, $\tau_{exp,2} = 5.2 \pm 1$ sweeps, of a $32 \times 32 \times 32$ system (bottom left) with $\tau_{exp,1} = 2.3 \pm 0.6$ sweeps and a $64 \times 64 \times 64$ system (bottom right) with $\tau_{exp,1} = 2.1 \pm 0.2$ sweeps.

Spatial correlation function $G(L/2)$ in axial directions in d=2

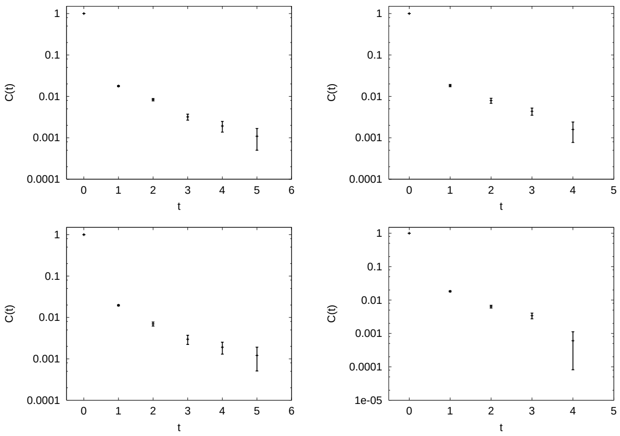

Figure A.6: Autocorrelation function of the spatial correlation function $G(L/2)$ in axial directions on a 16×16 system (top left), a 32×32 system (top right), a 64×64 system (bottom left) and a 128×128 system (bottom right).

Spatial correlation function $G(L/2)$ in axial directions in d=3

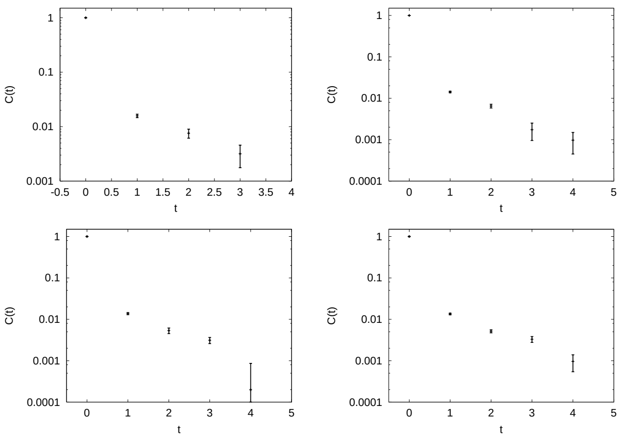

Figure A.7: Autocorrelation function of the spatial correlation function $G(L/2)$ in axial directions on a $8 \times 8 \times 8$ system (top left), a $16 \times 16 \times 16$ system (top right), a $32 \times 32 \times 32$ system (bottom left) and a $64 \times 64 \times 64$ system (bottom right).

Spatial correlation function $G(L/2)$ in radial directions in d=2

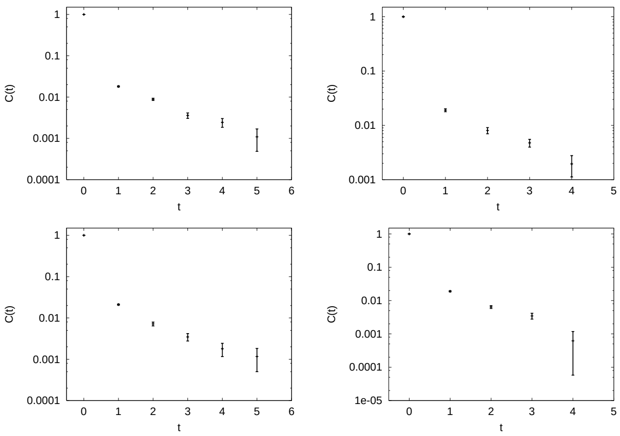

Figure A.8: Autocorrelation function of the spatial correlation function $G(L/2)$ in radial directions on a 16×16 system (top left), a 32×32 system (top right), a 64×64 system (bottom left) and a 128×128 system (bottom right).

Spatial correlation function $G(L/2)$ in radial directions in d=3

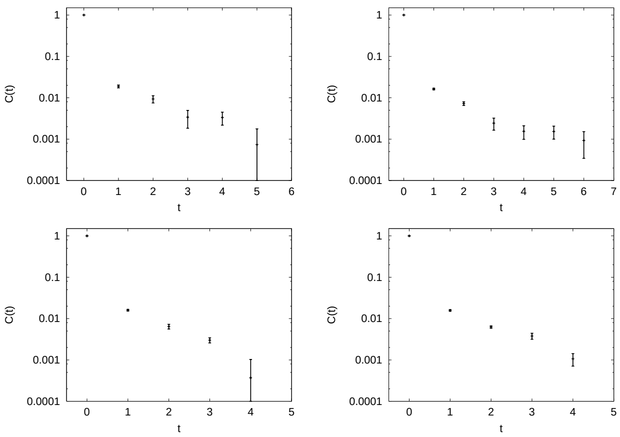

Figure A.9: Autocorrelation function of the spatial correlation function $G(L/2)$ in radial directions on a $8 \times 8 \times 8$ system (top left), a $16 \times 16 \times 16$ system (top right), a $32 \times 32 \times 32$ system (bottom left) and a $64 \times 64 \times 64$ system (bottom right).

Appendix B

Autocorrelation functions Capillary Condensation

All measurements are taken at the critical point with $H_1 = -0.015$ and $H = H_c(D), T = T_c(D)$ given in table 9.1. The number of sweeps per system can be extracted from table 9.2 resp. table 9.3. The time t is given in units of $(1/V)\langle\sum_{ij} s_i s_j\rangle$ elementary steps.

Magnetization

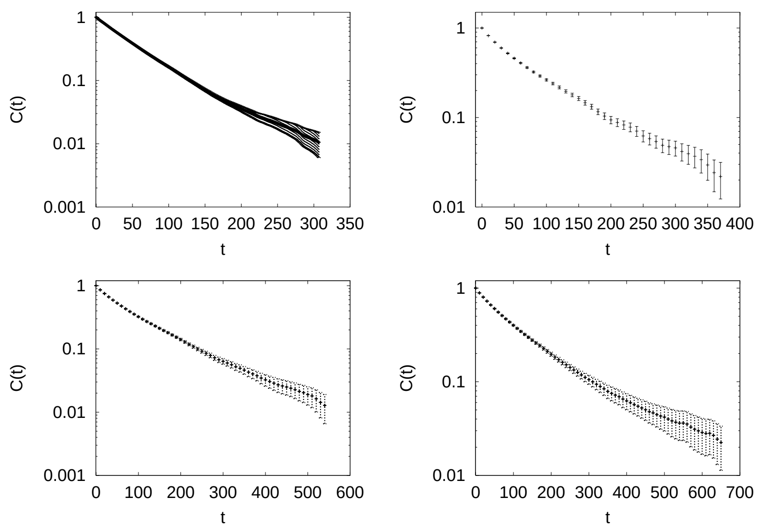

Figure B.1: Autocorrelation function $C(t)$ of the magnetization on a $16\times16\times4$ (top left), a $32\times32\times8$ (top right), a $48\times48\times12$ (bottom left) and a $64\times64\times16$ (bottom right) system.

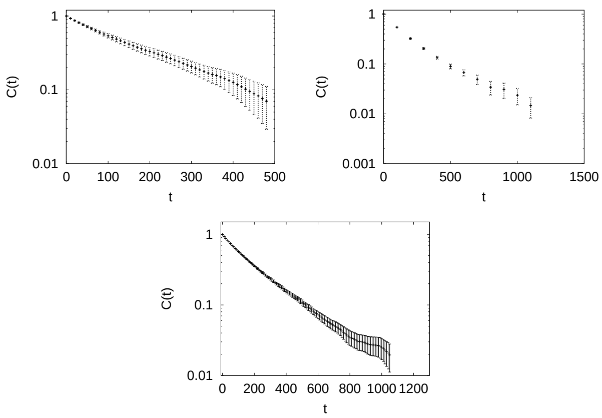

Figure B.2: Autocorrelation function $C(t)$ of the magnetization on a $96 \times 96 \times 24$ (top left) system, a $112 \times 112 \times 28$ system (top right) and a $128 \times 128 \times 32$ system (bottom)

Energy

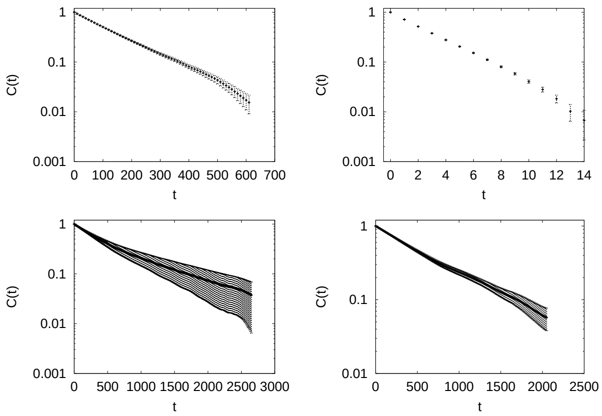

Figure B.3: Autocorrelation function $C(t)$ of the energy on a $16 \times 16 \times 4$ (top left),a $32 \times 32 \times 8$ system (top left), a $48 \times 48 \times 12$ (bottom left) and a $64 \times 64 \times 16$ system (bottom right). The number of sweeps per system can be extracted from table 9.3 resp. table 9.2.

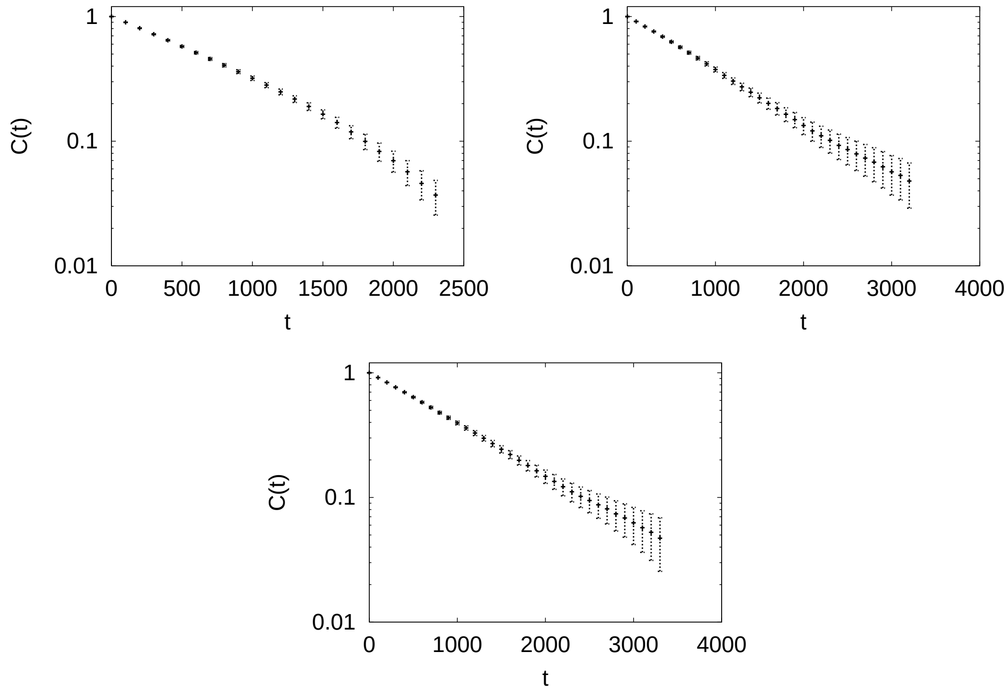

Figure B.4: Autocorrelation function $C(t)$ of the energy on a $96 \times 96 \times 24$ (top left) system, a $112 \times 112 \times 28$system (top left) and a $128 \times 128 \times 32$ system (bottom)

Bibliography

[1] J. M. Yeomans. *Statistical Mechanics of Phase Transitions.* Oxford, 1993.

[2] Onsager. Crystal statistics. I. A two-dimensional model with an order-disorder transition. *Phys. Rev. 65*, 1944.

[3] H.M. Erkinger H.G. Evertz and W. von der Linden. *Computer Simulation Studies in Condensed Matter Physics XIV*, page 123. Springer, 2002.

[4] D. P. Landau and K. Binder. *A Guide to Monte Carlo Simulations in Statistical Physics.* Cambridge University Press, 2000.

[5] Guggenheim. *Journal of Chemical Physics*, 13 253.

[6] J. J. Binney, N. J. Dowrick, A. J. Fisher, and M. E. J. Newman. *The Theory of Critical Phenomena An Introdoction to the Renormalization Group.* Oxford Science Publications, 1992.

[7] W. von der Linden and A. Pruell. Wahrscheinlichkeitstheorie Statistik und Datenanalyse. *Skriptum*, 2000.

[8] W. von der Linden. Numerical treatment of Many-Body Problems. *www.itp.tu-graz.ac.at.*

[9] H. G. Evertz. The Loop Algorithm. *cond-mat/9707221*, 2002.

[10] J. R. Heringa and H. W. J. Blöte. Geometric cluster Monte Carlo simulation. *Physical Review E*, 57(5), 1998.

[11] J. S. Wang, O. Kozan, and R. H. Swendsen. Comments on Sweeny and Gliozzi dynamics for simulations of Potts models in the Fortuin-Kasteleyn representation. *cond-mat/0206437*, 2002.

[12] R. H. Swendsen and J. S. Wang. *PRL*, 58:86, 1987.

[13] R.G. Edwards and A.D. Sokal. *Phys.Rev.D.*, 38:2009, 1988.

[14] J. S. Wang. Efficient Monte Carlo Simulation Methods in Statistical Physics. *cond-mat/0103318*, 2001.

[15] M. D'Onorio De Meo, D. W. Heermann, and K. Binder. Monte Carlo study of the Ising Model Phase Transition in Terms of the Percolation Transition of "Physical Clusters". *J Stat Phys*, 60(585), 1990.

[16] U. Wolff. Collective monte carlo updating for spin systems. *PRL*, 62:361, 1989.

[17] H. M. Erkinger. A new cluster algorithm for the Ising model. Diplomarbeit, 2000. Inst. f. Theor. Physik TU Graz.

[18] R. Ben-Av, D. Kandel, E. Katznelson, P. G. Lauwers, and S. Solomon. Critical Acceleration of Lattice Gauge Simulations. *J Stat Phys 58*, 125, 1990.

[19] N. Prokof'ev and B. Svistunov. Worm Algorithms for Classical Statistical Models. *Physical Review Letters*, 87(16), October 2001.

[20] N. V. Prokof'ev, B. V. Svistunov, and I. S. Tupitsyn. "Worm" Algorithm in Quantum Monte Carlo Simulations. *Physical Letters A*, 238:238–257, 1998.

[21] H. G. Evertz and W. von der Linden. Simulations on Infinite Size Lattices. *Phys. Rev. Lett.*, 86:5164, 2001.

[22] C. F. Baillie and P. D. Coddington. Comparison of cluster algorithms for two-dimensional Potts models . *Physical Review B*, 43, 1991.

[23] J. S. Wang. Clusters in the three-dimensional Ising model with a magnetic field. *Physica A*, 161:249, 1989.

[24] O. Redner, J. Machta, and L. F. Chayes. Graphical representations and cluster algorithms for critical points with fields. *cond-mat/9802063*, 1998.

[25] H. W. Blöte, J. R. Heringa, and M. M. Tsypin. Three-dimensional Ising Model in the fixed-magnetization ensemble: a Monte Carlo Study. *cond-mat/9910145*, 1999.

[26] O. Dillmann, W. Janke, M. Müller, and K. Binder. A Monte Carlo Test of the Fisher-Nakanishi-Scaling Theory for the Capillary Condensation Critical Point. *cond-mat/0101198*, 2001.

[27] H. Nakanishi and M. E. Fisher. Critical point shifts in films. *J. Chem. Phys.*, 78:3279, 1983.

[28] K. Binder, D. P. Landau, and A. M. Ferrenberg. Thin Ising films with competing walls: A Monte Carlo study. *Physical Review E*, 51(4), 1995.

[29] M. E. Fisher and H. Nakanishi. Scaling theory of the criticality of fluids between plates. *J.Chem.Phys.*, 75(12):5857, 1981.