Contents

1. Stochastic Methods

This chapter is concerned with methods which use stochastic elements to compute quantities of interest. These methods are not diametrically opposed to the deterministic ones. Brownian dynamics provides an example where the two methods are combined to form a hybrid technique. However, there are also inherently stochastic methods, such as the Monte-Carlo technique. An application of this simulation method was presented in the introductory chapter. The stochastic methods are built on concepts developed in probability theory and statistical mechanics. They allow not only a treatment of problems apparently probabilistic in nature, like the random walk, but also problems which are on the face deterministic. The scope of applications is broad, making it a flexible and exciting tool in simulational physics. One of the key elements in these types of simulations is the concept of the Markov process or Markov chain.

1.1 Monte-Carlo Method

In this section we outline the principles behind the Monte-Carlo method. In the subsections specific methods for different thermodynamic ensembles will be presented and how they are applied to problem from statistical mechanics.

1.1.1 Simple Sampling: Random Walks

Perhaps the most straight-forward application of probability can be seen in the *random walk model*. This model serves as a second introduction to sample techniques for configurations that one uses in stochastic simulation methods beside the percolation model that we introduced in the first chapter of this book.

We assume a lattice. For simplicity we take a simple square lattice. On this lattice a particle or walker is placed. The walker regards this initial position as the origin. The walker draws a random number and decides, according to the drawn random number, to go to a new position on the lattice. The new position must be one of the nearest neighbours, and each of the neighbours has the same probability to be visited. Once he is at the new position, the walker regards this position as his new origin. In other words, he immediately forgets where he came from. Every step is made as if it is the first step. All steps are then independent of each other. Schematically a walk of length 10 is shown in Figure 1.1.

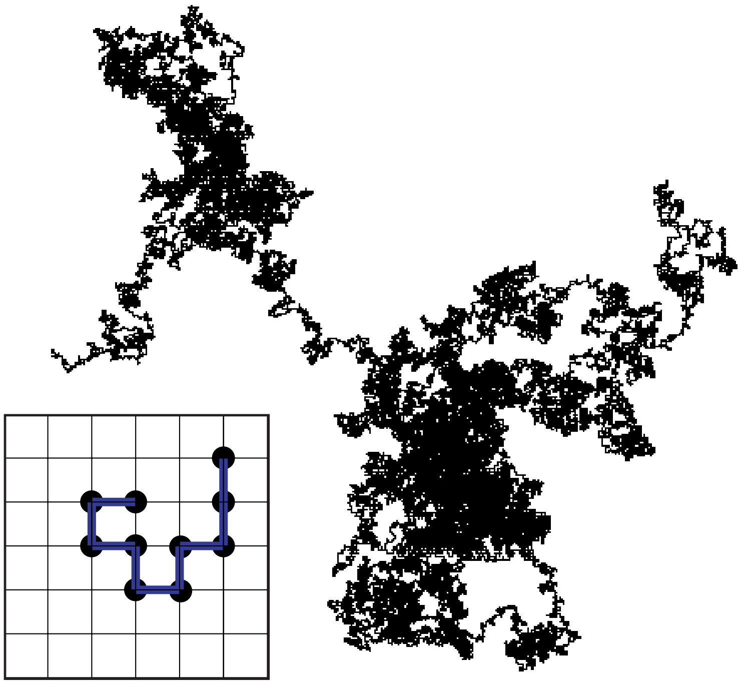

Figure 1.1. Example of a conformation of a random walk, the underlying lattice and the construction of the walk

It is very easy to simulate a random walk on the computer. The following program segment generates such random walks.

Simple Sampling of Random Walks

Example 5.1

It is assumed that in the array `random` are stored numbers which are uniformly distributed in the interval $(0, 1)$. A random number from the array is then multiplied by 4 and converted to an integer value. This integer value can either be $0, 1, 2$ or 3 labeling the four possible directions or nearest neighbours on the square lattice. The numbers $0, 1, 2$ and 3 are uniformly distributed as long as the numbers in the array random are so distributed. Depending on the direction the random number points to, the walker occupies the appropriate position on the lattice by increasing or decreasing the `x` or `y` variable. The variables `xn` and `yn` hold the new position of the random walker.

```
/* ---- Choose a new nn site ---- */
i = floor(random[index++]* 4.0);
switch (i) {
   case 0: xn = x-1;
           yn = y;
           break;
   case 1: yn = y-1;
           xn = x;
           break;
   case 2: yn = y+1;
           xn = x;
           break;
   case 3: xn = x+1;
           yn = y;
           break;
} /* ---- switch i ---- */
```

Let us assume that the walker performed N steps. This constitutes one realization of a random walk. We may now be interested in computing properties of such a walk. From just one realization we cannot draw any conclusion since the walk may be atypical. We need to generate many walks, calculate for every walk the desired property and then average over the results. The point which we want to make is that the generation of the samples, i.e., all the realizations of random walks are generated independently. Let A_i be the observable property computed for the i-th realization of a random walk. We define the average, or expectation value for the observable A, denoted by $<A>$, as the arithmetic mean over all A_i

$$<A> \quad = \frac{1}{n}\sum_{i=1}^{n} A_i \qquad , \tag{1.1}$$

just as we did for the random numbers to calculate the first moment of the distribution. The distribution of the random walks is uniform.

One property we may want to compute is the *end-to-end distance* R_N of a walk of length N. The end-to-end distance is the Euclidean distance between the starting point and the end point of a walk and define

$$R_N^2 =< [x(0) - x(N)]^2 + [y(0) - y(N)]^2 > \qquad . \tag{1.2}$$

Here again the average means that we take all walks that we generated into account and all have the same statistical weight. But what is the error in the quantity. For this we need to compute the fluctuations or the expected deviation from the mean value

$$\mathit{\Delta}(R_N, n) =< R_N^2 > - < R_N >^2 \qquad . \tag{1.3}$$

The error depends on the number of observations (n) and the length of the walk (N). We may be inclined to think that the error tends to zero if we make the length of the walks longer and longer, suggesting that the length N of the walk has the same meaning as the volume of a system. Recall that for an infinite system a single observation suffices to obtain the observable.

Note that the error for the end-to-end distance is computed as the second central moment, which we defined earlier. Our notation of the angular bracket for the average quantities then implies that there is distribution for the values of the end-to-end distance and the radius of gyration.

For the simple random walk it is easy to show that the end-to-end distance scales as

$$R_N \propto N^{\nu} \propto N^{1/2} \qquad . \tag{1.4}$$

Let us recapitulate our findings so far. From the random numbers we obtained a configuration, here a random walk. This configuration was generated independently from previous configurations to make a sample of configurations. We call this a sample generated by the *simple sampling* method.

At this point it is clear why we need uncorrelated random numbers. We need the absents from correlation to generate a true random walk. Every correlation in the random numbers must appear as a bias in the walks. This bias carries over to the observables computed form a simulation.

We modify the above random walk model by requiring the walk avoids itself. Not only is the random walk not allowed to immediately fold back onto itself, but also all sites which are occupied by the walk can not be accessed again. While in the pure random walk model all steps are independent of each other, here the future steps, with this modification, depend on the past. The walk immediately after the first step builds spatial correlation. A random walk with this modification is called a *self-avoiding random walk* (SAW). In the following we have written down a simple sampling program for the self-avoiding walk model.

Simple Sampling of Self-Avoiding Random Walks

Example 5.2

In this example of program code we show the main loop of a simple sampling algorithm of self-avoiding random walks. The algorithm to perform the basic steps of the walker is the same as for the random walk. To implement the self-avoidance, we have an array (`w`) which mimics the underlying lattice.

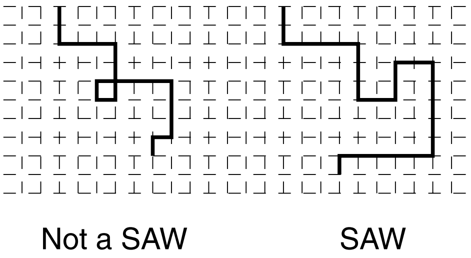

Figure 1.2.

Each site the walker goes to will be labeled by the number the walker is assigned to. This allows several walkers to travel the lattice without resetting it. At each step the algorithm checks in the array `w` whether the walker has visited the site. If so, the walk terminates signalled by the flag `occupancy`. For a full walk we calculate the radius of gyration.

```
while ( sample < sample_size ) {

  /* ==== Reset the walker to the origin ==== */

  w[0][0]   = xc;
  w[0][1]   = yc;
  x         = xc;
  y         = yc;
  l         = 0;
  occupancy = 0;
  walk++;

  return_code = r250( N,ran,mf);

  while ( (l < N)  &&  (occupancy == 0) )  {
    d  = ran[l] * 4;
    switch (d) {
        case 0:  x++;
                 break;

        case 1:  y++;
                 break;
```

```
      case 2:  x--;
               break;

      case 3:  y--;
               break;

  }

  if ( ( x < 0 ) || ( x == L ) || ( y < 0 ) || ( y == L ) ) {
      /* Random walker not on the lattice */
      exit(-1);
  }

  if ( g[x][y] < walk ) {
      g[x][y] = walk;
      l++;
      w[l][0] = x;
      w[l][1] = y;
      occupancy = 0;
  }
  else {
      occupancy = 1;
  }
}

/* ==== Now check if a SAW was generated. If yes, then ==== */
/* ==== we do the analysis, else we must try again     ==== */

if ( l == N ) {
  /* ---- we can compute the end-to-end distance etc. ---- */

  x = xc - w[N-1][0];
  y = yc - w[N-1][1];
  end_to_end += x*x + y*y;

  cmx = 0;
  cmy = 0;
  for (i=0; i < N; i++) {
    cmx += w[i][0];
    cmy += w[i][1];
  }
  cmx /= N;
  cmy /= N;
```

```
    rad = 0;
    for (i=0; i < N; i++) {
        x    = cmx - w[i][0];
        y = cmy - w[i][1];
        rad += x*x + y*y;
    }
    radius_of_gyration += rad;

    sample++;
  }
}
```

Performing the simple sampling simulation it becomes immediately evident that we have a problem with the simple sampling technique for the self-avoiding random walk model. As we increase the number of steps the walker should travel, it becomes harder and harder to find a walk. In almost all cases the walk terminates earlier because there is a violation of the self-avoiding condition! This shows that the simple sampling, even though being the simplest and perhaps even most powerful method has clear limitations.

The way out of the disastrous dependence of the probability to generate a walk of length N, once N is large, is to start with a walk that fulfils the requirement of self-avoidance. We can then generate a new walk from the already present one. We follow up on this idea when we discuss the importance sampling techniques.

Lets assume that we have solved the problem of generating large self-avoiding random walks. What we will find is that this type of a random walk does not fill space because of its spatial correlation. The end-to-end distance scales as

$$R_N \propto N^{\nu} \propto N^{0.59} \qquad . \tag{1.5}$$

The fractal dimension d_f, is equal to 1.69, i.e., less than the space dimension!

1.1.2 Correlated Sampling

One of the key features of the simple sampling is that every state is generated completely independent from the previous state. For the examples of percolation and random walk that worked nicley. For the self avoiding random walk we saw that this methods fails. Assume that we have generated a state and lets keep the self-avoiding random walk as our example. Lets say the generated state would be a straight line. The key idea is to develop a new conformation of the walk incorating the, self avoiding walk condition, from the existing conformation. One possible solution is the slithering snake algorithm.

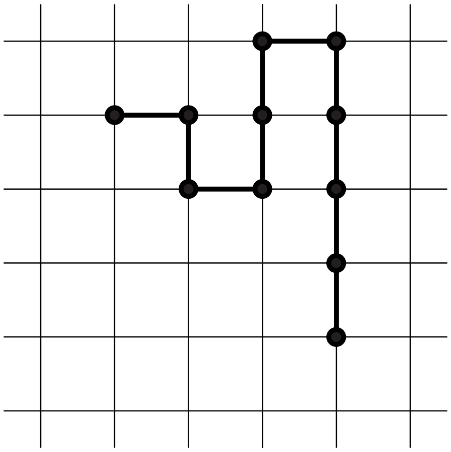

Figure 1.3. Self avoiding random walk

Algorithm: Slittering Snake

1. Assume that we have generated a random walk
2. Choose one of the end points and delete this point
3. Add the point to the opposite end choosing a random direction

The sequence of conformations of chains that we generate is $\underline{x}_1, \ldots, \underline{x}_N$ and is correlated. It follows that the probability to get the above sequence is

$$\begin{aligned} P(\underline{x}_1, \underline{x}_2, \underline{x}_3, \ldots, \underline{x}_n) &= P(\underline{x}_2, \ldots, \underline{x}_n | \underline{x}_1) P(\underline{x}_1) \\ &= P(\underline{x}_3, \ldots, \underline{x}_n) P(\underline{x}_2, \underline{x}_1) P(\underline{x}_1) \end{aligned}$$

and by recursion

$$P(\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n) = P(\underline{x}_1) \prod_{i=1}^{n} P(\underline{x}_i | \underline{x}_{i-1}) \tag{1.6}$$

1.1.3 Sampling a Distribution, the Partition Function

So far we concerned ourselves with the generation of configurations using the simple sampling technique and started to venture into the correlated sampling. For this we needed, at least for the examples presented here so far, only uniformly distributed random numbers. This situation arose because all the configurations of the examples had the same probability or the same weight. Every configuration counted the same in the averaging process irrespective of the nature of the configuration. Let us turn to a situation were we are given

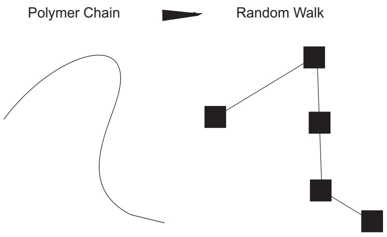

Figure 1.4.

a Hamiltonian $\mathcal{H}$. The Hamiltonian depends on the variables $\underline{x}$ that describe the degrees of freedom of the system. These can be the position of particles in space, angles, spin orientations etc. If the energy is conserved all states $\underline{x}$ have the same a priori probability. We could thus simulate such a system generating independent configurations with the fixed energy E. This immediately raises the questions: How simple is it to generate a configuration with given energy and would it be better once we have a configuration with given energy to change the configuration leaving the energy invariant? The idea to generate one configuration from the other is appealing. Indeed in the previous chapter we have learned methods to generate a new configuration from the immediate predecessor by integrating the equations of motion leaving the energy invariant!

Let us take again a look at the random walk. We may look at the random walk also as a model for a polymer chain. Each site of chain corresponds to a monomer unit, or an atom of the polymer chain and the edge connecting two sites corresponds with a bond in the polymer chain. Schematically this is shown in Figure 1.4.

We could displace a monomer unit from its original position to obtain a new configuration or conformation of the chain, once all monomers were given the chance for a displacement. This approach also adds to the sampling a time dimension. The polymer, or in general a configuration, evolves from an initial configuration. The time evolution of the configuration is then governed by the method to update the configuration. However, as we will derive in the next section, this evolution is not entirely stochastic. It is governed by a master equation giving meaning to the notion of time in a probabilistic simulation.

Let us come back to the probabilities for configurations. If all configurations with a given energy E have the same probability, then we can write down the sum or the integral over all possible states under the constraint of a fixed energy

$$Z = \int_{\Omega} \delta(\mathcal{H}(\underline{x}) - E) d\underline{x} \qquad . \tag{1.7}$$

The integral extends over all possible configurations the system can attain. The space of all these states is called the *configuration space* Ω or also called the state space. The *partition function* Z has all the information needed to describe the statistical mechanical behaviour. What we have written down is the partition function for the *micro-canonical ensemble.*

If the energy is not a conserved quantity but the temperature T is held fixed, the distribution of the states of the system, i.e. the probability for the occurrence of a configuration in configuration space, is governed by the *Boltzmann distribution*

$$P(\underline{x}) \propto e^{-\mathcal{H}(\underline{x})/k_B T} \tag{1.8}$$

and the partition function is given by

$$Z = \int_{\Omega} e^{-\mathcal{H}(\underline{x})/k_B T} d\underline{x} \qquad . \tag{1.9}$$

This is the partition function for the *canonical ensemble.* k_B is the Boltzmann constant.

Each state of the system occurs with a specific probability. If we want to perform a simulation of a system governed by a Hamiltonian in the canonical ensemble (constant temperature) we need to devise a method to generate states that follow the Boltzmann distribution. This is much like the generation of random numbers. For these we required that they follow the uniform distribution. Here the distribution is non-uniform and has a non-trivial form.

We are then required in a simulation to generate states or configurations which follow a distribution other than the uniform. To pave the way to the Monte Carlo method, let us see how we can generate a distribution using uniformly distributed random numbers. Let

$$f(x) \propto \exp\{-G(x)\} \tag{1.10}$$

be the distribution that we want to generate. $G(x)$ is some function in the open interval $(0,1)$. We obtain the distribution by devising an acceptance/rejection mechanism based upon uniformly distributed random numbers. For this we draw a random number from a uniform distribution. We rescale this number such that the number is in the interval over which G is defined. Let $t = G(x)$, where x is the number from the uniform distribution. Earlier in this chapter we considered the occurrence for a run down or a run up in a sequence of random numbers. We use a similar concept to accept or reject a drawn number x to be a sample from the distribution f. Generate k numbers from a uniform distribution where k is determined from the condition

$$t > x_1 > x_2 > ... > x_{k-1} < x_k \qquad . \tag{1.11}$$

If k is even reject x and start with a new x. If k is odd, we have a sample from the desired distribution. Using this method it is easy to generate for example random numbers that are distributed with a maxwell distribution or any other distribution.

1.1.4 Introduction into the Importance Sampling Method

How can we generate states with a specified distribution? In the previous section this essential question was answered for the case of only one variable. If the number of variables gets larger then the approach given in the previous section becomes prohibitive. We must devise a new approach that samples the states from the configuration space.

In the Monte Carlo method one calculates the expectation value for an observable A by computing the average with respect to an appropriate distribution. The distribution function is determined by the statistical mechanical ensemble. Assume that the system is described by a Hamiltonian $\mathcal{H}(\underline{x})$ with $\underline{x}$ being the degrees of freedom of the system. For the canonical ensemble with fixed temperature T, volume V and number of particles N the expectation value for an observable A is given by

$$< A > = \frac{1}{Z} \int_{\Omega} A(\underline{x}) e^{-\mathcal{H}(\underline{x})/k_B T} d\underline{x} \tag{1.12}$$

where

$$Z = \int_{\Omega} e^{-\mathcal{H}(\underline{x})/k_B T} d\underline{x} \qquad . \tag{1.13}$$

Here Ω denotes the phase space, i.e., all configurations that are available to the system. If the number of configurations of the system under the given constraints is very large, the task of evaluating (1.12) becomes formidable and one has to resort to sampling. Sampling here means that we want to pick up mainly those contributions to the integral that make the largest impact. If we were to randomly sample the available phase we would, for the most part, obtain states that give a very small contribution to the expectation value. We cannot apply simple sampling to a distribution of states that is sharply peaked. To sample the major contributions of the integrant to the integral (1.12) one constructs a *Markov chain* of states where each state or configuration is generated from the previously generated configuration:

$$\mathcal{P} : \underline{x}_0, \ldots, \underline{x}_n \tag{1.14}$$

with $\underline{x}_i \in \Omega$. The state $\underline{x}_i$ is derived from the state $\underline{x}_{i-1}$. This is not done deterministically as for the integration of motion of the equations of motion in Newtonian dynamics, but probabilistically. The state $\underline{x}_i$ followed the state $\underline{x}_{i-1}$ with a probability. There is a *transition probability*

$$W(\underline{x}_i, \underline{x}_{i-1}) \tag{1.15}$$

from one state to the other. If the system is in state $\underline{x}_{i-1}$, and it is so with a probability $P(\underline{x}_{i-1})$, then the state changes to the state $\underline{x}_i$ with the probability W. This evolution seems to be quite different from what we have learned in the preceding chapter. There, the evolution was deterministic. Given the initial conditions the entire evolution of the states of the system is determined for ever.

To make the approach more transparent we formulate again the main points. In the simple sampling method we generate the states directly from the distribution if it is simple enough to be known a priori. Here, we use a generating process. The process generates states, one from the other, ensuring that the states eventually have the correct distribution.

One may consider the Markov chain a walk or path through phase space. The label k, which sequences the states in the Markov chain, can be thought of as time. Then starting from initial state $(\underline{x}, t{=}0)$ the Monte Carlo procedure generates time ordered states

$$\mathcal{P} : (\underline{x}, t{=}0), \ldots, (\underline{x}, t{=}n) \qquad . \tag{1.16}$$

Due to the construction the states will ultimately be distributed in some way. What we need to specify is that the distribution of the states is guaranteed to be the distribution in thermal equilibrium. Before developing the necessary conditions to ensure that the transitions from one state to the next yield the correct distribution, let us dwell on the idea that we generate a trajectory in configuration space.

The definition for the time-dependent average of an observable is

$$<A(t)> \quad = \sum_{\underline{x}} A(\underline{x}) P(\underline{x}, t) \qquad , \tag{1.17}$$

where $P(\underline{x}, t)$ is the time-dependent probability density for the states. It can be shown that

$$<A(t)> \quad = \frac{1}{t} \sum_{\underline{x}} A(\underline{x}(t)) \qquad , \tag{1.18}$$

i.e., one may average over the quantity one is interested in along a *trajectory* generated by the Monte Carlo method. This is similar to the trajectories generated in molecular dynamics simulations described in the previous chapter.

The Markov chain is constructed such that the states are distributed as in thermal equilibrium, i.e., here with the canonical distribution

$$P(\underline{x}) \propto e^{-\mathcal{H}(\underline{x})/k_B T} \qquad . \tag{1.19}$$

If one has constructed transition probabilities from one state to another which give the distribution

$$d\lambda = \frac{1}{Z} e^{-\mathcal{H}(\underline{x})/k_B T} d\underline{x} \tag{1.20}$$

one obtains for the observable A

$$<A> \simeq \frac{1}{n} \sum_{k}^{n} A(\underline{x}_k) \tag{1.21}$$

i.e., again a simple arithmetic average. However, we have not performed a simple sampling but an *importance sampling*. This is because the states that we use to sample the observable are generated with the correct distribution!

Rests to define how one has to construct the transition probabilities such that we are guaranteed that the states that we generate follow the desired distribution. Let $W(\underline{x}', \underline{x})$ denote the *transition probability* to move from state $\underline{x}$ to the state $\underline{x}'$. A sufficient condition to reach equilibrium and that the states are distributed according to the desired distribution $P(\underline{x})$ is the *microscopic reversibility* or *detailed balance*

$$W(\underline{x}', \underline{x}) P(\underline{x}) = W(\underline{x}, \underline{x}') P(\underline{x}') \tag{1.22}$$

Here P is the equilibrium distribution. This condition is very much like the time reversibility in the newtonian equation of motion! It alone does not yet guarantee that the states are distributed in the correct way. We also require that every state can be reached! We do not want the available phase space to separate into disjoint parts. Further, once in a state, the probability to jump to any state must be one. That is to say, the probability to jump is a sure event.

From the above condition we can immediately derive the functional form of the transition probabilities, given the equilibrium distribution. The transition probability must have two parts. There is one part that proposes a new state. A new state $\underline{x}'$ is proposed with the probability $p^0_{x,x'}$. The other part of the transition probability deals with the acceptance of the proposed change. This is an important point. We propose a change of the state with a certain probability and the proposition is accepted with a probability that we call $a_{x,x'}$. We define the transition probability W as

$$W(\underline{x}', \underline{x}) = p^0_{x,x'} a_{x,x'} \qquad \text{if } x \neq x' \tag{1.23}$$

$$W(\underline{x}, \underline{x}) \ = p^0_{x,x} + \sum_{y \neq x} p^0_{x,x} (1 - a_{x,x'}) \tag{1.24}$$

Form this definition and the condition of the detailed balance we get

$$\frac{a_{\underline{x},\underline{x}'}}{a_{\underline{x}',\underline{x}}} = \frac{p^0_{\underline{x}',\underline{x}} P(\underline{x}')}{p^0_{\underline{x},\underline{x}'} P(\underline{x})} \tag{1.25}$$

for the only interesting case that the two states are not equal. Hence we can define $a_{x,x'}$ as a function F with the argument

$$z := \frac{p^0_{\underline{x}',\underline{x}} P(\underline{x}')}{p^0_{\underline{x},\underline{x}'} P(\underline{x})} \tag{1.26}$$

$$a_{\underline{x},\underline{x}'} = F(z) \tag{1.27}$$

to find that F needs to satisfy the following relation

$$\frac{F(z)}{F(1/z)} = z \tag{1.28}$$

Any function satisfying this relation, together with the obvious requirement that every state can be reached and that we jump somewhere, guarantees that the equilibrium distribution P will be reached and all our generated states in the simulation eventually follow this distribution.

1.1.5 Microcanonical Ensemble Monte-Carlo Method

This section introduces one possible algorithm to perform simulations at constant energy and as an example the two-dimensional Ising model is treated using this algorithm.

The kind of systems we would like to study with the microcanonical Monte-Carlo technique are those described by a Hamiltonian $\mathcal{H}$. In the microcanonical molecular-dynamics method the system has state variables $(\underline{x}, \underline{p})$ representing the generalized coordinates $\underline{x}$ and the corresponding conjugate momenta $\underline{p}$. To perpetuate the system in phase space the equations of motion are set up and solved numerically. For the microcanonical Monte-Carlo simulation we drop the kinetic energy term from the Hamiltonian. To compute properties of the system we thus cannot use the equations of motion. The approach taken is to evaluate the properties using the partition function Z. The dynamics will not reflect the true intrinsic system dynamics but the dynamics generated by a Markov chain. The configurational properties are, however, the same as those obtained by the MD method.

For a conservative system as considered here with a fixed number of particles N in a given volume V, the microcanonical ensemble distribution is expressed by a delta function, so that the partition function is

$$Z = \int_{\Omega} \delta(\mathcal{H}(\underline{x}) - E) dx \tag{1.29}$$

where E is the fixed energy of the system. The only configurations counted are those where the Hamiltonian is constrained to E. Using the partition function, quantities are computed as follows. With any observable A is associated a function $A(\underline{x})$ which depends on the state of the system. The usual assumption is that the observable A is equal to the ensemble average

$$\langle A \rangle_{NVE} = \frac{1}{Z} \int_{\Omega} A(x) \delta(\mathcal{H}(x) - E) dx \tag{1.30}$$

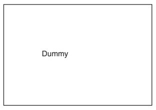

Figure 1.5. Schematic representation of a random walk on a constant energy surface in phase space

In the microcanonical ensemble, all states have a priori equal weight, as expressed by the delta function in 1.30. The general idea of the Monte-Carlo method for computing the integral on the right-hand side is to sample the available phase space of the system and carry out a summation. Similarly to the microcanonical MD technique, an algorithm must be constructed such that the system travels the constant energy surface in an ergodic manner. In the microcanonical MC method the system moves on the surface guided by a random walk (Fig 1.5) since all states have a priori equal weight. If the random walk is simple and not, for example, a self-avoiding random walk, where each state depends on the history, then a Markov chain is defined.

Suppose that somehow a state $\underline{x}$ is generated such that $W(\underline{x}) = E$. Once on the surface a sampling algorithm has to produce further states on the surface. Assume that we relax the surface restriction a little and allow for ϵ variations in the region $E - \epsilon < \mathcal{H}(\underline{x}) < E + \epsilon$ away from the surface. We may do so by introducing an extra degree of freedom [4.53-55], called a *demon*, with energy E_D, into the partition function

$$Z = \sum_{x} \sum_{E_D} \delta(\mathcal{H}(x) + E_D - E) \tag{1.31}$$

The demon plays a role similar to the kinetic energy term in molecular dynamics. It produces changes in the configuration by travelling around the system and transferring energy. Thereby the demon creates a random walk of the system on the surface. We must, however, restrict the demon's energy, otherwise it will absorb all the energy! Such a restriction may, for example, limit the demon's energy to positive values. Algorithmically the outlined procedure looks as follows.

Algorithm NVE Monte-Carlo

1. Construct a state such that $\mathcal{H}(\underline{\S}) = \mathcal{E}$.
2. Set the demon energy E_D (for example, $E_D = 0$).
3. Choose a part of the system.
4. Change the local state of the system so that $\underline{x} \to \underline{x}'$
5. Calculate the energy change produced, i.e., $\varDelta\mathcal{H} = \mathcal{H}(\underline{x}') - \mathcal{H}(\underline{x})$
6. If the energy is lowered, accept the change, set $E_D \leftarrow E_D - \varDelta\mathcal{H}$ and count $\underline{x}'$ as a new configuration. Return to step 3.
7. Otherwise, accept the change only if the demon carries enough energy, i.e., $E_D - \mathcal{H} > 0$. In this case $E_D \leftarrow E_D - \varDelta\mathcal{H}$ and count $\underline{x}'$ as a new configuration.
8. Return to Step 3.

The algorithm guarantees with Steps 6 and 7 that the system relaxes to thermal equilibrium. In addition, Step 7 also ensures the positivity of the demons energy.

Conceptually we may view the demon as a thermometer. Indeed, the demon can take up or lose energy as it is successively brought in contact with parts of the system. Initially, the demon has an arbitrary distribution. The system acts as a reservoir and thermalizes the demon. Ultimately the energies become Boltzmann distributed [4.56], allowing the calculation of the temperature

$$P(E_D) \propto \exp\left(\frac{-E_D}{k_B T}\right) \tag{1.32}$$

The Ising model serves as an example for application of the Monte-Carlo simulation method.

Example 5.2

The Ising model [146] is defined as follows. Let $G = L^d$ be a d-dimensional lattice. Associated with each lattice site i is a spin s_i which can take on the values $+1$ or -1. The spins interact via an exchange coupling J. In addition, we allow for an external field H. The Hamiltonian reads

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} s_i s_j + \mu H \sum_i s_i \qquad (1.33)$$

The first sum on the right-hand side of the equation runs over nearest neighbours only. The symbol μ denotes the magnetic moment of a spin. If the exchange constant J is positive, the Hamiltonian is a model for ferromagnetism, i.e., the spins tend to align parallel. For J negative the exchange is anti ferromagnetic and the spins tend to align antiparallel. In what follows we assume a ferromagnetic interaction $J > 0$.

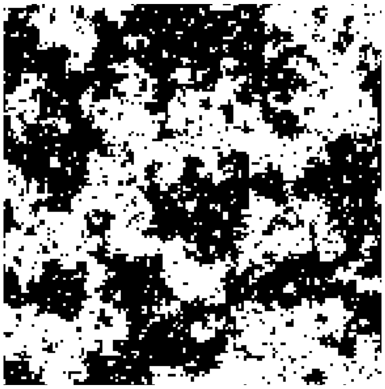

Figure 1.6.

The Ising model exhibits a phase transition (see, for example, [127] by Stanley for an introduction to phase transitions). It has a critical point T_c, where a second-order transition occurs. For temperatures T above T_c, the order parameter, i.e., the magnetization m (number of "up" spins minus number of "down" spins divided by the total number of spins), is zero in zero magnetic field. For temperatures T below T_c, there is a two-fold degenerate spontaneous magnetization. The phase diagram for the model is displayed schematically in Figure 1.7.

To calculate, for example, the magnetization of the three-dimensional model we can use the microcanonical Monte-Carlo method. The magnetization will be a function of the energy. However, with the distribution of the

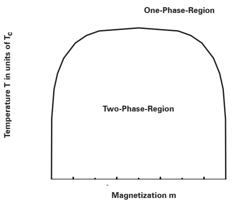

Figure 1.7. Schematic phase diagram of the three dimensional Ising model. M is the magnetization and T the temperature. T_c, is the critical point

demon energy we also obtain the magnetization as a function of temperature. For simplicity, we set the applied field to zero.

Let E be the fixed energy and suppose that a spin configuration $s = (s_1, ..., s_N)$ was constructed with the required energy. We set the demon energy to zero and let it travel through the lattice. At each site the demon attempts to flip the spin at that site. If the spin flip lowers the system energy, then the demon takes up the energy and flips the spin. On the other hand, if a flip does not lower the system energy the spin is only flipped if the demon carries enough energy. A spin is flipped if

$$E_D - \Delta\mathcal{H} > 0 \tag{1.34}$$

and the new demon energy is

$$E_D = E_D - \Delta\mathcal{H} \tag{1.35}$$

After having visited all sites one time unithas elapsed and a new configuration is generated. In Monte-Carlo method language the time unit is called the MC step per spin. After the system has relaxed to thermal equilibrium, i.e., after n_0 Monte-Carlo Steps (MCS), the averaging is started. For example, we might be interested in the magnetization. Let n be the total number of MCS, then the approximation for the magnetization is

$$\bar{m} = \frac{1}{n - n_0} \sum_{i \geq n_0}^{n} m(s_i) \tag{1.36}$$

where s_i is the ith generated spin configuration. Since the demon energies ultimately become Boltzmann distributed, it is easy to show that

$$\frac{J}{k_B T} = \frac{1}{4} \ln\left(1 + 4\frac{J}{\langle E_D \rangle}\right) \tag{1.37}$$

To carry out the simulation we use a simple cubic lattice of size 32^3. Initially all spins are set down. Then we select spins at random and turn them over until the desired energy is reached. From then on we proceed as developed above. Figure 1.8 shows the resulting distribution of E_D at the fixed energy E after 3000 MCS and 6000 MCS. The exact value of the temperature is $T/T_C = 0.5911$, corresponding to E. The results from the simulations are

```
       DO  200  MCS=1,MCSMAX
        DO  100  IZ=1,L
         IMZ = IM(IZ)
         IPZ = IP(IZ)
          DO      100  IY=1,L
           IMY = IM(IY)
           IPY = IP(IY)
           DO  100  IX=1,L
C
            ICI    = ISS(IX,IY,IZ)
            IVORZ  = ISIGN(1,ICI)
            IEN    = ICI * IVORZ - 7
C
            IF ( DEMON - IEN -  H * IVORZ  .LT. 0 ) GOTO 100
             DEMON = DEMON - IEN - H * IVORZ
C-------------FLIP SPIN
             M            = M - IVORZ
             ISS(IX,IY,IZ)      = ICI - IVORZ * 14
             ICH            = - 2 * IVORZ
             ISS(IM(IX),IY,IZ)= ISS(IM(IX),IY,IZ) + ICH
             ISS(IP(IX),IY,IZ)= ISS(IP(IX),IY,IZ) + ICH
             ISS(IX,IMY,IZ)      = ISS(IX,IMY,IZ)    + ICH
             ISS(IX,IPY,IZ)      = ISS(IX,IPY,IZ)    + ICH
             ISS(IX,IY,IMZ)      = ISS(IX,IY,IMZ)    + ICH
             ISS(IX,IY,IPZ)      = ISS(IX,IY,IPZ)    + ICH
 100          CONTINUE
C
C        IPTR = 10 * DEMON + 1
C        IDIST( IPTR ) = IDIST( IPTR ) + 1
        DEMAV = DEMAV + DEMON
        MAGAV = MAGAV + M
        FLDEM = FLDEM + DEMON * DEMON
C
 200       CONTINUE
```

$$T/T_c = 0.587, 3000MCS$$

$$T/T_c = 0.592, 6000MCS$$

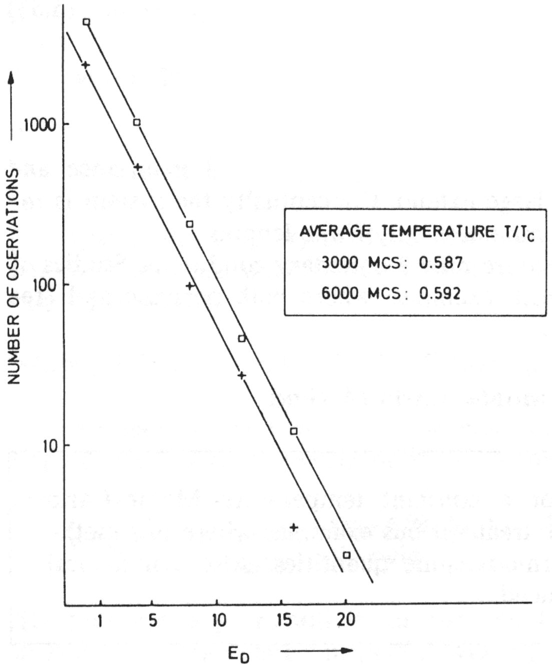

Figure 1.8. Distribution of the demon energy E_D in a microcanonical Monte-Carlo simulation of the three dimensional Ising model in zero field

A fairly large number of Monte-Carlo steps are needed before the demon reflects the real temperature. This is to be expected since the relaxation into thermal equilibrium is governed by conservation laws. Due to the energy conservation a slow approach to equilibrium results for the demon representing the temperature.

In the foregoing example no mention was made of the boundary conditions imposed on the system. How does a particle interact across the boundary? Several possible choices exist, which we group as

1. periodic boundary conditions,
2. free boundary conditions, and
3. non-standard boundary conditions.

In the third category we lump together boundary conditions which create effects that are not yet fully understood. An example falling into this class is the self-consistent field boundary condition [119, 120, 147]. Better understood in their behaviour [139] are the periodic and the free boundary conditions. The periodic boundary condition applies to a hypercubic system and was employed in the MD simulation. There this boundary condition was selected to eliminate surface effects to simulate the bulk behaviour. The same applies here because we are primarily concerned with the behaviour of the system in the thermodynamic limit. Let $L_1, ..., L_d$ be the linear dimensions of the box. For any observable A we have

$$A(x) = A(x + L_i), \quad i = 1, ..., d \tag{1.38}$$

$$L_i = (0, ..., 0, L_i, 0, ..., 0) \tag{1.39}$$

The periodic boundary condition establishes translational invariance and eliminates surface effects to a large extend. Conceptually the system is infinite; however, it can still accommodate only finite lengths. Some types of problems require mixed boundary conditions. Studies of wetting phenomena [148] furnish examples where both periodic and free boundaries are combined [139].

1.1.6 Canonical Ensemble Monte-Carlo Method

The Metropolis method for a constant temperature Monte-Carlo simulation is introduced. We treat various examples where this method is used to calculate thermodynamic quantities. Also a non-local simulation method is introduced.

In contrast to the microcanonical ensemble where all states have equal a priori weight, in the canonical ensemble some states are assigned different weights. A simple random walk through phase space is not applicable for the evaluation of observables in the (N, V, T)-ensemble. In thermodynamic equilibrium some states occur more frequently. To generate a path such that the states occur with the correct probability, a Markov process has to be constructed, yielding a limit distribution corresponding to the equilibrium distribution of the canonical ensemble.

In the canonical ensemble the particle number N, the volume V, and the temperature T are fixed. In such a situation an observable A is computed as

$$\langle A \rangle = \frac{1}{Z} \int_{\Omega} A(x) exp\left(\frac{-\mathcal{H}(x)}{k_B T}\right) dx \tag{1.40}$$

$$Z = \int_{\Omega} exp\left(\frac{-\mathcal{H}(x)}{k_B T}\right) dx \tag{1.41}$$

To develop a heat-bath Monte-Carlo method we note that in equilibrium the distribution of states is

$$P(x) = Z^{-1} \exp\left(\frac{-\mathcal{H}(x)}{k_B T}\right) \tag{1.42}$$

If we impose the detailed balance condition in equilibrium we find

$$W(x, x')P(x) = W(x', x)P(x') \tag{1.43}$$

$$\frac{W(x, x')}{W(x', x)} = \frac{P(x')}{P(x)} \tag{1.44}$$

Due to the property of the exponential, the ratio of the transition probabilities depends only on the change in energy $\Delta\mathcal{H}$ on going from one state to another

$$\frac{W(x, x')}{W(x', x)} = \exp\left[-\frac{\mathcal{H}(x') - \mathcal{H}(x)}{k_B T}\right] = \exp\left(\frac{-\Delta\mathcal{H}}{k_B T}\right) \tag{1.45}$$

We may use the form (4.41) developed in Sect.4.3 to specify the transition probability for the Metropolis MC method

The numbers w are still at our disposal. The only requirements they have to fulfil are those stated in (4.40). W(x,x') is the transition probability per unit time and the u/s determine the time scale.

Algorithm NVT Monte-Carlo

1. Specify an initial configuration.
2. Generate a new configuration x'.
3. Compute the energy change E
4. If b,
5. Compute exp(-EW/k¿ T).
6. Generate a random number R e [0, 1].
7. If R is less than exp(-b,W/k¿ T), accept the new configuration ani return to Step 2.
8. Otherwise, retain the old configuration as the new one and return ti Step 2.

At this point we see more clearly the meaning of the choice of transition probabilities. The system is driven towards the minimum energy corresponding to the parameters (N, V, T). Step 4 says that we always accept: new configuration having less energy than the previous one. Configuration which raise the energy are only accepted with a Boltzmann probability.

Example 5.3

To demonstrate an implementation of the canonical-ensemble Monte-Carlo method, we use again the Ising model already familiar to us from th previous section. The first step in constructing an algorithm for the simulation of the model is the specification of the transition probabilities froa one state to another. The simplest and most convenient choice for the actual simulation is a transition probability involving only a single spin; all other spins remain fixed. It should depend only on the momentary state of the nearest neighbours. After all spins have been given the possibility of a flip a new state is created. Symbolically, the *single-spin-flip* transition probability is written as

$$W_i($$

where W_i is the probability per unit time that the ith spin changes from s_i to $-s_i$. With such a choice the model is called the single-spin-flip Ising model [150]. Note that in the single-spin-flip Ising model the numbers of up spins $N_\uparrow$ and down spins $N_\downarrow$ are not conserved, though the total number N $= N_\uparrow + N_\downarrow$ is fixed. It is, however, possible to conserve the order parameter [116]. Instead of flipping a spin, two nearest-neighbour spins are exchanged if they are of opposite sign. This is the Ising model with so-called Kawasaki dynamics [171]. In this particular example the volume is an irrelevant parameter. The volume and the number of particles enter only through their ratios, i.e., $(V/N, T)$ are the parameters.

To proceed we have to derive the actual form of the transition probability. Let $P(s)$ be the probability of the state s. In thermal equilibrium at the fixed temperature T and field K, the probability that the i-th spin takes on the value s_i is proportional to the Boltzmann factor

$$P_{eq}(s_i) \propto exp\left(\frac{-\mathcal{H}(s_i)}{k_B T}\right) \tag{1.46}$$

The fixed spin variables are suppressed. We require that the detailed balance condition be fulfilled:

$$W_i(s_i)P_{eq}(s_i) = W_i(-s_i)P_{eq}(-s_i) \tag{1.47}$$

or

$$\frac{W_i(s_i)}{W_i(s_i)} = \frac{P_{eq}(-s_i)}{P_{eq}(s_i)} \tag{1.48}$$

With (4.48) it follows that

$$\frac{W_i(s_i)}{W_i(s_i)} = \frac{\exp(-s_i/E_i)}{\exp(s_i/E_i)} \tag{1.49}$$

where

$$E_i = J \sum_{\langle i,j \rangle} s_j \tag{1.50}$$

The derived conditions (4.59-61) do not uniquely specify the transition probability W. We have a certain freedom to choose W to be numerically efficient. At least two choices of transition probabilities are consistent with (4.61):

The Metropolis function [4.30]

$$W_i(s_i) = min\left\{\tau^{-1}, \tau^{-1} exp(-\varDelta\mathcal{H}/k_BT)\right\} \tag{1.51}$$

and the Glauber function [4.61]

$$W_i(s_i) = \frac{(1 - s_i \tanh E_i/k_BT)}{2\tau} \tag{1.52}$$

where τ is an arbitrary factor determining the time scale. Usually τ is set to unity. To simulate the physical system, for which the Hamiltonian (4.48) is a model, more closely, we could consider the factor τ to depend on parameters like the temperature.

In Sect.4.3 we described a dynamic interpretation of the MC method. The question arising is how far dynamic properties such as dynamic correlation functions are influenced by the choice of the transition probabilities. Near thermal equilibrium this choice leads only to a renormalization of the time scale [4.31]. However, for states far from equilibrium, the choice greatly influences the relaxation towards equilibrium [4.62].

In what follows we choose the Metropolis function. Having specified the transition probabilities guaranteeing the relaxation into thermal equilibrium, the essential step in the development is done. Suppose an initial configuration is specified. First a lattice site has to be selected. This can be done either by going through the lattice in a typewriter fashion, or by selecting sites at random. Then W_i is computed. Next a random number is generated to be compared with the transition probability. If the probability of a transition is larger than the random number, the transition from s_i to $-s_i$ is accepted. Otherwise the spin remains in the state s_i. The algorithm proceeds by selecting a new site. After all sites have been visited once by the typewriter method, or N choices of sites in a random fashion have been made, a new state of the system is generated. This comprises one time unit, or one Monte-Carlo step. H ow far the Monte-Carlo time, which depends on τ, corresponds to time in a physical system is still an unresolved question [4.49, 50].

Algorithmically the Metropolis MC method looks as follows:

1. Specify an initial configuration.
2. Choose a lattice site i.
3. Compute W;.
4. Generate a random number R e [0, 1].
5. If W;(s;) ¿ R, then s;   -s;.
6. Otherwise, proceed with Step 2 until N attempts have been made.

Figure 1.9 shows the results of Monte-Carlo simulations for the magnetization of the three-dimensional Ising model at various temperatures. The simulation had a duration of 1000 MCS. The first 500 steps were discarded and the magnetization averaged over the second 500 steps. The different symbols denote lattices of various sizes. To give a feeling for the computational needs, the inset shows the required execution time in seconds for one Monte-Carlo step. The time increases proportional to the system size $N = L^d$. These execution times were obtained with the progam PL4 listed in Appendix A2. That the execution time increases linearly with the system size is not true in general. Some algorithms, especially those for vector machines and parallel computers, perform in a different way (see references listed in conjunction with the discussion of the program PL4).

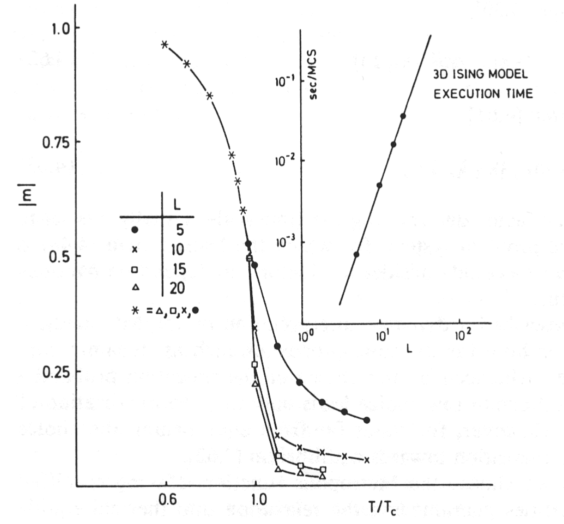

Figure 1.9. Magnetization for various temperatures and lattice sizes for the three dimensional Ising model with single spin flip. The inset shows the execution time requirements. The Monte-Carlo simulations proceeded for 1000 MCS and the averages were performed using the second 500 steps

From the observed values it is apparent that the magnetization depends on the lattice size. The effect is most dramatic near the critical temperature. For low temperatures, i.e., T much smaller than T_c the results are less sensitive to the lattice size. Indeed, the magnetization there converges to the true thermodynamic limit value rapidly. For high temperatures the magnetization is non-zero, though in the thermodynamic limit there is no spontaneous magnetization.

The behaviour of the magnetization is one typical example of finite-size effect occurring near second-order phase transitions [128, **?**] [4.40,63-66]. It can be understood by considering the correlation length. As the critical temperature is approached, the correlation length diverges, so that the finite system can accommodate only finite lengths. Hence, there will be rounding effects. In the case of first- and second-order hase transitions, the finite-size effects can be treated systematically [4.50]. Other situations require at least at the moment an ad hoc analysis.

Note that in Figure 1.9 the magnetization is plotted with absolute values. This is due to the two-fold degeneracy of the magnetization in the thermodynamic limit. For each temperature below the critical temperature there is a spontaneous magnetization $+m(T)$ or $-m(T)$. For finite systems the delta functions are smeared out to two overlapping Gaussians, and the system has a finite probability for going from a positive to a negative magnetization. It is therefore essential to accumulate the absolute values for the average.

Here we come back again to the question of ergodicity. In the Ising model an effectively broken ergodicity occurs. For a temperature below the critical temperature, the system may have either a positive or negative magnetization. During the course of a simulation both orderings are explored in a finite system if the observation time is long enough. The free-energy barrier between the two orderings is of the order $N^{(d-1)/d}$ [4.42] and the relaxation time is roughly $\exp(aN^{(d-1)/d})$. Depending on the observation time and the size of the system, the states generated by the MC simulation may explore only one ordering.

There is a difficulty with the transition probability. Suppose $\mathcal{H} >> k_BT$ or suppose $k_BT \approx 0$. Due to the exponential function, Monte-Carlo moves in such a situation occur very infrequently. The acceptance probability is proportional to $exp(-\mathcal{H}/k_BT)$! The motion through phase space is slow and an enormous number of states have to be generated in order for the system to reach equilibrium. If the system has continuous state variables, for example, in a simulation of the Lennard-Jones system, with MC methods, we can speed up the convergence. Let x_i denote the position of an atom. We generate a trial position $xxxxxxxxx$ where r is a random number from the interval $[-\delta, +\delta]$. To raise the acceptance rate of the Monte Carlo moves we simply choose δ appropriately. However, there is a danger that the constraint introduces inaccuracies.

In the case where $k_BT \approx 0$ we have to resort to other methods to speed up convergence [4.36,65,67,68]. In particular, we could develop an algorithm where only successful moves are made (cf. the discussion on the Monte-Carlo realization of the Master equation in Sect.4.3). The time intervals in such a method are then not equidistant.

Example 5.4

Up to now we have discussed examples with a discrete local state. In the Ising model the local state, i.e., the spin orientation s_i can be either $+l$ or -1. What we want to study in this example is a model with the Hamiltonian

$$\mathcal{H}(c) = \sum_i \left(\frac{r}{2} {c_i}^2 + \frac{u}{4} {c_i}^4 \right) + \frac{C}{2} \sum_{\langle i,j>\rangle} (c_i - c_j)^2 \tag{1.53}$$

where r, u, C are constants, and the local state variable c_i may assumes values between $-\infty$ and $+\infty$. This Hamiltonian is related to the coarse-grained Landau-Ginzburg-Wilson free-energy functional of Ising models [4.82,83]. We shall not be concerned with the precise relation [4.84]. We just mention that the parameters and the c_i's are the result of a coarse-graining procedure involving blocks of spins. Here we want to develop a Monte-Carlo algorithm to simulate the model given by the above Hamiltonian.

The first step we shall carry out is to scale the Hamiltonian to reduce the number of parameters. For this we consider the mean-field approximation of the solution to the model. In the mean-field approximation possible spatial fluctuations of the order parameter are neglected. Hence, the second sum on the right-hand side of (4.66) can be ignored and the partition function is

$$Z = Tr_{\{c_i\}} \exp \left[- \sum_i \left(\frac{r}{2} {c_i}^2 + \frac{u}{4} {c_i}^4 \right) \right] \tag{1.54}$$

If r is less than zero we can work out the free energy and find the order parameter

$$c_{MF} = \pm \sqrt{-r/u} \tag{1.55}$$

If r is larger or equal to zero, the order parameter is identical to zero. Next we need to know the susceptibility χ and the correlation length ξ

$$\begin{aligned} \chi(q) \; &= \chi_{MF} \left(1 + q^2 \xi^2 \right)^{-1} \\ \chi_{MF} &= (-2r)^{-1} \\ \xi_{MF} \; &= \sqrt{\frac{C}{-2r}} \end{aligned} \tag{1.56}$$

where q is the wave vector, and the above equations were obtained in linear response. Having derived the order parameter we normalize c_i with the mean-field order parameter $m_i = c_i / c_{MF}$ to find

$$\mathcal{H} = \frac{r^2}{u} \left[\sum_i \left(- \frac{m^2}{2} + \frac{m^4}{4} \right) + {\chi_{MF}}^2 \sum_{\langle i,j \rangle} (m_i - m_j)^2 \right] \tag{1.57}$$

Essentially, we are left with two parameters to vary. However, we would like to have the Hamiltonian in a slightly different form. Let us evaluate the square in the second term on the right-hand side of (4.66) and rear-range the terms yielding

$$\mathcal{H} = \sum_i \left(\frac{r+2dC}{2} {c_i}^2 + \frac{u}{4} {c_i}^4 \right) - C \sum_{\langle i,j \rangle} c_i c_j \tag{1.58}$$

where d is the dimension of the lattice. Recall that with each lattice site a local site variable c_i is associated, and that there are $2d$ nearest neighbours. Performing again a normalization

$$m_i = c_i \left[- \frac{r+2dC}{u} \right]^{-1/2} \tag{1.59}$$

we find

$$\mathcal{H} = \alpha \sum_i \left(-\frac{1}{2} {m_i}^2 + \frac{1}{4} {m_i}^4 \right) - \beta \sum_{\langle i,j \rangle} m_i m_j \tag{1.60}$$

Notice the resemblance of the Hamiltonian to the Ising Hamiltonian in the previous examples. The recognition of the formal resemblance is the major step for the development of the algorithm. Why do we not set up the algorithm directly? In simulating directly a model such as (4.66) one encounters the difficulty that the local variable m_i is not bound. One may replace the interval $(-\infty, \infty)$ by $[-a, a]$. However, serious inaccuracies result from such a choice due to the truncation. Instead of truncating the interval we choose m_i's with the single-site probability

$$P_i \propto \exp \left[-\alpha \left(-\frac{1}{2} {m_i}^2 + \frac{1}{4} {m_i}^4 \right) \right] \tag{1.61}$$

and allow all values of m, within the possible numerical accuracy. Figure 4.6 shows the distribution for two parameter values as obtained during a simulation.

Let us split the Hamiltonian into two parts [4.85, 86]

$$\mathcal{H} = \mathcal{H}_1 + \mathcal{H}_2 \tag{1.62}$$

where

$$\mathcal{H}_1 = -\beta \sum_{\langle i,j \rangle} m_i m_j, \quad \mathcal{H}_2 = \alpha \sum_i \left(-\frac{1}{2} {m_i}^2 + \frac{1}{4} {m_i}^4 \right) \tag{1.63}$$

What have we gained? Let us write the expectation for an observable as

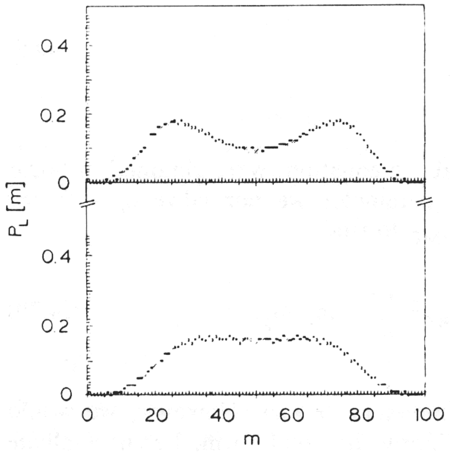

Figure 1.10. Probability distribution (4.74) of the local variable m_i as obtained by sampling

$$
\begin{aligned}
\langle A \rangle &= \frac{1}{Z} \int_{\Omega} A(m) exp\left[-\mathcal{H}(m)\right] dm \\
&= \frac{1}{Z} \int_{\Omega} A(m) exp\left[-\mathcal{H}_1(m) - \mathcal{H}_2(m)\right] dm \qquad (1.64)
\end{aligned}
$$

Because the Hamiltonian is split into two parts we may introduce a new measure (recall the procedure to introduce to the Monte-Carlo technique to reduce the variance)

$$
d\lambda(m) = Z^{-1} exp\left[-\mathcal{H}_2(m)\right] dm \qquad (1.65)
$$

and obtain

$$
\langle A \rangle = \frac{1}{Z'} \int_{\Omega} A(m) exp\left[-\mathcal{H}_1(m)\right] d\lambda(m) \qquad (1.66)
$$

With this we have succeeded in reducing (4.66) to the problem of calculating the expectation of A within the Ising model with continuous spin variables. Instead of just flipping a spin we must choose a trial m; distributed according to the measure $d\lambda A$. Clearly $d\lambda A$ can vary between zero and one; thus, a homogeneously distributed set of points in the phase space is mapped on the interval $[0, 1]$ with a density governed by the factor $exp[-\mathcal{H}_2(m)]$ by means of the cumulative distribution

$$
C(y) = \frac{\int_{-\infty}^{y} d\lambda(m)}{\int_{-\infty}^{\infty} d\lambda(m)} \qquad (1.67)
$$

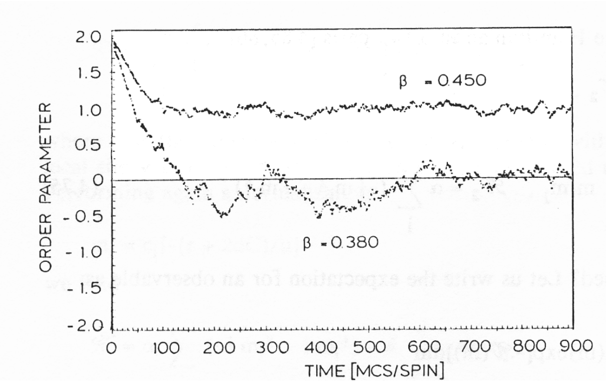

Figure 1.11. Shown is the order parameter relaxation for two values of the parameter β

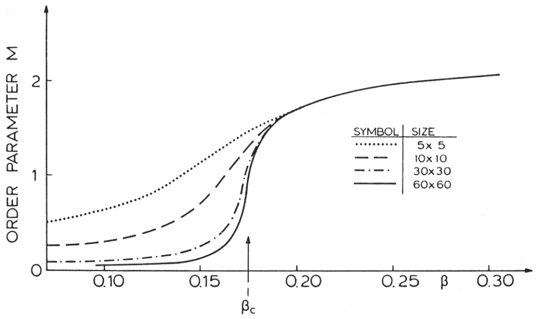

Figure 1.12. Finite size dependence of the order parameter

The trial m_i is obtained by generating first a random number $r \in [0,1]$ and then calculating the inverse of $C(r)$. The Monte-Carlo algorithm using the above looks like

1. Select a site i of the lattice.
2. Generate a random number $r \in [0,1]$.
3. Invert $C(r)$ to get a trial value for m_i.
4. Compute the change b,W¿ in the "Ising part" of the Hamiltonian
5. Generate a random number R C [0, 1].
6. If R is less than exp(-b.SF ¿) accept the trial value m;.
7. Otherwise, reject the trial m; and the old state becomes the new state.
8. Go to Step l.

Of course, there is no need to invert the functional for each trial. One may store a convenient number of $C^{-1}(r)$ in a table and interpolate for r-values not stored.

The relaxation of the system from an initial state has already been mentioned several times. Figure 1.11 displays how the order parameter relaxes into equilibrium for two values of the parameter β with fixed α. We notice that the relaxation for $\beta = 0.45$ proceeds faster than for $\beta = 0.28$. Accordingly, different portions of the initial chain have to be discarded.

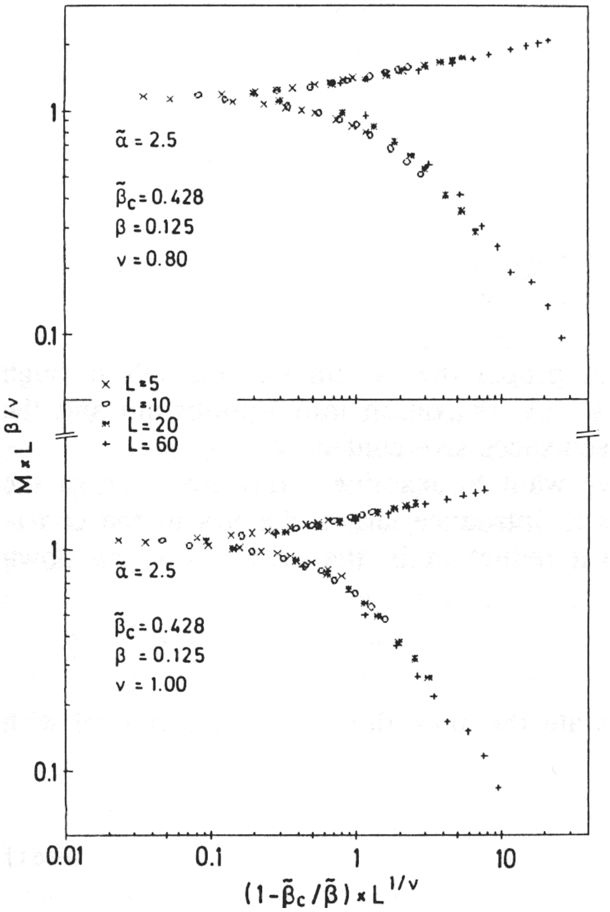

Figure 1.13. Finite size scaling plot of the order parameter

The results for the order parameter as a function of p are shown in Figure 1.12. As for the Ising model with a discrete local state, we observe a pronounced finite-size dependence. Below the critical point where the correlation length is small, the finite-size effects start to disappear. The data can, however, be collapsed to a single curve, as shown in Figure 1.13. The values or the magnetization above and below near the critical point s(hlji'. is is an example of finite-size scaling. Finite-size effects are also dramatic in the susceptibility (Figure 1.14)

$$\chi_L \propto \left\langle m^2 \right\rangle - \langle m \rangle^2 \tag{1.68}$$

At the critical point the susceptibility diverges in the thermodynamic limit. Due to the finite size of the system, the divergence is rounded. In addition, the finite size leads to a shift in the critical temperature. CI

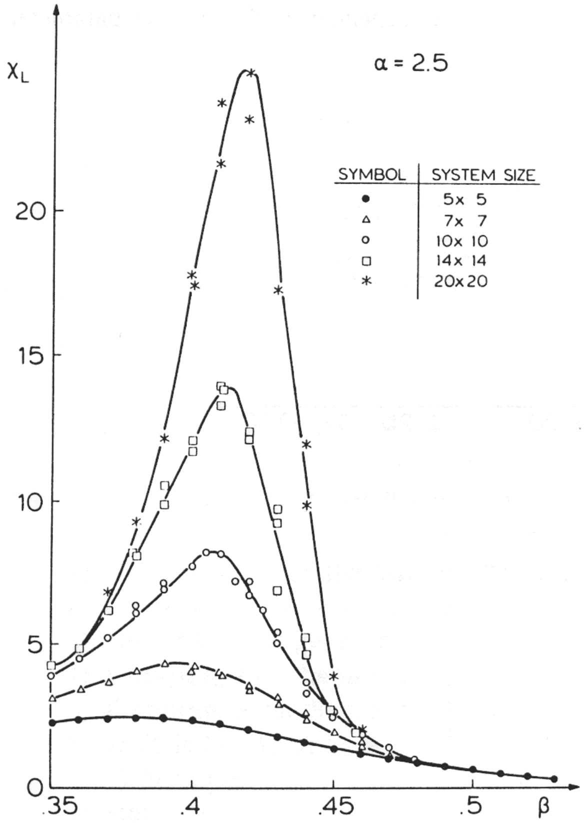

Figure 1.14. Finite size dependence of the susceptibility

1.1.7 Cluster Algorithms

So far we have only encountered Monte-Carlo simulations on lattices with simple local objects, or off-lattice simulations where the particles could be moved locally. There exists the possibility of introducing changes in the configuration on a more global level. One of the problems, especially close to second-order phase transitions, is the critical slowing down. The system there behaves in a very correlated fashion. Local changes, as they are produced, for example with the Metropolis importance sampling, cannot propel the system fast enough through phase space. The result is a very slow relaxation into equilibrium and the continued large correlation between successive configurations. In the following example we want to examine a reformulation of the Ising model which will allow us to introduce larger changes to the configurations. This in turn leads to a reduction in the critical slowing down [4.87-89].

The system for which we formulate the algorithm is the Ising model with the Hamiltonian

$$\mathcal{H}_{Ising} = -J \sum_{i,j} s_i s_j \tag{1.69}$$

which we have met before several times. This also allows an immediate comparison. In principle, the algorithm can also be formulated for the Potts model.

To understand the reasoning and the algorithm it is perhaps best to first give a quick run through the main ideas and then go into a little bit more detail.

The main idea was put forward by Fortuin and Kastelyn [179]. They proposed, and succeeded in showing, that the Ising model Hamiltonian could be mapped onto the percolation problem, which we encountered at the very beginning of this text. The mapping gives a new partition function

$$Z = \sum_{n} B(\beta, n) 2^{c(n)} \tag{1.70}$$

i.e., a combinatorial factor and contributions from the two possible cluster orientations. Instead of single spins we now have to talk about patches, or clusters of spins. Each cluster is independent of the other clusters.

We see now the advantage of such a reformulation. Instead of turning over single spins, we are able to turn over entire clusters of spins. This brings about large changes from one configuration to the other.

To perform a Monte-Carlo simulation using this idea Swendsen and Wang [154] designed an algorithm to produce the clusters and to go from one configuration to another. A configuration in the Swendsen-Wang method consists of an assignment of spin orientations to the lattice sites and an assignment of bonds between parallel spins. Consider such a configuration of a lattice with spin up and spin down. On top we have the bonds which are always broken between spins of opposite direction. Between spins of the same direction a bond can be present. An example is depicted in Figure 1.15. A cluster is defined as follows. Two up spins belong to the same cluster if they are nearest neighbours and if there is a bond between them.

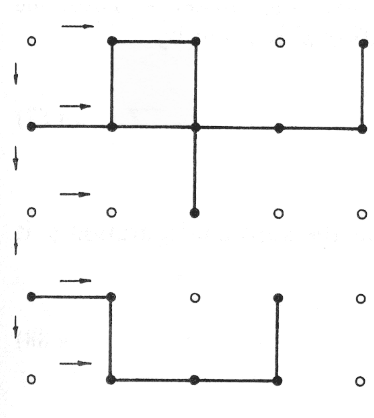

Figure 1.15. Shown is a spin configuration with one set of bonds, i.e., those between, say, the up spins

Once all clusters of up spins and all clusters of down spins have been identified we can proceed to generate a new configuration. The first step consists in choosing a new orientation for each cluster. In the model without a magnetic field, the new orientation for each cluster is chosen at random, i.e., with a probability k the orientation is reversed. After this reorientation all bonds are deleted so that only the spin configuration remains. Now the process of a bond assignment and new cluster orientation is repeated.

We shall now derive the probability with which we must assign a bond between parallel spins [4.89]. Let us derive this for the Ising model with a magnetic field

$$\mathcal{H}_{Ising} = -J \sum_{\langle i,j \rangle} s_i s_j + \mu H \sum_i s_i \tag{1.71}$$

Let $P(s) = Zexp(-\beta\mathcal{H})$ be the probability for the occurrence of the configuration s. We shall define a new Hamiltonian

$$\tilde{\mathcal{H}}_{Ising} = J \left[N_B - \sum_{\langle i,j \rangle} s_i s_j \right] + H \left[N_s + \sum_i s_i \right] \tag{1.72}$$

This Hamiltonian is zero for the ground state at zero temperature. Here N_B and N_S are the number of bonds on the lattice and the number of sites, respectively.

Denote by $N_p(s)$ the set of all bonds that lie between two parallel spins and $c(\mathit{\Gamma})$ the set of all closed bonds $[c(\mathit{\Gamma})cN_p(s)]$. Define p to be the probability of the presence of a bond and q the absence of a bond. Then we have for the probability of getting a bond configuration $\mathit{\Gamma}$)

$$P(\mathit{\Gamma}) = \sum_s P(s)P(\mathit{\Gamma}|s) \tag{1.73}$$

where the conditional probability that a bond configuration $\mathit{\Gamma}$ is generated from a spin configuration s is

$$P(\mathit{\Gamma}|s) = \delta_{\mathit{\Gamma},s} p^{c(\mathit{\Gamma})} q^{N_p(s)-c(\mathit{\Gamma})} \tag{1.74}$$

Suppose we have a spin configuration. Choose the following spin orientations for the clusters λ

- -1 with probability s = (l + exp[2pHN(A)])
- +1 with probability t = 1 - s

where $N(\lambda)$ is the number of sites associated with the cluster λ. Thus, the probability of generating a new spin configuration s' is given by

$$P(s') = \sum_{\mathit{\Gamma}} P(\mathit{\Gamma})P(s'|\mathit{\Gamma}) \tag{1.75}$$

where $P(s'|\mathit{\Gamma})$ is the conditional probability that the spin configuration s' is generated from $\mathit{\Gamma}$, i.e.,

$$P(s'|\mathit{\Gamma}) = \delta_{\mathit{\Gamma},s'} s^{\gamma^-(\mathit{\Gamma})} t^{\gamma^+(\mathit{\Gamma})} \tag{1.76}$$

with $\gamma^-(\mathit{\Gamma})$ being the number of clusters of spin $-l$ and $\gamma^+(\mathit{\Gamma})$ the number of clusters of spin +1. The total number of clusters is given by $\gamma(\mathit{\Gamma}) = \gamma^-(\mathit{\Gamma}) + \gamma^+(\mathit{\Gamma})$

We can now work out $P(\mathit{\Gamma})$ and find that

$$q = exp\,(-2\beta J) \tag{1.77}$$

Hence bonds are present on the lattice with p = l-exp(-2pJ) and clusters are reoriented with probability s. Finally the partition function for the normalized Hamiltonian is

$$\tilde{Z} = \sum_{\mathit{\Gamma}} p^{c(\mathit{\Gamma})} q^{N_B - c(\mathit{\Gamma})} \prod_{\lambda}^{\gamma(\mathit{\Gamma})} [1 + exp(2\beta HN(\lambda))] \tag{1.78}$$

Figure refChap5sw1 gives the result of a Monte-Carlo simulation of a 128^2 lattice with different applied magnetic fields. At this point we must exercise

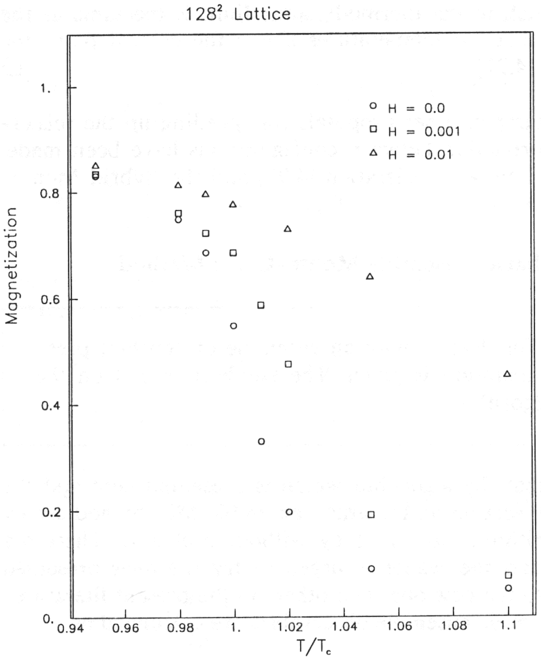

Figure 1.16. Monte-Carlo results using the Swendsen-Wang algorithm for the Ising model with a magnetic field. The system size is $l28^2$

some caution. There is no a priori argument guaranteeing us that the finite size effects are the same as those for the Metropolis or Glauber Ising model. Indeed, if we analyse the configurations using the percolation probability as the order parameter, which in the thermodynamic limit is the same as the magnetization, we find that for temperatures above the critical point the size effects are different [4.91].

Beside the above example, other proposals for speeding up the relaxation and reducing the correlation between configurations have been made. Most prominent are the Fourier acceleration [4.92] and the hybrid Monte-Carlo method [4.93] that we will meet later on.

1.1.8 Multi-Canonical Monte Carlo

1.2 Multi-Grid Monte Carlo

1.3 Quantum Monte Carlo

1.4 Path Intergal Monte Carlo

1.5 Hybrid Monte Carlo

It has become widely accepted that the Hybrid Monte Carlo (HMC) algorithm proposed by Duane *et al.* in their seminal paper [**?**] is a promising alternative to both microcanonical and Monte Carlo simulations in the context of lattice gauge theories [**?**, **?**, **?**, **?**, **?**]. HMC is a global algorithm, it has been shown to reduce critical slowing down for free field theories [**?**]. Moreover it is an exact algorithm, i.e. the ensemble averages do not depend on the step size chosen and the algorithm does not suffer from numerical instabilties due to finite step size as Molecular Dynamics (MD) algorithms do.

1.6 Problems

1. Devise an algorithm to compute the mean square displacement (x(t)) =   p ([x (t) - x (0))) for N particles in a volume with periodic boundary conditions.
2. Write an algorithm for a system of N uncoupled Brownian particles.
3. Show that the symmetrical choice of transition probabilities satisfies the Restrictions 4.11ii-iv.
4. Show that for the Ising model the temperature can be computed from the demon energy E¿ as Show also that if the magnetic field is non-zero then
5. Show that the Metropolis function and the Glauber function are mathematically equivalent.
6. Show that the Metropolis and the Glauber functions are limiting cases of a transition probability with one more parameter z (taking r' = 2)
7. The Ising model may also be simulated with a conserved order parameter. This is the so-called Kawaski dynamics [4.37]. Instead of fliping a spin, two unequal nearest neighbours are selected and exchanged if a comparison of the drawn random number and the transition probability permits this. Modify the program PL4 in Appendix A2 for the conserved order-parameter case.
8. Adapt the program given in Appendix A2 for the three-dimensional single-spin-flip Ising model to two dimensions. [The exchange coupling in two dimensions is J/k T, = 41n(l+v 2).)

9. Finite-size effects are an important consideration in simulations. For many types of problems large systems are required. Invent an algorithm for the single-spin-flip Ising model which minimizes the required storage for a simple square lattice.
10. Rewrite the grand ensemble Monte-Carlo method with symmetrical transition probabilities.
11. Self-Avoiding Random Walk. The self-avoiding random walk is a random walk where the walker does not cross its own path. At each step the walker checks if the neighbouring sites have been visited before. Of course, the walker is not allowed to retrace its steps. Quite often the walker encounters a situation where all the sites in the im- mediate neighbourhood have been visited before. The walk then terminates. Write a program which shows on a screen how a self-avoiding random walk proceeds across a two-dimensional grid.
12. An obvious extension to the Creutz algorithm is to introduce more than one demon, i.e., allow more degrees of freedom. Can you use this to parallelize the Creutz algorithm?
13. In the limit of one demon per lattice site the Creutz algorithm crosses over to the usual Metropolis Monte-Carlo algorithm. What are the pitfalls?
14. **Q2R Ising Model.** Beside the Creutz idea of performing simulations at constant energy, one can do simulations without introducing an extra degree of freedom. Take the two-dimensional Ising model. Each spin is surrounded by four nearest neighbours. Suppose the sum of the nearest neighbour spins of spin i is zero, 0 = E”” q)SJ In this case the energy is unaffected by a reversal of the central spin i. Starting from a configuration at the desired energy sweep through the lattice and reverse all spins where the energy is left invariant. Consider the ergo- dicity of the algorithm. How must one perform the sweeps? Is the algorithm ergodic at all?
15. Show that for the Q2R one is able to obtain the temperature, i.e., exp(-pJ) by sampling.
16. Can you design an algorithm where several spins are coded into the same computer word, the decision and updating are done using logical operator's? [193].
17. **Cellular Automata.** The above exercise concerns a special cellular automaton. Consider a lattice. At each lattice site there is an automaton with a given set of states S = (s¿,...s”). The states are changed by a set of rules R = (r ,...r). The rules usually depend on the states of the neighbouring automata.
18. Ergodicity of Cellular Automata. Cellular automata can be updated synchronously and asynchronously. Consider the ergodicity and develop a criterion [194].
19. **Kauffman Model.** There is an interesting class of cellular automata which is intended to simulate some biological features. The Kauffman model [195] is a random Boolean cellular automaton. The states of one

cell can be either one or zero. In two dimensions there are four nearest neighbour cells.

20. **Helical Boundary Conditions.** Consider a simple two-dimensional lattice L(i, j) with i and j ranging from 0 to n-1. For helical boundary conditions we make the identification Are the finite size effects influenced by the choice of boundary conditions? Can you give analytical arguments? Perform simulations with free, periodic, and helical boundary conditions for the Ising model and compare the results for the order parameter and the susceptibility. Which of your algorithms is the fastest?
21. Program the Swendsen-Wang algorithm for the two-dimensional Ising model (you will need a cluster identification algorithm). The magnetization and the susceptibility can be obtained from the cluster size distribution. Are the size effects the same as for the usual Metropolis algorithm [180]?
22. Derive along the lines given in the Example 4.5 the bond probability p for the Ising model without a magnetic field. Show that the probabilities derived for the case with a magnetic field reduce to the one in zero field.
23. bf Block Distribution Function. There exists another way of introducing fluctuations in the number of particles to calculate such quantities as the isothermal compressibility [4.107]. Imagine the computational box partitioned into small boxes. Let the linear box size be S, i.e., n = L/S is the number of boxes into which the computational volume has been split. In each of the boxes of sides of length S we find in general a different number of particles. For a fixed overall particle number and fixed block size we can construct the probability function P,(N) giving the probability of finding N particles inside the box of volume S . How do you compute the isothermal compressibility from the probability function P,(N)?
24. **Heat-Bath Algorithm.** Once again consider the Ising model. The heat bath algorithm to simulate the model consists of selecting the new spin orientation independent of the old one by setting where r is a random number between 0 and 1 and

    Is there a difference between the Glauber probabilities and the heat bath probabilities?
25. **Bond Fluctuation Model**

26. **Diffusion Limited Aggregation**

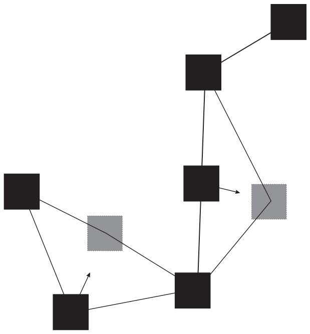

Figure 1.17.

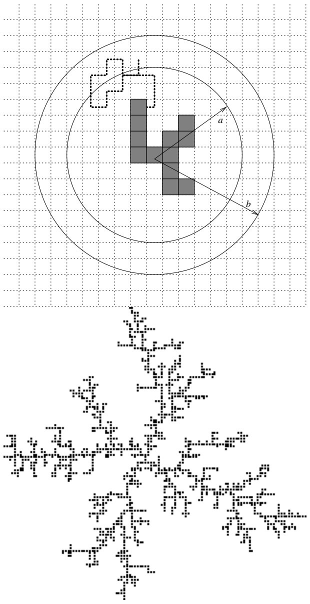

Figure 1.18.

References

1. D. Knuth, The Art of Computer Programming, Vol. 2, Wiley, New York, 1979
2. F. James, *Comput. Phys.* **60**, 329-344 (1990)
3. R.C. Tausworth, *Math Comput*, **19** 201 (1965)
   T.G Lewis and W.H. Payne *JACM*, **20**, 456-468 (1973)
   S. Kirkpatrick and E.P. Stoll, *J. Comput. Phys.* **40**, 517-526 (1981)
4. S.K. Park, K.W. Miller *Comm. ACM* **31**, 1192 (1988)
   T.G. Lewis, A.S. Goodman und J.M. Miller *IBM Sys. J.* **8**, 136 (1969)
   S.L. Anderson, *SIAM Review* **32**, 221 (1990)
   P. L'Ecuyer, *Comm. ACM* **33**, 86 (1990)
5. R.W. Hamming: Numerical Methods for Scientists and Engineers
6. Collected Algorithms of the ACM
7. J. von Neumann: Various Techniques Used in Connection with Random Digits, Collected Works, Vol. 5 (Pergamon, New York 1963)
8. G. Marsaglia, B. Narasimhan und A. Zaman, *Comp. Phys. Commun.* **60**, 345 (1990)
9. D.H. Lehmer, *Proc. 2nd Symposim on Large-Scale Digital Computing Maschinery*, 142, Harvard University Press, Cambridge, 1951
10. R.C. Tausworth, *Math Comput.*, **19**, 201 (1965)
11. G.E. Forsythe: Math. Comput. **26**, 817 (1972)
12. G.E.P. Box, M.E. Muller, G. Marsaglia: Ann. Math. Stat. **28**, 610 (1958)
13. J. Lach: Unpublished, 1962
14. E. Friedberg, J,E, Cameron; J. Chem. Phys. **52**, 6049 (1970)
15. E.B. Smith, B.H. Wells: Mol. Phys. **52**, 709 (1984)
16. D. Finchman: Daresbury Lab. Information Quaterly for Computer Simulation of Condensed Phases**17**, 43 (1985)
17. K. Binder, D.W. Heermann: Monle Carlo, in Statistical Physics, Springer Ser. Solid-State Sci., Vol .80 (Springer, Berlin, Heidelberg 1988)
18. W.G. Hoover. Molecular Dynamics (Springer, Berlin, Heidelberg 1986)
19. M.H. Kalos, P.A. Whitlock: Monte Carlo Methods, Vol.l (Wiley, New York 1986)
20. D. Stauffer, F.W. Hehl, V. Winkelmann, J.G. Zabolitzky: Computational Physics (Springer, Berlin, Heidelberg 1988)
21. D. Rapaport Comp. Phys. Rep. 9, I (1988)
22. J.L. Lebowitz, J.K. Percus: Phys. Rev. **24**, 1673 (1961)
23. J.L. Lebowitz, J.K. Percus, L. Verlet Phys. Rev **124**, 1673 (1967)
24. B.J. Alder, T.E. Wainwright: J. Chem. Phys. **27**, l208 (1957)
25. T.E. Wainwright, B.J. Alder. Nuovo cimento **9**, Suppl. Sec. 116 (1958)
26. B.J. Alder, T.E. Wainwright J. Chem. Phys. **31**, 456 (1959)
27. B.J. Alder, T.E. Wainwright J. Chem. Phys. **33**, 1439 (1960)
28. J.R. Beeler, Jr:. In *Physics of Many-Particle Syslems*, ed. by C. Meeron (Gordon and Breach, New York 1964)

29. A. Rahman: Phys. Rev. **136**, A405 (1964)
30. L. Verlet: Phys. Rev. **159**, 98 (1967)
31. K. Binder, M.H. Kalos: In *Monte Carlo Methods in Statistical Physics*, 2nd edn., ed. by K. Binder, Topics Curr. Phys., Vol.7 (Springer, Berlin, Heidelberg 1986)
32. J.P. Ryckaert, G. Ciccotti, H.J.C. Berendsen: J. Comput. Phys. **23**, 327 (1977)
33. G. Ciccotti, M. Ferrario, J.P. Ryckaert Mol. Phys. **47**, 1253 (1982)
34. J.P. Ryckaert, G. Ciccotti: J. Chem. Phys. **78**, 7368 (1983)
35. W.F. van Gunsteren, H.J.C. Berendsen: Mod. Phys. **34**, 131 l (1977)
36. D.J. Tidlesley, P.A. Madden: Mol. Phys. 42, I 137 (1981)
37. R.M. Stratt, S.L. Holmgreen, D. Chandler. Mol. Phys. 46,1233 (1981)
38. M.K. Memon, R.W. Hockney, S.K. Mitrx J. Comput. Phys. 43, 345 (1982)
39. S. Nose, M.L. Klein: Mol. Phys. 50, 1055 (1983)
40. H.C. Anderson: J. Comput. Phys. **52** , 24 (1983)
41. L.D. Landau, E.M. Lifshitz: Slaiistical Physics Vol.5, 3rd edn. (Pergamon, Oxford 1980) p.42
42. D.W. Heermann and A.N. Burkitt, **Parallel Algorithms of Computational Science Problems** Springer Verlag, Heidelberg 1990
43. W.W. Wood, F.R. Parker. J. Chem. Phys. 27, 720 (1957)
44. J.N. Shaw: Monte Carlo calculations for a system of hard-sphere ions, PhD Thesis, Duke University (1963)
45. P.P. Ewald: Ann. Physik 64, 253 (1921)
46. S.G. Brush, H.L. Sahlin, E. Teller. J. Chem. Phys. 45, 2102 (1966)
47. M. Parinello, A. Rahman: Phys. Rev. Lett. 45, l 196 (1980)
48. M. Parinello, A. Rahman: J. Appl. Phys. 52, 7182 (1981)
49. M. Parinello, A. Rahman: J. Chem. Phys. 76, 2662 (1982)
50. M. Parinello, A. Rahman, P. Vashishtrx Phys. Rev. Lett. 50, 1073 (1983)
51. G. Dahlquist, A. Bjorck: Numerical Methods (Prentice Hall, Englewood Cliffs, NJ 1964)
52. J. Stoer, R. Bulirsch: Eiirfhrung in die Numerische Mathematik II (Springer, Berlin, Heidelberg 1973)
53. A. Norsieck: Math. Comput. 16, 22 (1962)
54. C.W. Gear. Numerical Initial Value Problems in Ordinary Differenlial Equalions (Prentice Hall, Englewood Cliffs, NJ 1971)
55. C.W. Gear. Report ANL 7126, Argonne National Lab. (1966)
56. C.W. Gear. Math. Comput. 2l, 146 (1967)
57. D. Beeman: J. Comput. Phys. 20, 130 (1976)
58. S. Toxvaerd: J. Comput. Phys. 47, 444 (1982)
59. A. LaBudde, D. Greenspan: J. Comput. Phys. 15, 134 (1974)
60. D. Greenspan: J. Comput. Phys. 56, 28 (1984)
61. K. Kanatani: J. Comput. Phys. 53, 181 (1984)
62. K. Aizu: J. Comput. Phys. 55, 154 (1984)
63. K. Aizu: J. Comput. Phys. 58, 270 (1985)
64. J.H. Wilkinson: Rounding Errors in Algebraic Processes (HMSO, London 1963)
65. O. Penrose, J.L. Lebowitz: In Sludies in Statistical Mechanics, Vol.7, ed. by J.L. Lebowitz, E.W. Montroll (North-Holland, Amsterdam 1979)
66. D.W. Heermann, W. Klein, D. Stauffer. Phys. Rev. Lett. 49, 1262 (1982)
67. W.C. Swope, H.C. Andersen, P.H. Berens, K.R. Wilson: J. Chem. Phys. 76, 637 (1982)
68. L.V. Woodcock: Chem. Phys. Lett. 10, 257 (1970)
69. B. Quentrec, C. Brot J. Comput Phys. l, 430 (1973)
70. J.M. Haile, S. Gupta J. Chem. Phys. 79, 3067 (1983)

71. W.G. Hoover, D.J. Evans, R.B. Hickman, A.J.C. Ladd, W.T. Ashurst, B. Moran: Phys. Rev. A 22, l690 (1980)
72. W.G. Hoover. Physica A 18, l l l (1983)
73. W.G. Hoover. In Norrlinear Fluid Behaviour, ed. by H.J. Hanley (North-Holland, Amsterdam 1983)
74. W.G. Hoover, A.J.C. Ladd, B. Moran: Phys. Rev. Lett. 48, 3297 (1983)
75. D.J. Evans, G.P. Morriss: Chem. Phys. 77, 63 (1983)
76. D.J. Evans: J. Chem. Phys. 78, 3297 (l983)
77. D.M. Heyes, D.J. Evans, G.P. Morriss: Daresbury Lab. Information Quarterly for Computer Simulation of Condensed Phases 17, 25 (1985)
78. F. van Swol, L.V. Woodcock, J.N. Cape: J. Chem. Phys. 75, 913 (1980)
79. J.Q. Broughton, G.H. Gilmer, J.D. Weeks: J. Chem. Phys. 75, 5128 (1981)
80. D. Brown, J.H.R. Clarke: Mol. Phys. Sl, 1243 (1984)
81. S. Nose: Mol. Phys. 52, 255 (1984)
82. S. Nose, M.L. Klein: Mol. Phys. 50, 1055 (1983)
83. S. Nose: J. Chem. Phys. 81.511 (1984)
84. A. DiNola, J.R. Haak: J. Chem. Phys. 81,3684 (1984)
85. J.R. Ray: Am. J. Phys. 40,179 (1972)
86. H.C. Andersen: J. Chem. Phys. 72,2384 (1980)
87. J.M. Haile, H.W. Graben: J. Chem. Phys. 73,2412 (1980)
88. D. Adams: Daresbury Lab. Information Quarterly for Computer Simulation of Condensed Phases 10,30 (1983)
89. D. Fincham: Daresbury Lab. Information Quarterly for Computer Simulation of Condensed Phases 12,43 (1984)
90. W. Feller: An Introduction to Probability Theor and Its Applications(Wiley, New York 1968)
91. G.R. Grimmett, D.R. Stirzaker: P'.obabili'y Q]ld RQ]ldorn Processes (Oxford Univ. Press, Oxford 1987)
92. J.G. Kemeny, J.L. Snell: Fi/1i'e Markov Chai/1s (Springer, Berlin, Heidelberg 1976)
93. K. Kremer, K. Binder: Comput. Phys. Rep. 7,259 (1988)
94. P. Turq, F. Lantelme, H.L. Friedman: J. Chem. Phys. 66, 3039 (1977)
95. G.S. Grest, K. Kremer: Phys. Rev. Lett. 33,3628 (1986)
96. S. Chandrasekhar: Rev. Mod. Phys. 15, I (1943)
97. M.C. Wang, G.E. Uhlenbeck: Rev. Mod. Phys. 17,323 (1945)
98. A. Ludwig: Stochastic Differential Equations: Theory And Applications (Wiley, New York 1974)
99. S.O. Rice: Bell Syst. Tech. J. 23, (1946)
100. J.L. Doob: Ann. Math. 43,351 (1942)
101. J.L. Doob: Ann. Am. Stat. 15,229 (1944)
102. D.L. Ermak: J. Chem. Phys. 62,4189 (1974)
103. D.L. Ermak: J. Chem. Phys. 62,4197 (1974)
104. J.D. Doll, D.R. Dion: J. Chem. Phys. 65,3762 (1976)
105. T. Schneider, E. Stoll: Phys. Rev. B 13,1216 (1976)
106. T. Schneider, E. Stoll: Phys. Rev. B 17,1302 (1978)
107. W.F. van Gunsteren, H.J.C. Berendsen: Mol. Phys. 45,637 (1982)
108. M.P. Allen: Stochastic Dynamics (CECAM Workshop Report 1978)
109. M.P. Allen: Mol. Phys. 40, 1073 (1980)
110. M.P. Allen: Mol. Phys. 47,599 (1982)
111. H.J.C. Berendsen, J.P.M. Postma, W.F. van Gunsteren, A. DiNola, J.R. Haak: J. Chem. Phys. 81,3684 (1984)
112. H.C. Andersen: J. Chem. Phys. 72,2384 (1980)
113. A. Greiner, W. Strittmatter, J. Honerkamp: J. Stat. Phys. 51,95 (1988)

114. B. Dnweg, W. Paul: To be published
115. J.H. Halton: SIAM Rev. 12,4 (1970)
116. J.M. Hammersley, D.C. Handscomb: Monte Carlo Methods (Chapman & Hall, New York 1964)
117. A. Papoulis: Probability, Randoom Variables And Stochastic Processes (McGraw-Hill, Tokyo 1965)
118. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller: J. Chem. Phys. 21,1087 (1953)
119. H. Mller-Krumbhaar, K. Binder: J. Stat. Phys. 8, I (1973)
120. K. Binder: Adv. Phys. 23,917 (1974)
121. K. Binder: In Phase Transitions and Critical Phenomena, ed. by C. Domb, M.S. Green (Academic, New York 1976)
122. L.D. Fosdick: Methods Comput. Phys. 1,245 (1963)
123. I.Z. Fisher: Sov. Phys.-Usp. 2,783 (1959)
124. K. Binder: In Monte Carlo Methods in Statistical Physics, 2nd ed., ed. by K. Binder, Topic Curr. Phys., Vol.7 (Springer, Berlin, Heidelberg 1986)
125. K. Kawasaki: In Phase Transitions and Critical Phenomena, Vol.2, ed. by C. Domb, M.S. Green (Academic, New York 1972)
126. W.W. Wood: In Physics 0f Simple Liquids, ed. by H.N.V. Temperley, J.S. Rushbrook (North-Holland, Amsterdam 1968)
127. H.E. Stanley: Introduction to Phase Transitions and Critical Phenomena (Oxford Univ. Press, London 1971)
128. O.G. Mouritsen: Computer Studies 0f Phase Transitions and Critical Phenomena (Springer, Berlin, Heidelberg 1984)
129. T.L. Hill: Thermodynamics of Small Systems (Benjamin, New York 1963)
130. G.S. Pawley, G.W. Thomas: Phys. Rev. Lett. 48,410 (1982)
131. R.G. Palmer: Adv. Phys. 31,669 (1982)
132. D. Stauffer: Private communication
133. A.E. Ferdinand, M.E. Fisher: Phys. Rev. 185,832 (1969)
134. P.C. Hohenberg, B.I. Halperin: Rev. Mod. Phys. 49,435 (1977)
135. K. Binder, D. Stauffer: Adv. Phys. 25,343 (1976)
136. K. Binder: J. Stat. Phys. 24,69 (1981)
137. E. Stoll, K. Binder, T. Schneider: Phys. Rev. B 8,3266 (1973)
138. N. Y. Choi, B.A. Huberman: Phys. Rev. B 29,2796 (1984)
139. K. Binder: J. Comput. Phys. 59, I (1985)
140. B.K. Chabrabarty, H.G. Baumgrtel, D. Stauffer: Z. Phys. B 44, 333 (1981)
141. C. Kalle: J. Phys. A 17, (1984)
142. M. Creutz: Phys. Rev. Lett. 50, 1411 (1983)
143. G. Bahnot, M. Creutz, H. Neuberger: Nucl. Phys. B 235 [FSII], 417 (1984)
144. D.W. Heermann, R.C. Desai: Comput. Phys. Commun. 50,297 (1988)
145. R.C. Tolman: The Principles 0f Statistical Mechanics (Dover, New York 1979)
146. E. Ising: Z. Phys. 31,253 (1925)
147. K. Binder, H. Mller-Krumbhaar: Phys. Rev. B 7,3297 (1973)
148. J.W. Cahn: J. Chem. Phys. 66,3667 (1977)
149. K. Binder, D.P. Landau: J. Appl. Phys. 57,3306 (1985)
150. R.J. Glauber: J. Math. Phys. 4,294 (1963)
151. W. Paul, D.W. Heermann, K. Binder: J. Phys. A 22,325 (1989)
152. M.N. Barber: In Phase Transitions And Critical Phenomena, ed. by C. Domb. J.L. Lebowitz (Academic, New York 1983)
153. D.P. Landau: Phys. Rev. B 13,2997 (1967)
154. R.H. Swendsen, S. Krinsky: Phys. Rev. Lett. 43,177 (1979)
155. K. Binder, D.W. Heermann: The Monte Carlo Method in Statistical Physics, Springer Ser. Solid-State Sci., Vol.80 (Springer, Berlin, Heidelberg 1988)

156. A.B. Bortz, M.H. Kalos, J.L. Lebowitz: J. Comput. Phys. 17, 10 (1975)
157. A. Sadiq: J. Comput. Phys. 55, 387 (1984)
158. Z.W. Salsburg, J.D. Jacobsen, W. Fickett, W.W. Wood: J. Chem. Phys. 30, 65 (1959)
159. J.P. Hansen, L. Verlet Phys. Rev. 184,151 (1969)
160. K. Binder: J. Stat. Phys. 24,69 (1981)
161. K. Binder: Z. Phys. B 45,61 (1981)
162. T.L. Polgreen: Phys. Rev. B 29,1468 (1984)
163. S.K. Ma: J. Stat. Phys. 26,221 (1981)
164. Z. Alexandrowicz: J. Chem. Phys. 55,2765 (1971)
165. H. Meirovitch, Z. Alexandrowicz: J. Stat. Phys. 16,121 (1977)
166. H. Meirovitch: Chem. Phys. Lett. 45,389 (1977)
167. H. Meirovitch: J. Stat. Phys. 30681 (1983)
168. J.P. Valleau, D.N. Cord: J. Chem. Phys. 57,5457 (1972)
169. G. Torrie, J.P. Valleau: Chem. Phys. Lett. 28,578 (1974)
170. G. Torrie, J.P. Valleau: J. Comput. Phys. 23, 187 (1977)
171. K. Kawasaki, T. Imaeda, J.D. Gunton: In Perspectives in Statistical Physics, ed. by H.J. Raveche (North-Holland, Amsterdam 1981)
172. J.D. Gunton, M. San Miguel, P. Sahni: In Phase Transitions and Critical Phenomena, Vol.8, ed. by C. Domb, J.L. Lebowitz (Academic, New York 1985)
173. L.P. Kadanoff: Physics 2,263 (1966)
174. B. Freedman, P. Smolensky, D. Weingarten: Phys. Lett. 113B, 481 (1982)
175. A. Milchev, D.W. Heermann, K. Binder: J. Stat. Phys. 44,749 (1986)
176. R. Swendsen, Wang: Phys. Rev. Lett. 50,297 (1988)
177. U. Wolff: Phys. Rev. Lett. 50,297 (1988)
178. A.N. Burkitt, D.W. Heermann: to be published
179. C.M. Fortuin, P.W. Kastelyn: Physica 57,536 (1972)
180. M. DeMeo, D. W. Heermann, K. Binder: J. Stat. Phys. in press
181. J.D. Doll, R.D. Coalsen, D.L. Freeman: Phys. Rev. Lett. 55, I (1985)
182. S. Duane, A.D. Kennedy, B.J. Pendelton, D. Roweth: Phys. Rev. Lett. 2, 195 (1987)
183. G.E. Norman, V.S. Filinov: Hjgh Temp. (USSR) 7,216 (1969)
184. J.P. Valleau, L.K. Cohen: J. Chem. Phys. 72,5935 (1980)
185. D.J. Adams: Mol. Phys. 28, 1241 (1974)
186. D.J. Adams: Mol. Phys. 29,307 (1975)
187. L.A. Rowley, D. Nicholson, N.G. Parsonage: J. Comput. Phys. 17,401 (1975)
188. L.A. Rowley, D. Nicholson, N.G. Parsonage: Mol. Phys. 31,365 (1976)
189. J. Yao: PhD Dissertation, Purdue University (1981)
190. J. Yao, R.A. Greenkorn, K.C. Chao: Mol. Phys. 46,587 (1982)
191. A.Z. Panagiatopoulos: Mol. Phys. 61,813 (1987)
192. A.Z. Panagiatopoulos, N. Quirke, M. Stapleton, D. Tildesley: Mol. Phys. 63, 527 (1988)
193. H. Herrmann: J. Stat. Phys. 45, 145 (1986)
194. G. Grinstein, C. Jayaprakash, Y. He: Phys. Rev. Lett. 55,2527 (1987)
195. Kauffman. Phys. Rev. Lett. (1987)
196. M. Rovere, D.W. Heermann, K. Binder: Europhys. Lett. 6,585 (1988)