Monte Carlo simulations in statistical physics and field theory

M. Hasenbusch*

Dipartimento di Fisica dell'Università di Pisa
110, I-56127 Pisa, Italia
e-mail: Martin.Hasenbusch@df.unipi.it

Abstract

I give an introduction to the Monte Carlo simulation applied to problems in statistical physics and QFT discretised on the lattice. As example the Ising model is often used.

A Markov chain is used to generate configurations with the Boltzmann factor as weight.
To this end, various algorithms are used:
Metropolis, Heatbath, overrelaxation and cluster algorithms.

The data that are generated during the simulation must be analyzed and interpreted. A few key words: statistical error, autocorrelation times, binning, jackknife.

Based on the simulations we like to learn something about physics:

Mostly we shall study phase transitions. Questions are:
How can we determine the transition temperature? How can one determine the order of the transition? How does one determine critical exponents?

*e–mail: Martin.Hasenbusch@df.unipi.it

Contents

1 Introduction

1.1 Physics

1.1.1 Ising Model

Often we shall consider the Ising model as example. This model is a simple model of a magnet. It was proposed around 1925 by Lenz. His student Ising solved the model on a 1-dimensional lattice [Z.Phys.31 (1925) 253]. In one dimension there is no spontaneous magnetisation at any finite temperature. In general we have a D-dimensional lattice. (For simplicity we shall only consider square lattices in 2 dimensions and simple cubic lattices in 3 dimensions.) The sites are labelled by $x = (x_1, ..., x_d)$ with

$$x_i \in \{1, 2, 3, ..., L_i\} \quad , \tag{1}$$

(the lattice constant is set to one $a = 1$.) Mostly we consider periodic boundary conditions: $(x_1, .., L_i, ..., x_d)$ is the nearest neighbour of $(x_1, .., 1, ..., x_d)$. On each site of this lattice there is one degree of freedom (often called "spin")

$$s_x \in \{-1, 1\} \quad . \tag{2}$$

This can be seen as an elementary magnet. The physics question is, whether, at some temperature, these elementary magnets order and there is macroscopic order in the system. I.e. there is a spontaneous magnetisation. To discuss the statistical

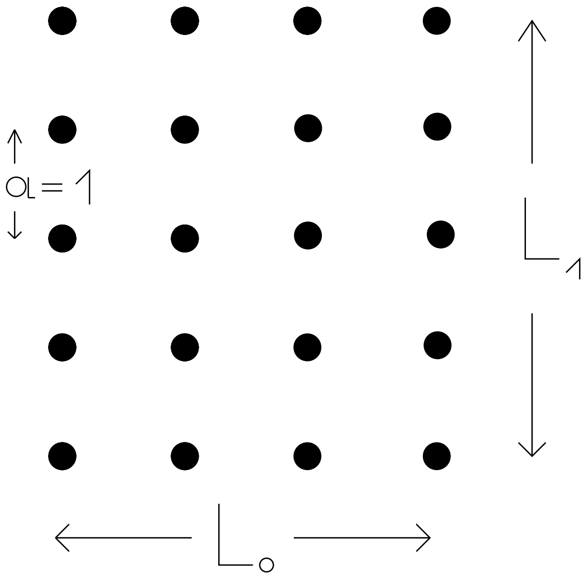


Figure 1: sketch of a 2 d lattice.

physics of the model we have to introduce a Hamiltonian:

$$H(\{s\}) = -J \sum_{<xy>} s_x s_y - h \sum_x s_x \tag{3}$$

$< xy >$ denotes a pair of nearest neighbours on the lattice. J is the coupling constant and h an external field.

In the canonical ensemble a configuration appears with the Boltzmann factor $\exp(-\beta H)$ as weight, where

$$\beta = \frac{1}{k_b T} \quad . \tag{4}$$

T is the temperature and k_b the Boltzmann constant.

The partition function is given by

$$Z = \sum_{s_{0,\ldots,0}=\pm 1} \ldots \sum_{s_{L_1,\ldots,L_d}=\pm 1} \exp(-\beta H) \tag{5}$$

In the following we shall use the abbreviation

$$\sum_{\{s\}} := \sum_{s_{0,\ldots,0}=\pm 1} \ldots \sum_{s_{L_1,\ldots,L_d}=\pm 1} \quad . \tag{6}$$

The expectation value of an observable A is given by

$$< A > := \frac{\sum_{\{s\}} \exp(-\beta H) A}{Z} \tag{7}$$

Examples of observables

Magnetisation :

$$m := \frac{1}{V} \sum_x s_x \quad , \tag{8}$$

correlation function:

$$G(x,y) := s_x s_y \quad . \tag{9}$$

More examples will be given later.

In 1944 Onsager found an exact solution of the two dimensional Ising model for $h = 0$. [L. Onsager, Phys.Rev. 65 (1944) 117]. In two dimensions (and also higher dimensions) there is a phase transition at a finite temperature. In three dimensions there is still no exact solution of the Ising model. Here we have to rely on approximations (e.g. high temperature series expansions, field theoretic methods) and (as discussed in this lecture) Monte Carlo simulations. In four and higher dimensions important properties of the model are given exactly by the "Mean-field" approximation.

Discussions of the Ising model are given in many text books like chapters 14-18 in the second edition of Kerson Huang, Statistical Mechanics , John Wiley & Sons.

The Ising model can be viewed as a model for a variety of multi-particle systems: E.g. binary mixtures. On type of particle is identified with $s = -1$ while the other is identified with $s = +1$.

By a proper choice of boundary conditions and the geometry of the lattice a large variety of phenomena can be studied:
thin films, cappilary effects, surfaces, interfaces, ...

Choosing coupling constants $J_{<xy>}$ that depend on the link, effects caused by defects of the lattice can be studied.

example

Critical exponents of the three dimensional diluted Ising model
H. G. Ballesteros, L. A. Fernandez, V. Martin-Mayor, A. Munoz Sudupe, G. Parisi, J. J. Ruiz-Lorenzo
cond-mat/9802273

Choosing $J_{<xy>}$ randomly (e.g. with equal probability for $+1$ and -1 or a Gaussian distribution around 0.), one obtains so called "Spin glasses".

example form the literature

On the Phase Structure of the 3D Edwards Anderson Spin Glass
E. Marinari, G. Parisi, J. J. Ruiz-Lorenzo
cond-mat/9802211

1.1.2 Other lattice models

Taking the spin as an 3-component unit vector, one obtains the classical Heisenberg model. With two components the XY model

$$H = -J \sum_{<xy>} \vec{s}_x \vec{s}_y - \vec{h} \sum \vec{s}_x \tag{10}$$

An other generalization of the Ising model are so called q-state Potts models. Here, the spin can take q different discrete states: $s_x \in \{1, 2, ..., q\}$

$$H = -J \sum_{<xy>} \delta_{s_x, s_y} \tag{11}$$

1.1.3 Field theory, Lattice QCD

1.2 Mathematics

Numerical integration of a function $f(x)$ from t_0 to t_1:

$$\int_{t_0}^{t_1} f(x) \mathrm{d}x = \lim_{n \to \infty} \frac{t_1 - t_0}{n} \sum_{i=1}^{n} f(x_i) \tag{12}$$

with

$$x_i = t_0 + \frac{i}{n}(t_1 - t_0) \tag{13}$$

The points x_i are called nodes
Errors of the numerical integration are: $O(a)$
with $a = (t_1 - t_0)/n$

Improved methods are Trapezoidal rule, Simpson rule ...
with errors : $O(a^2)$, ...

Usually, for real functions these methods work without problem. (See text books on Numerical analysis or chapter 4 of Press et. al. Numerical Recipes)

However it becomes difficult to generalize them to higher dimensions:

$$I = \int \mathrm{d}x_1 \int \mathrm{d}x_2 \; ... \int \mathrm{d}x_D \;\; f(x_1, x_2, ..., x_D). \tag{14}$$

For simplicity assume that the integration runs in each dimension from 0 to 1. The distance a between the nodes is equal in each dimension. Then the number of nodes is $n = (1/a)^D$. The computational effort is proportional to the number of nodes. Lets take $a = 0.1$ Then the number of nodes is $n = 10^D$.

Assume, that it costs $10^{-7}s$ to evaluate the function for one node. then we get the CPU times of table 7.

This rough estimate shows that the numerical integration with equidistant nodes becomes un-feasible for dimensions larger than about 12.

High dimensional integrals are typical for statistical physics and the path integral formalism in quantum field theories.

Table 1: CPU-time as a function of the dimension of the integral. Details are given in the text.

D	CPU-time
1	10^{-6} s
2	10^{-5} s
3	10^{-4} s
4	10^{-3} s
...	...
10	16.7 min
11	2.8 h
12	1.16 days

2 Monte Carlo Integration

A way out of the problem is given by the Monte Carlo integration. Instead of regularly distributed nodes one chooses them "randomly" in the integration range.

For a 1-dimensional integral one gets:

$$\int_{t_0}^{t_1} f(x)\mathrm{d}x \approx \frac{t_1 - t_0}{N} \sum_{i=1}^{N} f(x_i) \quad , \tag{15}$$

where the random numbers x_i are uniformly distributed in the interval $t_1 - t_0$. The statistical error of the method is proportional to $1/\sqrt{N}$. It is important to note that this relation between the error and the number of nodes also holds in higher dimensions.

2.1 Random numbers

Let us discuss how we can get a random number with a certain distribution. Let us give some examples first

- Dice. With equal probability we get a number out of $\{1, 2, 3, 4, 5, 6\}$. A single result is not predictable.
- Lottery. (hopefully random)
- Radioactive decay.

One could think of coupling a physical system, as discussed above, with the computer. However, for our purposes this is typically too slow.

Therefore, usually so called pseudo random numbers are used in the Monte Carlo integration. These numbers are generated with a completely deterministic rule. However, looking at the numbers, it is hard to guess the underlying rule.

Often so called "linear congruential" generators are used. They are a deterministic sequence given by

$$I_{j+1} = (aI_j + c) \bmod m \quad . \tag{16}$$

where a,c and m are integer. For sure $0 \leq I_j < m$ for all j.

An important property of a generator is its period. The period p says when the generator repeats itself. I.e. p is the smallest number such that $I_j = I_{j+p}$. One should chose a, c and m such that one obtains a long period.

In the optimal case the period is given by $p = m$. Indeed a and c can be found such that $p = m$ holds. These a and c are completely classified: (Hull and Dubell, SIAM Review 4 (1962) 230)

A linear congruential generator with the modulus m has a period m if and only if:
(i) c and m have no common prime factor.
(ii) $a - 1$ is a multiple of each prime factor of m.
(iii) $a - 1$ is a multiple of 4, if m is a multiple of 4.

Linear congruential generators with $m = 2^n$ can be implemented very efficiently using bit-wise operations.

For this special case, (i)-(iii) are satisfied for $a = 4k + 1$ for any positive integer k and c odd.

Examples for a, c and m are
drand48 in the C-library:
a = 0x5DEECE66D (base 16) = 0273673163155 (base 8)
c = 0xB (base 16) = 013 (base 8)
In Dezimal numbers $a = 25214903917$, $c = 11$, $m = 2^{48}$. The conditions (i) -(iii) hold; Therefore the period of drand48() is $m = 2^{48}$.

G05CAF from the NAG-library:
$a = 13^{13}$, $c = 0$, $m = 2^{59}$
The period of G05CAF is $p = 2^{57}$.

The generator produces an integer number between 0 and $m-1$. I_j is multiplied by $1/m$ to obtain random numbers in $[0, 1)$.

Linear congruential generators are not the only generators that are used. Others are "Fibonacci" or "shift register" generators.

A very safe choice (but also relatively slow) choice is the ranlux generator proposed by M. Lüscher that has entered the GNU scientific library; For the definition and a discussion of its properties see
Martin Lüscher, A PORTABLE HIGH QUALITY RANDOM NUMBER GENERATOR FOR LATTICE FIELD THEORY SIMULATIONS, Comput.Phys.Commun.79:100-110,1994, hep-lat/9309020.

A long period is, unfortunately, not the only requirement on a good random number generator. The study of the quality of these generators is a research topic in its own, see e.g.

New tests of random numbers for simulations in physical systems, I. Vattulainen, cond-mat/9411062
The RANLUX generator: resonances in a random walk test, Lev N. Shchur, Paolo Butera, hep-lat/9805017
A new test for random number generators: Schwinger-Dyson equations for the Ising, H. G. Ballesteros, V. Martin-Mayor, cond-mat/9806059.

For the purpose of this lecture "G05CAF" and "drand48" are sufficient, and we shall use them in the coming exercises.

2.2 Distributions of Random Numbers

a) **Discrete random numbers; Example: dice; lottery**
Throwing dice, the elementary events are the numbers $X \in \{1, 2, 3, 4, 5, 6\}$. ($S = \{1, 2, 3, 4, 5, 6\}$ is the sample space).

Each of these events has the probability $W(X)$:

$$0 \leq W(X) \leq 1 \quad . \tag{17}$$

For sure, one of the events occurs:

$$\sum_X W(X) = 1 \tag{18}$$

If A and B are sub-sets of the sample space with $A \cap B = \oslash$ then

$$W(A \cup B) = W(A) + W(B) \tag{19}$$

Experimentally $W(X)$ is determined be throwing the dice N times: Then

$$W(X) \approx \frac{N_X}{N} \quad , \tag{20}$$

where X has been obtained N_X times.

b) **Continuous Random numbers:**
The probability distribution $w(x)$ of a continuous random variable x can be defined based of probabilities of discrete events: If $W(a, b)$ is the probability that x lies in the interval $[a, b]$. Then the probability distribution $w(x)$ is defined such that for the interval $[a, b]$

$$W(a, b) = \int_a^b \mathrm{d}x \; w(x) \tag{21}$$

holds. The pseudo-random number generators discussed above generate (to some approximation) a uniform distribution in $[0, 1)$. I.e. $w(x) = 1$ for $x \in [0, 1)$ and $w(x) = 0$ elsewhere.

Important properties of $w(x)$ are:

$$w(x) \geq 0 \tag{22}$$

for all x and

$$\int \mathrm{d}x \;\; w(x) = 1 \;\; . \tag{23}$$

I.e.: "Something happens in each instance".

Expectation value and variance of a function f:
The expectation value is defined by

$$< f >_w := \int \mathrm{d}x \;\; w(x) f(x) \tag{24}$$

The variance

$$\begin{aligned} \sigma_w^2(f) \;\; &:= \;\; \int \mathrm{d}x \;\; w(x)\;(f(x) - \; < f >_w)^2 \\ &= \;\; \int \mathrm{d}x \;\; w(x)\; f(x)^2 - \; < f >_w^2 \\ &= \;\; < f^2 >_w \; - \; < f >_w^2 \;\; . \end{aligned} \tag{25}$$

2.3 Generating distributions different from the uniform one

The generators discussed above give a uniform distribution in $[0, 1)$. Often one needs something else (e.g. Gaussian distribution).

A simple method to obtain a new distribution is to map a random variable x_j using a function $y_j = f(x_j)$. The probability distribution of y is given by

$$\tilde{w}(y) = w(x) \; \left| \frac{\mathrm{d}x}{\mathrm{d}y} \right| \;\; . \tag{26}$$

The first example are random numbers with a uniform distribution in the interval $[a, b)$. Such a distribution can be obtained by the mapping $y = a \; + \; (b - a) \; x$, where x is uniformly distributed in $[0, 1)$.

Our second, more interesting example is an exponentially decaying distribution in $[0, \infty)$: $\tilde{w}(y) = \exp(-y)$. A random variable that is uniformly distributed in $(0, 1]$ is mapped by $y = -\ln(x)$. Since $x = \exp(-y) = -dx/dy$ we get indeed the desired result.

Not all desired distributions can be obtained this way. A more general (and more expensive) method is the rejection method:

$\tilde{w}(y)$ is the desired distribution. Now we take a distribution $f(y)$ that can be generated with the method discussed above. Let us choose a const c such that

$$c f(y) \geq \tilde{w}(y) \tag{27}$$

for all y.

Now we generate a $\tilde{y}$ with a probability distribution $f(y)$. This $\tilde{y}$ is taken as y_j with a probability

$$A(y) = \frac{w(y)}{cf(y)} \tag{28}$$

Implementation of the "rejection method" in C:

(Simple example:
f(x): uniform distribution in $[0, 1)$.
desired distribution: $w(x) = 2x$ in $[0, 1)$.

```
double reject()
  {
  double pro;
/* : */
  do
    {
    pro=drand48();
    }
/* rejection-step;  a =w(pro)/(2 f(pro)) = pro */
  while(pro <  g05caf());
  return pro;
  }
```

2.4 Monte Carlo Integration

Lets have a closer look at the integration method discussed in the introduction

$$I = \int_a^b \mathrm{d}x \; g(x) \;=\; \int_a^b \mathrm{d}x \; w(x) \; f(x) \; = < f >_w \quad , \tag{29}$$

where $w(x)\; f(x) \;=\; g(x)$ in the interval $[a, b)$. The expectation value $< f >_w$ can be approximated by the average over a finite number (N) of random numbers

$$< f >_w \approx \bar{f}_N := \frac{1}{N} \sum_{i=1}^{N} f(x_i) \quad , \tag{30}$$

where $w(x)$ is the distribution of the random numbers x_i.
How large is the error? I.e. how large is the difference $< f > - f_N$?

2.5 Statistical error

More precisely we should think of the distribution of an ensemble of Monte Carlo integrations using N random numbers each.

The variance of $\bar{f}_N$ is given by

$$\sigma^2(\bar{f}_N) := \lim_{M\to\infty} \frac{1}{M} \sum_{j=1}^{M} \left(\langle f \rangle - \bar{f}_{N,j} \right)^2 \tag{31}$$

As error of the Monte Carlo integration one usually quotes $\sqrt{\sigma^2(\bar{f}_N)}$. The problem is that we like to perform only a single integration and not many. Therefore we need a relation between $\sigma^2(\bar{f}_N)$ and the variance $\sigma^2(f)$ of f itself:

$$\begin{aligned}
& \lim_{M\to\infty} \frac{1}{M} \sum_{\alpha=1}^{M} \left(\langle f \rangle - \bar{f}_{N,\alpha} \right)^2 \\
= \; & \lim_{M\to\infty} \frac{1}{M} \sum_{\alpha=1}^{M} \left(\frac{1}{N} \sum_{i=1}^{N} (\langle f \rangle - f(x_{i,\alpha})) \right)^2 \\
& \text{where } \; x_{i,\alpha} \; \text{ is the } i^{th} \text{ random number in the } \alpha^{th} \text{ integration.} \\
= \; & \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lim_{M\to\infty} \frac{1}{M} \sum_{j=1}^{M} \left(\langle f \rangle - f(x_{i,\alpha}) \right) \left(\langle f \rangle - f(x_{j,\alpha}) \right) \\
= \; & \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \delta_{i,j} \sigma^2(f) \\
= \; & \frac{1}{N} \sigma^2(f)
\end{aligned} \tag{32}$$

$\sigma^2(f)$ can be determined from our sequence of random numbers

$$\sigma^2(f) \approx \frac{1}{N-1} \sum_{i=1}^{N} (\bar{f}_N - f(x_i))^2 \tag{33}$$

The "error" of the stochastic integration is given by

$$\sqrt{\sigma^2(\bar{f}_N)} = \sqrt{\sigma^2(f)/N} \quad . \tag{34}$$

The central limit theorem says that the distribution of $\bar{f}_N$ becomes Gaußian as $N \to \infty$:

$$w(\bar{f}_N) \approx \frac{\sqrt{N}}{2\pi\sigma(f)} \exp\left(-N\frac{(< f > - \bar{f}_N)^2}{2\sigma^2(f)}\right) \quad . \tag{35}$$

Table 2: The probability $W(n)$, that the result $\bar{f}$ of the Monte Carlo Integration is in the interval $[< f > -n\sigma(\bar{f}), < f > +n\sigma(\bar{f})]$.

n	W(n)
1	0.682689...
2	0.954499...
3	0.997300...
4	0.999936...
5	0.999999...

2.6 Importance Sampling, Reduction of the Variance

Our aim is to obtained a small statistical error, of course.

This can be achieved by brute force, i.e. generating more nodes, using more CPU time.

On the other hand, the distribution $w(x)$ should be chosen such that the variance of the resulting f is as small as possible.

The ideal case is given by $w(x) \propto g(x)$. As a result, $f(x)$ is constant and the variance vanishes. In this case we have solved the integral analytically, and we need not to use the computer at all.

Let us consider an example:

$$I = \int_0^t \mathrm{d}x \;\exp(-x)\; x \tag{36}$$

The integral can be computed analytically

$$\int_0^t \mathrm{d}x \;\exp(-x)\; x = -\exp(-x)x - \exp(-x)|_0^t = 1 \;-\; (1+t)\;\exp(-t) \tag{37}$$

As our first choice we take $w_1(x) = \frac{1}{t}$ in the interval $[0,t)$. Hence

$$f_1(x) = t\;\exp(-x)\;x \quad . \tag{38}$$

As second choice we consider

$$w_2(x) = \frac{1}{1-\exp(-t)}\exp(-x) \tag{39}$$

in the interval $[0,t)$. Hence

$$f_2(x) = (1-\exp(-t))\;x \quad . \tag{40}$$

Let us compute the variance for both choices:

$$\begin{aligned}
\sigma^2_{w_1}(f_1) &= \int_0^t \mathrm{d}x\;\frac{1}{t}\;(t\;\exp(-x)\;x)^2 - < f >^2 \\
&= t\left(\frac{1}{4} - (\frac{1}{2}t^2 + \frac{1}{2}t + \frac{1}{4})\exp(-2t)\right) - < f >^2 \\
&\approx \frac{1}{4}t
\end{aligned} \tag{41}$$

for large t. That means that the variance increases with t. In the limit $t \to \infty$ the distribution is no longer defined.

$$\begin{aligned}
\sigma^2_{w_2}(f_2) &= \int_0^t \mathrm{d}x\;\frac{1}{1-\exp(-t)}\exp(-x)\;((1-\exp(-t))x)^2 \;-\; < f >^2 \\
&= (1-\exp(-t))\int_0^t \mathrm{d}x\;\exp(-x)x^2 \;-\; < f >^2 \\
&= (1-\exp(-t))\;(2-(t^2+2t+2)\exp(-t)) - \; < f >^2 \\
&\approx 1
\end{aligned} \tag{42}$$

for large t. I.e. the variance stays roughly constant as t increases. I.e. we can use $w_2(x)$ even in the limit $t \to \infty$.

Rule of thumb: $g(x)/w(x)$ should fluctuate little in the integration area.

2.7 A simple Monte Carlo experiment

Let us consider a random walk; (Brownian motion, diffusion). Let us assume that a particle starts in $(0,0)$ (let us assume a 2 dim. space). The direction is chosen randomly, each direction has the same probability. The distance between two scattering events follows the distribution $p(r) = \exp(-r)$. It follows

$$w(\vec{r}) = w(r,\phi) = \frac{1}{2\pi}\frac{1}{r}\exp(-r) \quad . \tag{43}$$

```
r=-ln(g05caf());
phi=2.*3.1415926535898*g05caf();
r1=r*cos(phi);
r2=r*sin(phi);
```

The problem can also be posed on a lattice: Lets take a 2 dimensional square lattice. In each step of the walk, the particle goes with equal probability in one of the 4 possible directions.

How far did the particle move after N steps?

From the mathematical side this question is equivalent to the problem of the statistical error discussed above. To this end we have to divide the total path after N steps $\vec{R}(N)$ by N; we identify

$$\bar{f}_N = \frac{\vec{R}(N)}{N} \quad , \tag{44}$$

where the function $\vec{f} = \vec{r}$ is the progress of the particle in one step.

For both types of random walks we get $<\vec{f}>= (0,0)$. The variance is

$$\sigma^2(\vec{f}) = \int \mathrm{d}x_1 \int \mathrm{d}x_2 \vec{f}(\vec{x})^2 \quad . \tag{45}$$

In particular

$$\sigma^2(\vec{f}) = \frac{1}{4}((1,0)^2 + (0,1)^2 + (-1,0)^2 + (0,-1)^2) = 1 \tag{46}$$

for the walk on the lattice.

$$\sigma^2(\vec{f}) = \int_0^{2\pi} \mathrm{d}\phi \int_0^{\infty} \mathrm{d}r\ \frac{1}{2\pi}\exp(-r)r^2(\cos(\phi),\sin(\phi))^2 = \int_0^{\infty} \mathrm{d}r r^2\ \exp(-r) = 2 \tag{47}$$

for the continuous version of the random walk. The mean of the square of the length of the path is

$$< |\vec{R}(N)|^2 >= N \tag{48}$$

for the walk on the lattice

$$< |\vec{R}(N)|^2 >= 2N \tag{49}$$

for the continuous walk.

Exercise:

Determine numerically $< |\vec{R}(N)|^2 >$ for all $0 < N < 1000$ for both types of random walks that have been introduced above. To this end, average over 10000 random walks that you have generated on the computer. Plot the numerical result along with the theoretical prediction.

3 Importance Sampling, Markov Chains and Metropolis

3.1 "Simple sampling"

The simplest way to compute observables in the Ising model using Monte Carlo is the so called "simple sampling".

A configuration is generated by choosing at each site independently, with equal probability either $s_x = -1$ or $s_x = +1$.

```
for(i=0;i<V;i++)
  {
  if(g05caf()>0.5)
    {
    spin[i]=1;
    }
  else
    {
    spin[i]=-1;
    }
  }
```

where V is the number of sites on the lattice. All configurations have an equal probability to be generated. Since the number of configurations is 2^V, the probability for a single configuration is 2^{-V}.

Observables are defined by

$$\langle A \rangle \; := \; \frac{\sum_{\{s\}} \exp(-\beta \; H(\{s\}))A(\{s\})}{\sum_{\{s\}} \exp(-\beta \; H(\{s\}))} \quad , \tag{50}$$

where $Z = \exp(-\beta \; H(\{s\}))$ is the partition function. recall the notation $\{s\} = (s_{1,1,...}..., s_{L_1,L_2,...})$.

Remember: As soon as the number of sites becomes more than ≈ 40, the sums in eq. (50) can not be done on the computer in a reasonable time.

Generating N configuration as discussed above, we get

$$Z \approx \frac{2^V}{N} \sum_{i=1}^{N} \exp(-\beta \; H(X_i)) \tag{51}$$

and

$$\sum_X \exp(-\beta \; H(X))A(X) \approx \frac{2^V}{N} \sum_{i=1}^{N} \exp(-\beta \; H(X_i)) \; A(X_i) \quad . \tag{52}$$

The problem of "simple sampling" is that only a small fraction of the configurations has a large Boltzmann factor. Using simple sampling (I.e. generating all configurations with the same weight), one generates a lot of configurations that are almost useless, since they contribute only little to the partition function.

Let us look at the problem in the case of the 2 D Ising model at the transition temperature and a vanishing external field. Let us consider a square lattice with $L \times L$ sites and periodic boundary conditions. In table 3 I give the fraction of configurations that is needed, to get 90% of the partition function. To obtain these numbers all configurations had been generated (which is still possible for such small lattices). We see that simple sampling becomes less efficient as the lattice size increases.

Table 3: Fraction of configurations that is needed to obtain 90% of the partition function in the 2D Ising model at the phase transition.

L	fraction
2	0.875
3	0.133
4	0.0343
5	0.00283

3.2 Importance Sampling

The solution of the problem is to imitate the physical process and to generate configurations X with a probability that is proportional to their Boltzmann weights:

$$W(X) = \frac{1}{Z} \exp(-\beta \, H(X)) \tag{53}$$

The problem with that is that the normalization $1/Z$ is unknown. Therefore it is not possible to generate configurations ad hoc with the correct distribution.

This problem can be overcome by using a stochastic process. Here only ratios of probabilities are needed:

$$\frac{W(X)}{W(Y)} = \frac{\exp(-\beta \, H(X))}{\exp(-\beta \, H(Y))} \ , \tag{54}$$

which can be computed easily.

In classical physics the generation of the configurations is done by the dynamics of the particles following the equations of motion. In the Monte Carlo simulation we also use a kind of dynamics; i.e. we shall use a temporal sequence to generate the configurations with the Boltzmann weight. However we will not use the equations of motion, but some artificial, stochastic process.

In order to generate the configurations with a probability that is proportional to the Boltzmann factor certain conditions have to be satisfied.

3.3 Markov chain

A Markov chain of first order is characterised by the fact that the probability to generate an event (here configuration) Y that follows the configuration X, depends only on X and not on events (configurations) before X.

The Markov chain of **first order** is completely characterised by the transition probability $W(Y,X)$ from a configuration X to Y. Often one finds the notation $W(X \to Y)$ in the literature. The motivation for our notation is the following: For a finite number of events, the transition probability $W(Y,X)$ can be viewed as a matrix. Let us denote the probabilities at a time t is the Markov chain by $V_t(X)$. Then a step in the chain is given by

$$V_{t+1}(Y) = \sum_X W(Y,X) V_t(X) \quad . \tag{55}$$

The Markov chain has to satisfy certain conditions in order to generate the Boltzmann distribution:

i) normalization:

$$\sum_Y W(Y,X) = 1 \tag{56}$$

for all X.

ii) Ergodicity:

$$W(Y,X) > 0 \tag{57}$$

for all X and Y. I.e. all configurations can be reached.

iii) "Stability":
The Boltzmann distribution is an eigenvector of $W(Y,X)$ with eigenvalue 1:

$$\sum_X W(Y,X) B(X) = B(Y) \tag{58}$$

for all Y. Where $B(Y) = \frac{\exp(-\beta H(Y))}{Z}$.

If i,ii,iii are satisfied:
Assume a start distribution $V_0(X)$.
The chain is continued by $V_{t+1}(X) = \sum_Y W(X,Y) V_t(Y)$.
Then in the limit of large t we get the Boltzmann distribution:

$$\lim_{t \to \infty} ||V_t - B|| = 0 \tag{59}$$

with

$$||V_t - B|| := \sum_X |V_t(X) - B(X)| \ . \tag{60}$$

Proof:

$$\begin{aligned}
& \sum_X \ |V_{t+1}(X) \ - \ B(X)| \\
= & \sum_X \ |\sum_Y \ W(X,Y) \ V_t(Y) \ - \ B(X)| \\
= & \sum_X \ |\sum_Y \ W(X,Y) \ (V_t(Y) \ - \ B(Y))| \\
& \text{With } \tilde{W}(X,Y) = W(X,Y) - W_{min} \geq 0 \text{ and } W_{min} > 0 \\
= & \sum_X \ |\sum_Y \ \tilde{W}(X,Y) \ (V_t(Y) \ - \ B(Y)) \ + \ W_{min} \sum_Y (V_t(Y) \ - \ B(Y))| \\
= & \sum_X \ |\sum_Y \ \tilde{W}(X,Y) \ (V_t(Y) \ - \ B(Y))| \\
& \text{With the triangular inequality:} \\
\leq & \sum_X \ \sum_Y |\tilde{W}(X,Y) \ (V_t(Y) \ - \ B(Y))| \\
& \text{Da } \tilde{W}(X,Y) \geq 0 : \\
= & \sum_X \ \sum_Y \ \tilde{W}(X,Y) \ |(V_t(Y) - B(Y))| \\
& \text{With property i):} \\
= & (1 - N_{conf} W_{min}) \ \sum_Y \ |(V_t(Y) \ - \ B(Y))|
\end{aligned} \tag{61}$$

Hence for any start distribution we get the Boltzmann distribution in the limit of large t.

In particular

$$||V_t - B|| \leq \exp\left(-\frac{t}{R}\right) \quad ||V_0 - B|| \tag{62}$$

where $R = -1/\ln(1 - N_{conf} P_{min})$.

A sufficient condition for iii) is detailed balance :

$$\frac{W(X,Y)}{W(Y,X)} = \frac{B(X)}{B(Y)} \tag{63}$$

for all X and Y. **Prove**: (with i))

$$\sum_Y W(X,Y)B(Y) = \sum_Y W(Y,X)B(X) = B(X) \tag{64}$$

In practice “detailed balance” plays an important role; all algorithms that we shall discuss fulfil “detailed balance”.

3.4 Metropolis algorithm

A simple and and general usable algorithm is the so called Metropolis algorithm (N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller J.Chem.Phys. 21 (1953) 1087).

The transition probability $W(Y, X)$ is split into two parts. The first part is the probability $T(Y, X)$ to get a proposal Y starting from X. The proposal Y is accepted with the probability $A(Y, X)$, else the old configuration is kept. Hence $W(Y, X)$ becomes

$$W(Y, X) = A(Y, X)\, T(Y, X) \; + \; (1 - \sum_U A(U, X)\, T(U, X))\, \delta(X, Y) \tag{65}$$

T and A satisfy:

$$\sum_Y T(Y, X) = 1 \tag{66}$$

$$T(X, Y) = T(Y, X) \tag{67}$$

and

$$A(Y, X) = \min\left(1, \exp(-\beta(H(Y) - H(X)))\right) \quad . \tag{68}$$

The normalisation condition

$$\sum_Y W(Y, X) = 1 \tag{69}$$

is satisfied.

Also the proof of **detailed balance** is simple:
Assume $X \neq Y$
(The case $X = Y$ is trivial).

case 1) $H(X) > H(Y)$

$$\begin{aligned} \frac{W(Y,X)}{W(X,Y)} &= \frac{T(Y,X)A(Y,X)}{T(X,Y)A(X,Y)} \\ &= \frac{1}{\exp(-\beta(H(X) - H(Y)))} \\ &= \frac{\exp(-\beta H(Y))}{\exp(-\beta H(X))} \end{aligned} \tag{70}$$

case 2) $H(X) \leq H(Y)$

$$\begin{aligned} \frac{W(Y,X)}{W(X,Y)} &= \frac{T(Y,X)A(Y,X)}{T(X,Y)A(X,Y)} \\ &= \frac{\exp(-\beta(H(Y) - H(X)))}{1} \end{aligned}$$

$$= \frac{\exp(-\beta H(Y))}{\exp(-\beta H(X))} \tag{71}$$

In order to proof ergodicity, $T(Y, X)$ must be specified.

Note that eq. (68) is not the only possible choice for the acceptance probability. A simple modification is to multiply $A(Y, X)$ by a constant in $[0, 1)$. (Which however leads to a less efficient algorithm.)

3.5 Metropolis for the Ising Model

Now let us apply the Metropolis algorithm to the Ising model. In an elementary step, the proposal is to flip (i.e. multiply the spin with -1) the spin on a single site x on the lattice.

$$T_x(Y, X) = \delta([s_1, ..., -s_x, ..., s_V], [s_1, ..., s_x, ..., s_V]) \ . \tag{72}$$

In order to determine the acceptance probability, we have to compute the difference $H(Y) - H(X)$. Since the spin changes only at a single site, most terms cancel. We get

$$H(Y) - H(X) = 2\ s_x \sum_{y.nn.x} s_y \ \ , \tag{73}$$

where $y.nn.x$ denotes the sum over the nearest neighbours of x.

Remark on the implementation:
Computing the exponential function is rather expensive. Therefore we can speed up the program by computing all possible values of $A(Y, X)$ at the beginning of the simulation and putting them into an array. E.g. in 2 dimensions $\sum_{y.nn.x} s_y$ can take values in $-4, -2, 0, 2, 4$, for a vanishing external field.

Ergodicity:
Since in an elementary step only a single spin can be changed, such an elementary step is not sufficient to get ergodicity. To this end, a sequence of such elementary steps is needed.

This can be done in various ways:

- chose in each step the site x randomly

- Often a regular sequence of lattice points is used; e.g. $(1, 1)$,$(1, 2)$,$(1, 3)$ ... (L_1, L_2)

In the literature, this is called **"sweep"**.

Below I provide a function of a C-program that simulates the 2d Ising Model without external field using the Metropolis algorithm.

In the main program (or using other functions) a start configuration is provided and the observables are measured.

```
void metro()
  {
  int i0,i1,i0p,i0m,i1p,i1m,neigh;

/*sweep over the lattice*/
  for(i0=0;i0 < L0;i0++)
    {
/*take periodic boundary conditions into account*/
    i0p=i0+1; if(i0p==L0) i0p=0;  i0m=i0-1; if(i0==0) i0m=L0-1;
    for(i1=0;i1 < L1;i1++)
      {
/*take periodic boundary conditions into account*/
      i1p=i1+1; if(i1p==L1) i1p=0; i1m=i1-1; if(i1==0)   i1m=L1-1;

/*take the sum over the nearest neighbours*/
      neigh=spin[i0][i1p]+spin[i0][i1m]+spin[i0p][i1]+spin[i0m][i1];
      neigh=neigh*spin[i0][i1];

/* decide whether to accept the proposal spin to -spin */
      if(p[4+neigh]>drand48()) spin[i0][i1]=-spin[i0][i1];
      }
    }
  }
```

In main program we compute the table for the acceptance probability:

```
/* compute the acceptance probabilities */
p[0]=1.;
p[2]=1.;
p[4]=1.;
p[6]=exp(-4.*beta);
p[8]=exp(-8.*beta);
```

In the program above, the spins are stored in a 2 dimensional array. Alternatively, we can store them in a 1-dimensional array, and use in addition an array that contains the neighbourhood relations among the sites.
The entries of this array are computed once, before the simulation is started:

```
/* define the neighbours of a lattice point */
for(i0=0;i0 < L0;i0++)
  {
  /*take periodic boundary conditions into account*/
  i0p=i0+1; if(i0p==L0) i0p=0;  i0m=i0-1; if(i0==0) i0m=L0-1;
```

```
  for(i1=0;i1 < L1;i1++)
    {
/*take periodic boundary conditions into account*/
    i1p=i1+1; if(i1p==L1) i1p=0; i1m=i1-1; if(i1==0)   i1m=L1-1;

    nlist[i0*L1+i1][0]=i0p*L1+i1;
    nlist[i0*L1+i1][1]=i0*L1+i1p;
    nlist[i0*L1+i1][2]=i0m*L1+i1;
    nlist[i0*L1+i1][3]=i0*L1+i1m;
    }
  }
```

the function with the Metropolis update:

```
void metro()
{
int i,d,neigh;

/*sweep over the lattice*/
for(i=0;i < V;i++)
  {
  neigh=0;
  /*take the sum over the nearest neighbours*/
  for(d=0; d<2*D; d++) neigh+=spin[nlist[i][d]];
  neigh*=spin[i];
  /* decide whether to accept the proposal spin to -spin */
  if(p[4+neigh]>drand48()) spin[i]=-spin[i];
  }
}
```

Now we proceed with the simulation of the 2 D Ising model. For $J = 1$ (which we shall use throughout in the following), the inverse transition temperature is $\beta_c = 0.4406868....$ The transition is of second order. For temperatures that are larger than $T_c = 1/\beta_c$ there is no spontaneous magnetisation. For temperatures lower than T_c there is a spontaneous magnetisation. The exact result in the thermodynamic limit (i.e. for infinitely large volume) is

$$< m >= \left(1 - [\sinh(2\beta)]^{-4}\right)^{1/8} \quad . \tag{74}$$

(C. N. Yang, Phys. Rev. 85 (1952) 808.)

Let us first discuss the problem of thermalization. For large Monte Carlo times t we have

$$||B - V_t|| \propto \exp\left(-\frac{t}{R}\right) \quad . \tag{75}$$

To get correct estimates of the expectation values, the first nR (with n in practice 10 to 100) configurations have to be skipped.

In general R is not known a priori; hence we have to get a (rough) estimate from the simulation itself.

To this end it is useful to plot the evolution of the observables that we compute. In particular, if there is no experience with the algorithm applied to the particular model, it is very useful to perform simulations with different start configurations. (E.g. with an ordered and a disordered configuration.)

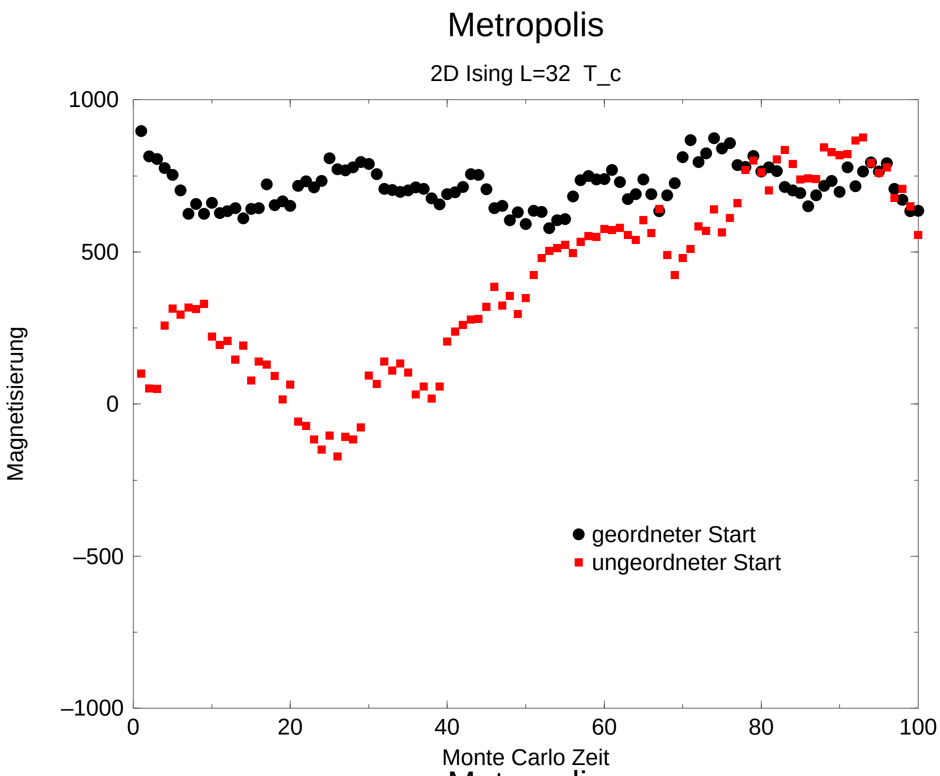
In fig. 2 the first 50 or 100 values of $-H$ and the magnetisation are given for a simulation of the 2D Ising model at $\beta = \beta_c$, $h = 0$ on a 32^2 lattice using the Metropolis algorithm. Between single "measurements" is one Metropolis sweep.

We see that the values of the magnetisation start to overlap for $t = 80$. To be on the save side, we skip the first 800 configurations. There is the "rules of thumb" of Alan Sokal who recommends to skip $100\tau_{exp}$ or even better $1000\tau_{exp}$. For the definition of the autocorrelation time τ_{exp} see below.

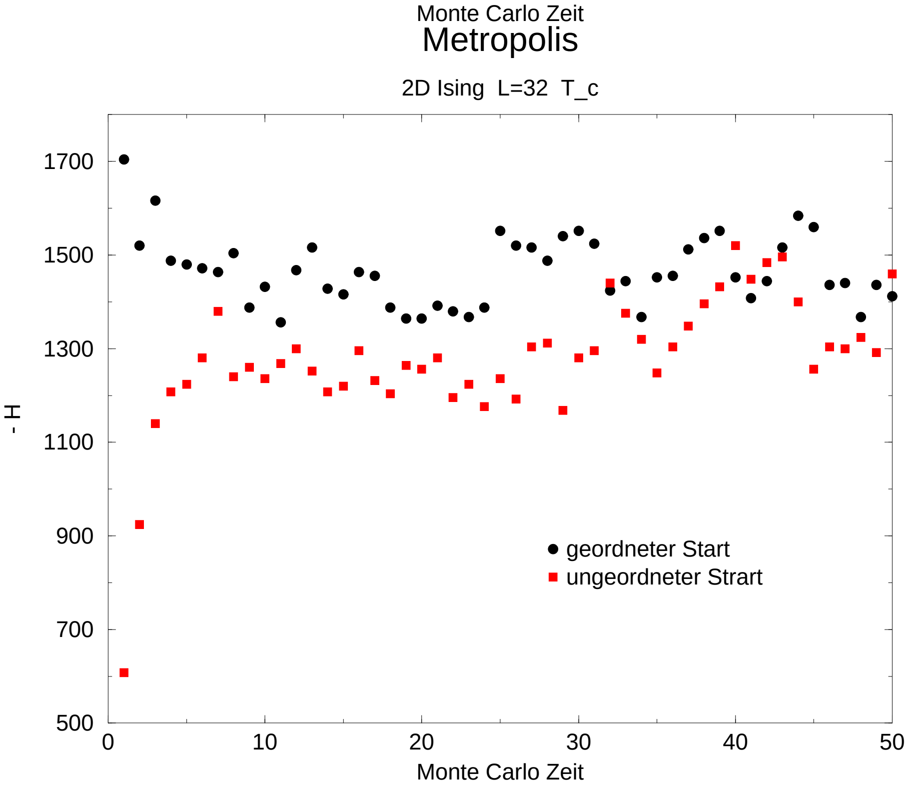
In Fig. 3 10000 measurements of the magnetisation for a 32^2 lattice are shown. The individual measurements are separated by one Metropolis-sweep. $\beta = 0.48, 0.4406868$ and 0.40. One run with 10000 sweeps on a 32^2 lattice took 44.4 s on an ancient computer at the HU Berlin.

We clearly see the difference between $\beta > \beta_c$ and $\beta < \beta_c$. In the first case, the system selects spontaneously a sign of the magnetisation and all measurements give a magnetisation that is larger than zero (This requires a reconsideration). For $\beta = 0.40$ the magnetisation fluctuates around zero.

We see in all three examples that the value of the magnetisation from subsequent measurements is similar. This is due to the fact that the configurations are generated by a Markov chain. There is a (unwanted) statistical correlation among the configurations that have been generated. In order to get a proper estimate of the statistical error of the estimates of the expectation values, we have to take into account the correlations.



Metropolis
2D Ising L=32 T_c
Magnetisierung
1000
500
0
–500
–1000
0
20
40
60
80
100
Monte Carlo Zeit
geordneter Start
ungeordneter Start



Metropolis
2D Ising L=32 T_c
- H
1700
1500
1300
1100
900
700
500
0
10
20
30
40
50
Monte Carlo Zeit
geordneter Start
ungeordneter Strart

Figure 2:

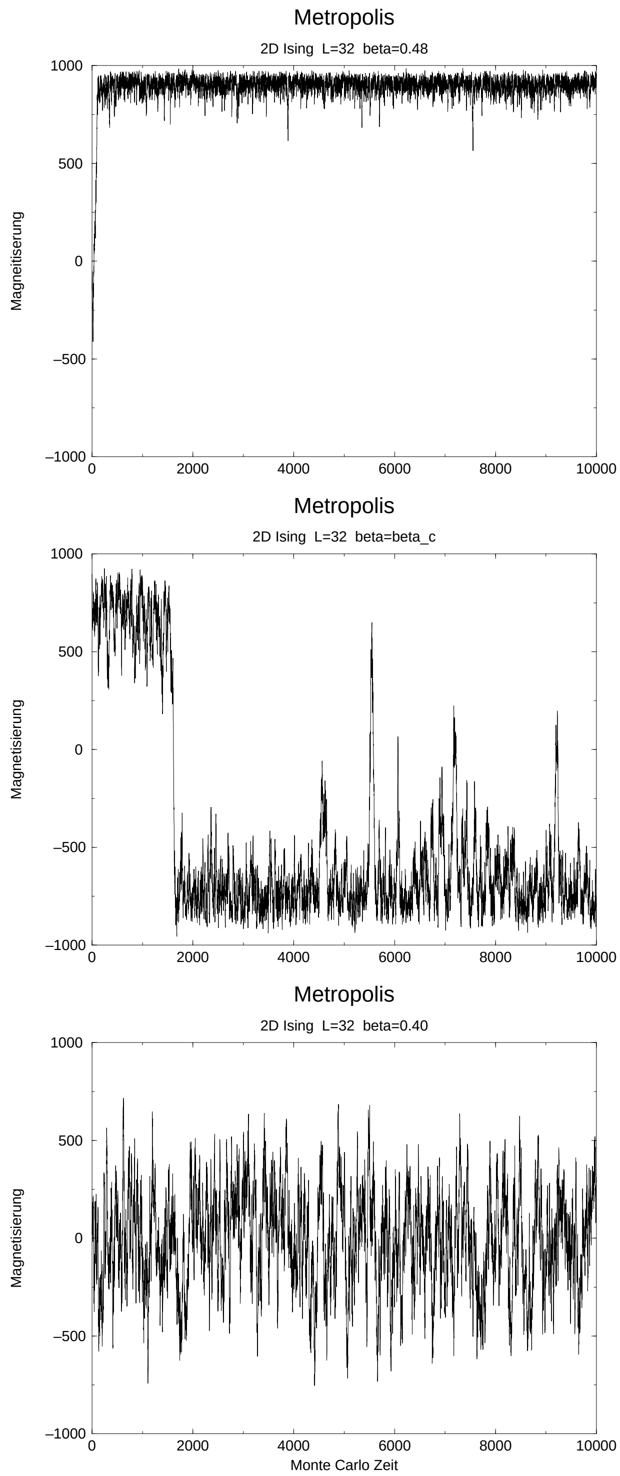

Metropolis
2D Ising L=32 beta=0.48
Magneitisierung
1000
500
0
−500
−1000
0
2000
4000
6000
8000
10000
Metropolis
2D Ising L=32 beta=beta_c
Magnetisierung
1000
500
0
−500
−1000
0
2000
4000
6000
8000
10000
Metropolis
2D Ising L=32 beta=0.40
Magnetisierung
1000
500
0
−500
−1000
0
2000
4000
6000
8000
10000
Monte Carlo Zeit

Figure 3:

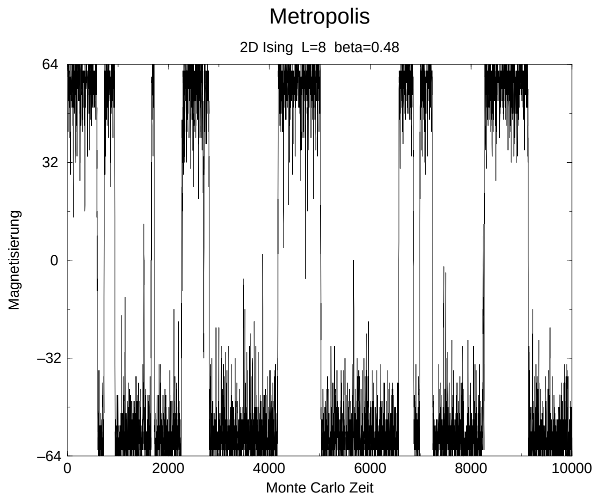


Figure 4:

4 Statistical error and Autocorrelation time

For data that are generated by a Markov chain, we have to generalize the analysis of the statistical error as we discussed before. The single "measurements",i.e. the values of the observables for the generated configurations are statistically correlated.

The simplest method to this end is the so called binning. The data divided into "bins" of the size n_b:

$$\hat{A}_j = \frac{1}{n_b} \sum_{i=(j-1)n_b+1}^{jn_b} A_i \; . \tag{76}$$

In the limit $n_b \to \infty$ the $\hat{A}_j$ become statistically independent. This allows us to use the error analysis for uncorrelated data:

$$\sigma^2(\bar{A}) = \frac{n_b}{N} \sigma^2(\hat{A}). \tag{77}$$

where N/n_b is the number of bins. In practice, one computes the statistical error for (i.e. multiply the spin with -1) various values of n_b, in order to check that with increasing n_b the error estimate stabilizes. Note however that with increasing n_b, at a fixed number of measurements, the number of bins decrease and hence the statistical error of the estimate of the error increases.

A more accurate estimate of the error can be obtained by the analysis of the "autocorrelation time.

Let us start with the variance of $\hat{A}$. The number of bins is $M = N/n_b$.

$$\sigma^2(\hat{A}) \quad = \quad \lim_{M\to\infty} \frac{1}{M} \sum_{j=1}^{M} \left(\bar{A} - \hat{A}_j\right)^2$$

$$
\begin{aligned}
&= \lim_{M\to\infty} \frac{1}{M} \sum_{j=1}^{M} \left(\hat{A}_j^2 - \bar{A}^2 \right) \\
&\quad \text{Now we use the definition of } \hat{A} \text{ and perform the square:} \\
&= \lim_{M\to\infty} \frac{1}{M} \sum_{j=1}^{M} \left(\frac{1}{n_b^2} \sum_{l=(j-1)n_b+1}^{jn_b} \; \sum_{k=(j-1)n_b+1}^{jn_b} A_l A_k - \bar{A}^2 \right) \\
&= \lim_{M\to\infty} \frac{1}{M} \sum_{j=1}^{M} \left(\frac{1}{n_b^2} \sum_{l=(j-1)n_b+1}^{jn_b} \; \sum_{t=(j-1)n_b+1-l}^{jn_b-l} A_l A_{l+t} - \bar{A}^2 \right) \\
&\approx \frac{1}{n_b} \sum_{t=-\infty}^{\infty} \left(< A_i A_{i+t} > - < A >^2 \right) \qquad (78)
\end{aligned}
$$

We make use of the fact that the distribution of A_i does not depend on i and that $< A_i A_{i+t} > - < A >^2$ decays exponentially with t.

The normalized autocorrelation function is defined by

$$
\rho_A(t) := \frac{< A_i A_{i+t} > - < A >^2}{< A^2 > - < A >^2} \; . \qquad (79)
$$

The **integrated autocorrelation time** is

$$
\tau_A := \frac{1}{2} \sum_{t=-\infty}^{\infty} \rho_A(t) = \frac{1}{2} + \sum_{t=1}^{\infty} \rho_A(t) \qquad (80)
$$

Using these definitions, the statistical error can be written as

$$
\sigma^2(\bar{A}) = \frac{2\tau_A}{N} \sigma^2(A) \; . \qquad (81)
$$

This can be interpreted such that the number of effectively independent measurements is reduced from N to $N/(2\tau_A)$.

In practice $\sigma^2(A)$ and τ_A are estimated from the simulation itself. Here we have to reduce the upper bound ∞ of the summation to some finite value. $t_{max} \approx 6\tau$ turns out to be a reasonable choice. Such a truncation is needed since the estimate of $\rho_A(t)$ is affected by a statistical error. For some t this error becomes larger than the actual value of $\rho_A(t)$. Hence summing beyond t_{max} as specified above, one just enlarges the statistical error of τ_A.

In the literature, we also find the so called exponential autocorrelation time defined by

$$
\rho_A(t) \sim \exp\left(-\frac{t}{\tau_{exp}} \right) \qquad (82)
$$

for large t.

5 Physics of the Phase Transition of the Ising Model

$$< m >\sim t^{\beta} \tag{83}$$

A collision of notation: β is used for the critical exponent as well as for the inverse temperature $1/T$.

The magnetic susceptibility is the derivative of the magnetisation with respect to the external field.

Writing the Boltzmann factor as $\exp(\beta \sum_{<x,y>} s_x s_y + h \sum_x s_x)$, we get

$$\chi = \frac{\partial < m >}{\partial h} = V \left(< m^2 > - < m >^2 \right) \quad . \tag{84}$$

Which follows from

$$\begin{aligned}
\frac{\partial < m >}{\partial h} &= \frac{(Z^{-1} \sum_s \exp(\beta \sum_{<x,y>} s_x s_y + h \sum_x s_x) \sum_x s_x / V}{\partial h} \\
&= Z^{-1} \sum_s \exp(\beta \sum_{<x,y>} s_x s_y + h \sum_x s_x) (\sum_x s_x)^2 / V \\
&+ \left(Z^{-1} \sum_s \exp(\beta \sum_{<x,y>} s_x s_y + h \sum_x s_x) \sum_x s_x / V \right)^2 \\
&= V \left(< m^2 > - < m >^2 \right) \quad .
\end{aligned} \tag{85}$$

For $h = 0$ the problem arises that there is no spontaneous magnetisation on a finite lattice. Therefore (at least for $\beta > \beta_c$) we study the modulus of the magnetisation.

The specific heat is given by the derivative of the internal energy density with respect to the temperature.

$$C = \frac{\partial < E >}{\partial T} = \frac{1}{T^2} V \left(< E^2 > - < E >^2 \right) \quad . \tag{86}$$

An important quantity is the so called correlation length ξ. It is the decay constant of the correlation function

$$G(x, y) =< s_x s_y > - < s_x >< s_y > \tag{87}$$

with

$$G(x, y) \sim \exp\left(-\frac{|x - y|}{\xi} \right) \quad . \tag{88}$$

In practice the so called "time-slice" correlation function is used to determine the correlation length.

One defines the average over a $D - 1$ line, plane, ... of the lattice

$$S_{x_0} = L^{D-1} \sum_{x_1,\ldots,x_D} s_x \quad . \tag{89}$$

(this way one projects to zero-momentum) The corresponding correlation function is

$$G(y) = < S_x S_{x+y} > - < S_x >< S_{x+y} > \tag{90}$$

Using the transfermatrix formalism one can show that

$$G(y) = \sum_i c_i \exp\left(-\frac{y}{\xi_i}\right) \ . \tag{91}$$

The reduced temperature is $t = |\beta - \beta_c|$ (In the literature there is no unique choice also $t = |\beta - \beta_c|/\beta_c$ and $t = |T - T_c|/T_c$ are used). At the transition the quantities introduced above behave as

$$\chi \sim t^{-\gamma} \ , \tag{92}$$

$$C \sim t^{-\alpha} \tag{93}$$

and

$$\xi \sim t^{-\nu} \tag{94}$$

From the exact solution of the 2 D Ising model one obtains:
$\beta = 0.125$, $\gamma = 1.75$, $\nu = 1$, $\alpha = 0$.

In 3 dimensions:
$\beta = 0.32648(18)$, $\gamma = 1.2371(4)$, $\nu = 0.63002(23)$, $\alpha = 0.1099(7)$.
(Massimo Campostrini, Andrea Pelissetto, Paolo Rossi, Ettore Vicari, cond-mat/9905078, Phys.Rev. E60 (1999) 3526-3563, High temperature series using information from Monte Carlo simulations)

In 4 Dimensions and higher:
$\beta = 0.5$, $\gamma = 1$, $\nu = 0.5$, $\alpha = 0$. (meanfield exponents)

5.1 Spontaneous Symmetry Breaking

For a finite system the magnetisation vanishes for all temperatures for a vanishing external field $h = 0$. This can be easily understood in the following way: Multiplying all spins of a configuration with -1 we get a configurations, with exactly the same energy, and hence the same Boltzmann factor, however with the other sign of the magnetisation. In the sum $\exp(-H)\sum_x s_x$ these pairs exactly cancel.

A value different from zero can only be obtained in the limit

$$< m > |_{h=0} := \lim_{h\to 0} \lim_{L\to\infty} \frac{1}{V} \frac{1}{Z} \sum_{conf} \exp(-H) \sum_x s_x \ . \tag{95}$$

In order to understand the problem, let us consider the histogram of the magnetisation:

$$P(m) := \frac{1}{Z} \sum_{conf} \exp(-\beta H) \ \delta(m, \frac{1}{V}\sum_x s_x) \ . \tag{96}$$

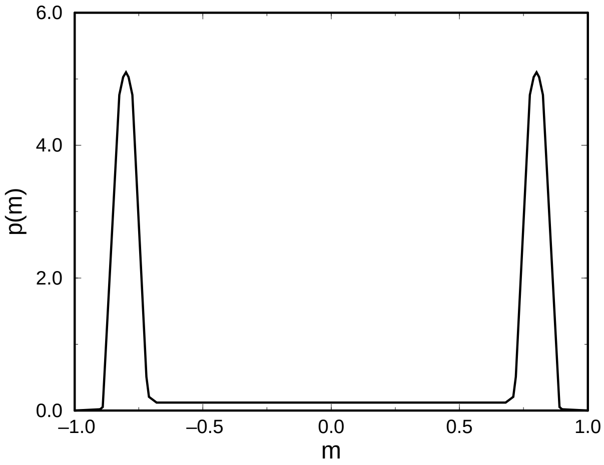


Figure 5: Schematic sketch of the magnetisation at low temperature $\beta > \beta_c$. To simplify the sketch, we assume continuous values of the magnetisation and given the probability distribution.

For $\beta < \beta_c$ and $h = 0$, $P(m)$ has its maximum at $m = 0$.

On the other hand, for $\beta > \beta_c$, the histogram has a double peak structure. The contributions around zero go down exponentially with the volume of the lattice. This is shown schematically in fig. 5.

Based on this histogram we can understand the time-evolution of the magnetisation at $\beta = 0.48$ given in figs. 3 and 4. In an elementary step of the Metropolis algorithm the sum $\sum_x s_x$ can only change by ± 2. On the way from negative to positive magnetisation (or vice versa) $m = 0$ has to be passed. The probability that the value $m = 0$ is assumed, is very small. Hence the sign of the magnetisation changes only after long Monte Carlo times. In particular, the time needed grows exponentially with the volume of the lattice. In fig. 4 for $L = 8$ the sign of the magnetisation changes frequently, while for $L = 32$, given in plot 3 there is no change of the sign (within our finite simulation).

In the Monte Carlo simulation it would be quite cumbersome, to perform the limits of eq. (95). It is better to consider

$$< m > |_{h=0,L=\infty} := \lim_{L\to\infty} \frac{1}{V} \frac{1}{Z} \sum_{conf} \exp(-H) \, |\sum_x s_x| \tag{97}$$

For the histogram discussed above, both definitions (95) and (97) lead to the same result.

In table 4 the expectation value of modulus of the magnetisation for square latices with periodic boundary conditions $L = 3, 4$ and 5 for various values of β.

We see that the deviations of the finite lattice result from the infinite volume limit becomes larger as the phase transition is approached.

Table 4: Expectation value of the modulus of the magnetisation for the 3^2,4^2 and 5^2 lattices at $h = 0$. For comparison we give the exact result for the thermodynamic limit in the last column.

L	0.48	0.50	0.55	0.60	0.70
3	0.910397	0.925985	0.954502	0.972103	0.989301
4	0.898543	0.918943	0.953589	0.972867	0.989969
5	0.891143	0.915209	0.953511	0.973291	0.990123
∞	0.877523	0.911319	0.953945	0.973609	0.990163

5.2 Finite Systems; Reweighting and Jackknife

Power laws are only valid for the **thermodynamic limit** (I.e. for the infinite volume).

Since we can simulate only finite systems, and the the lattice size can be varied only in a limited range, we need a good understanding of the dependence of the observables on the lattice size.

For a finite system, the partition function of a finite system is an analytic function of the temperature and the external field. The singularities at the transition, as given by the power laws, arises only in the thermodynamic limit.

E.g. the magnetic susceptibility can not become infinite on the finite lattice. However, it assumes larger and larger values in the neighbourhood of the transition, as the lattice size is increased.

Now it is useful to introduce a new sampling method, which is a mixture of "simple sampling" and "importance sampling":

$$< A > (\beta) \approx \frac{\sum_i \exp(-(\beta-\beta_s)H_i)A_i}{\sum_i \exp(-(\beta-\beta_s)H_i)} \quad , \tag{98}$$

where β_s is the inverse temperature that enters the Boltzmann factor that is used for the simulation. This method ("reweighting") allows to compute observables for β in the neighbourhood of β_s. With increasing $|\beta_s - \beta|$ however, the statistical error increases. This method is very simple, however it only became popular after the paper of Swendsen und Ferrenberg, Phys.Rev.Lett. 61 (1988) 2635.

An example is given in fig. 6. The 2 d Ising model at $\beta = 0.4406868$ for $L^2 = 4^2$, 8^2, 16^2 und 32^2 is simulated. We performed 100000 measurements of $-H$ and M.

Then, using reweighting for 100 β-values in the neighbourhood of β_c the magnetic susceptibility was computed. To this end $< |m| >$ and $< m^2 >$ was computed for these β-values.

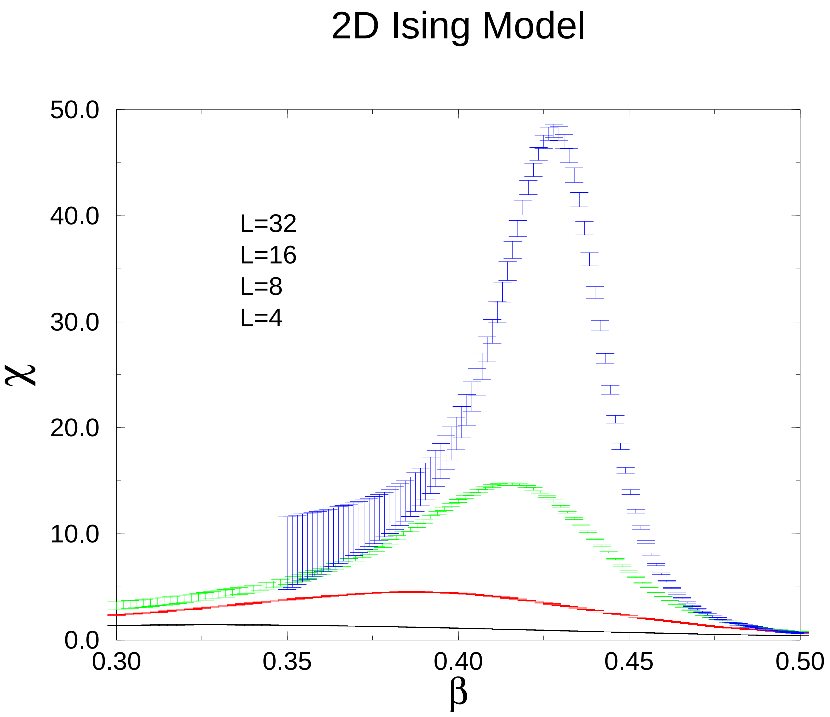


Figure 6:

The larger $|\beta - \beta_s|$, the large also the statistical error. The reason for this is similar to the problem of simple sampling.

In spite of this limitations, reweighting is quite useful. Using a single simulation, we can locate the maximum of the magnetic susceptibility.

For the estimate of the statistical error, we use a variant of the binning method, the so called Jackknife method.

For simple observables, this method is equivalent with binning. The advantage of the Jackknife method arises for combined quantities like the magnetic susceptibility or in connection with reweighting.

In the Jackknife method we consider the data sets that complementary to the bins of the binning method.

For the expectation value of an observable A this looks as follows:

$$\tilde{A}_j = \frac{1}{N - n_b} \left(\sum_{i=1}^{N} A_i - \sum_{i=(j-1)n_b+1}^{jn_b} A_i \right) \quad . \tag{99}$$

The statistical error (i.e. the root of the variance) of the average (i.e. the estimate of the expectation value) is given by

$$\sigma_{\bar{A}} = \sqrt{m[\frac{1}{m} \sum_{j=1}^{m} \tilde{A}_j^2 - (\frac{1}{m} \sum_{j=1}^{m} \tilde{A}_j)^2]} \quad . \tag{100}$$

Please note that in contrast to binning $\sqrt{m}$ appears in the enumerator. $m = N/n_b$ is the number of bins.

If we now want to determine the statistical error of a combined quantity that is a non-linear function $F(< A >, < B >, ...) \approx F(\bar{A}, \bar{B}, ...)$ of several observables, we proceed as follows:
First we compute

$$\tilde{F}_j = F(\tilde{A}_j, \tilde{B}_j, ...) \quad . \tag{101}$$

Here $\tilde{A}_j$ and so on are computed on the complements of bins as in eq. (99). Then the statistical error of F is given by

$$\sigma_{\bar{F}} = \sqrt{m[\frac{1}{m}\sum_{j=1}^{m}\tilde{F}_j^2 - (\frac{1}{m}\sum_{j=1}^{m}\tilde{F}_j)^2]} \quad . \tag{102}$$

Please note that the statistical error of F does not follow the simple error propagation

$$\sigma_{\bar{F}} = \sqrt{\left(\frac{\partial F}{\partial < A >}\sigma_{\bar{A}}\right)^2 + \left(\frac{\partial F}{\partial < B >}\sigma_{\bar{B}}\right)^2 + \ \ ... } \quad . \tag{103}$$

Since the measurements of A , B ... are correlated:

$$< AB > - < A >< B > \neq 0 \quad . \tag{104}$$

Similar to binning we have to perform the error analysis for a whole sequence of bin sizes n_b to check the stabilization of the error with increasing n_b.

6 Finite Size Scaling

For finite systems we can derive so called "Finite Size Scaling" laws using the Renormalisation group. E.g. for the magnetic susceptibility one obtains

$$\chi(L, \beta) = g(L/\xi)\ L^{\gamma/\nu} \quad . \tag{105}$$

The basic idea is that close to the transition the microscopic details of the model do not matter. The important length scales are L and ξ, while the lattice spacing can be ignored.

At the critical point we get power laws

$$\chi(L, \beta_c) \propto L^{\gamma/\nu} \quad . \tag{106}$$

The following rule holds: Compared with the power laws for the quantities in the thermodynamic limit, the reduced temperature t is replaced by L, and the exponent is multiplied with $-1/\nu$.

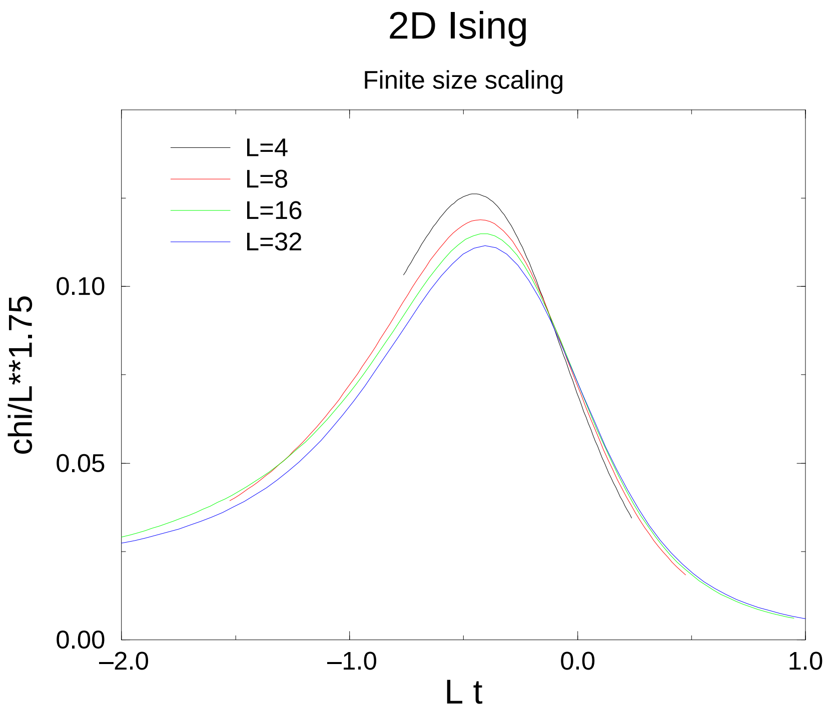


Figure 7:

For the following form of the Finite Size Scaling law knowledge of the correlation length ξ is not needed:

$$\chi(L,\beta) = g(L/\xi)\ L^{\gamma/\nu} = \tilde{g}(Lt^{\nu})L^{\gamma/\nu} \tag{107}$$

where we have used $\xi \propto t^{-\nu}$.

To verify this behaviour we give in fig. 7 the data of fig. 6 divided by $L^{\gamma/\nu}$ as a function of $L^{1/\nu}t$. If the finite size scaling law holds, the curves . Up to small corrections, this is indeed the case.

One way to determine the critical temperature is the so called cumulant method:

$$U(L,\beta) = 1 - \frac{1}{3}\frac{< m^4 >}{< m^2 >^2} \quad . \tag{108}$$

For large lattice size L and large correlation length ξ

$$U(L,\beta) = \tilde{U}(L/\xi) \tag{109}$$

holds. That means that the lattice constant is no importat length scale for the physics. The Binder cumulant takes at β_c a value that does not depend on L since at the transition $\xi = \infty$. This is the basis of the so called crossing method.

One determines the Binder cumulant for various values of L as a function of β. To this end, the reweighting method can be used. The intersection of these curves

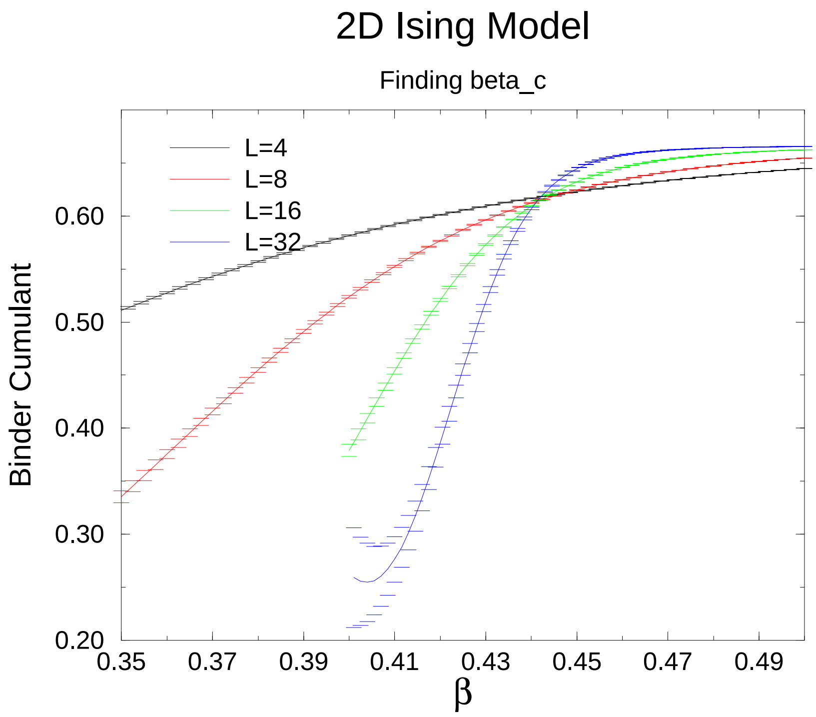


Figure 8:

gives an estimate of β_c. This is demonstrated at the example of the 2D Ising model in fig. 8.

7 Critical slowing down

Remember that the number of effectively independent measurements is given by $N_{eff} = N/(2\tau)$. The larger τ, the smaller the number of independent measurements. I.e. the statistical error becomes larger. Hence a good algorithm is characterized by a small value of τ. (Of course also the CPU time required for one update has to be taken into account.)

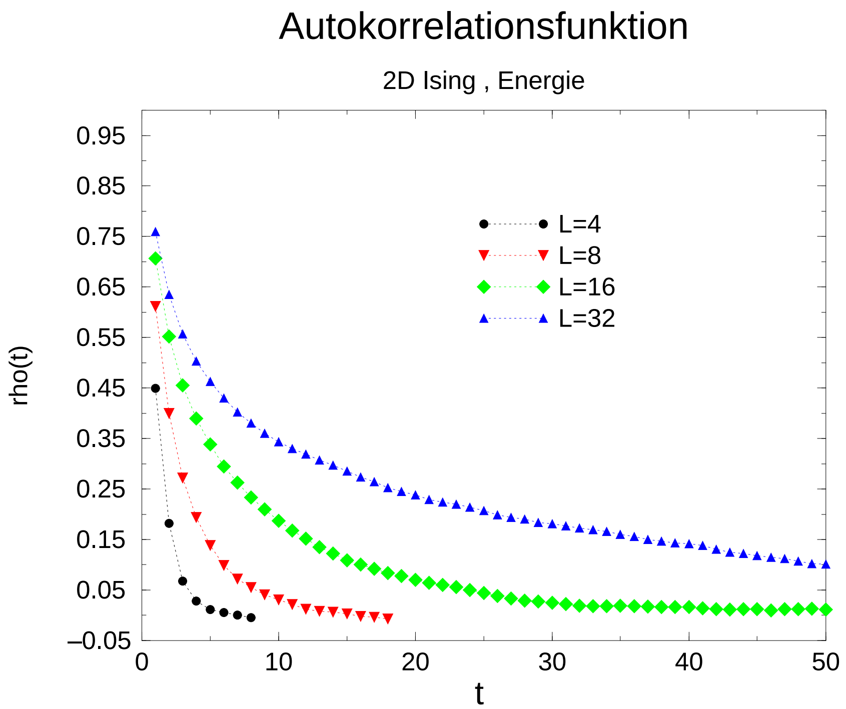
For local (local means, that in one elementary step of the algorithm only the spin at one site is changed) algorithms like the Metropolis algorithm the autocorrelation time grows as the transition temperature is approached. For $\xi << L$ we get

$$\tau \sim \xi^z \sim t^{-\nu z} \quad . \tag{110}$$

More general Finite-Size-Scaling-Form:

$$\tau \sim f(L/\xi) L^z \quad . \tag{111}$$

Assuming that a change in the configuration spreads like diffusion, one gets $z \approx 2$ for a local algorithm.


Autokorrelationsfunktion
2D Ising , Energie
rho(t)
t
0.95
0.85
0.75
0.65
0.55
0.45
0.35
0.25
0.15
0.05
–0.05
0
10
20
30
40
50
L=4
L=8
L=16
L=32

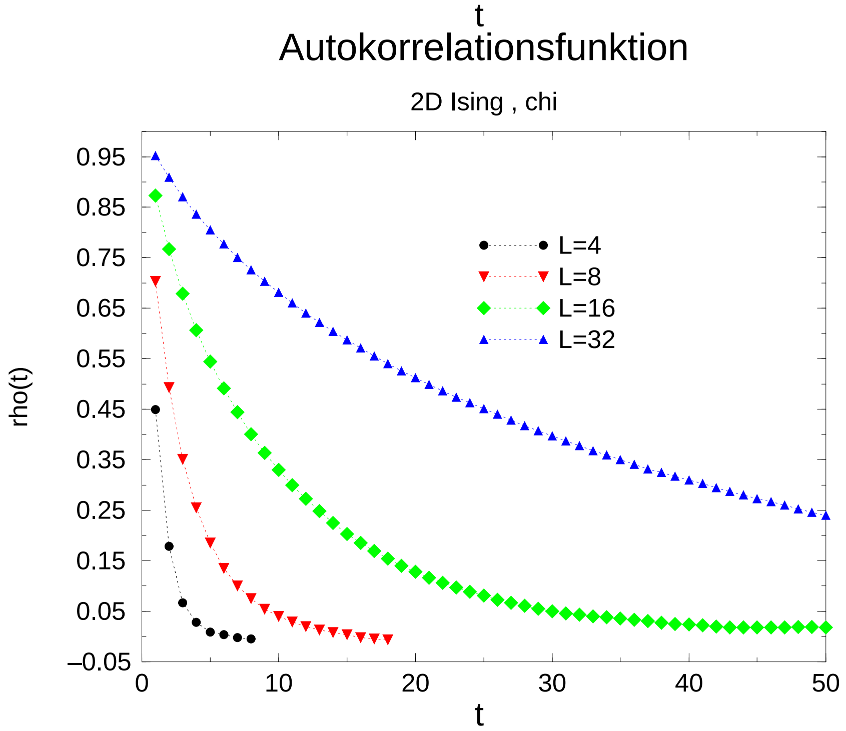

Autokorrelationsfunktion
2D Ising , chi
rho(t)
t
0.95
0.85
0.75
0.65
0.55
0.45
0.35
0.25
0.15
0.05
–0.05
0
10
20
30
40
50
L=4
L=8
L=16
L=32

Figure 9:

In figure 9 the autocorrelation function for the energy and the square of the magnetisation of the 2 D Ising model at the transition temperature is given.

The autocorrelation time increases with increasing lattice size L. The corresponding integrated τs, are summarized in table 3.

There is a factor of 2 in the lattice size from $L = 16$ to $L = 32$ while the τ s almost quadruple. In the literature one can find accurate estimates of the dynamical critical exponent, e.g. cond-mat/9601059 ($z = 2.1665(12)$).

Table 5: Integrated autocorrelation time for the energy and the square of the magnetisation for the Metropolis algorithm applied to the 2D Ising model. 100000 measurements each.

L	τ_E	τ_χ
4	1.24(2)	1.23(2)
8	2.45(7)	2.95(9)
16	6.1(3)	9.5(5)
32	16(2)	34(6)

One idea to fight critical slowing down is to change the spins on a large part of the lattice at once. E.g. in a elementary Metropolis step one could propose to change the sign of all spins on a l^d sublattice.

However, on average more neighbouring spins have the same sign than opposite sign. Hence mostly the proposal would lead to a large increase of the energy, roughly proportional to the surface of the sublattice. Hence the acceptance rate goes to zero as l increases.

8 Cluster algorithm

The solution of the problem discussed above is given by the cluster algorithm. The pioneering work is
R.H. Swendsen and J-Sh. Wang , Phys.Rev.Lett. 58 (1987) 86.

The cluster algorithm is based on the representation of the Ising model (and q-state Potts models) as weighted percolation model. This representation goes back to
C.M. Fortuin and P.W. Kasteleyn, Physica 57 (1972) 536

$$\begin{aligned} Z &= \sum_{s=\pm 1} \exp(\beta \sum_{<xy>} s_x s_y) \\ &= \sum_{s=\pm 1} \prod_{<xy>} \exp(\beta s_x s_y) \end{aligned}$$

$$
\begin{aligned}
&= \exp(-DV\beta) \sum_{s=\pm1} \prod_{<xy>} \Big(1 + \delta_{s_x,s_y}(\exp(+2\beta) - 1)\Big) \\
&= \exp(-DV\beta) \sum_{s=\pm1} \sum_{d=0,1} \prod_{<xy>} \Big((1 - d_{<xy>}) + d_{<xy>}\delta_{s_x,s_y}(\exp(+2\beta) - 1)\Big) \\
&. \qquad (112)
\end{aligned}
$$

The calculation of Fortuin and Kasteleyn continues by summing over the spins in order to arrive at a model that has only the d as degrees of freedom. (weighted percolation model).

For the discussion of the cluster algorithm it is best to to use the model that contains both degrees of freedom; the original spins s and the new variables d.

Roughly speaking, the cluster algorithm works in the following way: In the first step, the d get new values while the spins s are kept fixed, and in the second step the s get new values for fixed d.

Let us first discuss in detail how the d get their new values. We note that the relative weights for $d_{<xy>} = 0$ and $d_{<xy>} = 1$ don't depend of the values of the d on other edges. Hence the $d_{<xy>}$ can get new values completely independent of each other. There is only a dependence on the s_x and s_y at the two endpoints of the edge $< xy >$.

case 1: $s_x s_y = -1$

We get for the weight of $d_{<xy>}$:

$$
1 - d_{<xy>} + d_{<xy>}\delta_{s_x,s_y}(\exp(+2\beta) - 1)) = 1 - d_{<xy>} \tag{113}
$$

It follows that $d_{<xy>} = 0$ has the weight 1 and $d_{<xy>} = 1$ the weight 0.

case 2: $s_x s_y = 1$

We get for the weight of $d_{<xy>}$:

$$
1 - d_{<xy>} + d_{<xy>}\delta_{s_x,s_y}(\exp(+2\beta) - 1)) = 1 - d_{<xy>} + d_{<xy>}(\exp(+2\beta) - 1) \tag{114}
$$

Hence $d_{<xy>} = 0$ has the probability

$$
p(d_{<xy>} = 0) = \frac{1}{\exp(+2\beta) - 1 + 1} = \exp(-2\beta) \tag{115}
$$

and

$$
p(d_{<xy>} = 1) = 1 - \exp(-2\beta) \tag{116}
$$

As update of the d's we chose new values for the d's, following the probabilities given above:

$$
p(d_{<xy>} = 0) = \exp(-\beta(1 + s_x s_y)) \tag{117}
$$

and

$$
p(d_{<xy>} = 1) = 1 - p(d_{<xy>} = 0) \tag{118}
$$

In the **second step** we generate new spin variable s for fixed values of d.

To this end, let us consider the two cases $d_{<xy>} = 0$ and $d_{<xy>} = 1$.

Case 1: $d_{<xy>} = 1$
We get:

$$1 - d_{<xy>} + d_{<xy>}\delta_{s_x,s_y}(\exp(+2\beta) - 1)) = \delta_{s_x,s_y}(\exp(+2\beta) - 1)) \tag{119}$$

Since δ_{s_x,s_y}, $s_x s_y = 1$ must hold.

Case 2: $d_{<xy>} = 0$

$$1 - d_{<xy>} + d_{<xy>}\delta_{s_x,s_y}(\exp(+2\beta) - 1)) = 1 = \delta_{s_x,s_y} + \delta_{s_x,-s_y} \tag{120}$$

That means that $s_x s_y = 1$ and $s_x s_y = -1$ have the same weight.

Now we have to classify the configurations that fulfil the constraints that arise from $d_{<xy>} = 1$.

To this end we introduce the notion of a **cluster**. Two sites of the lattice belong to the same cluster if there is a line connection these points, such that all edges $< xy >$ that belong to the line have $d_{<xy>} = 1$.

It follows that all spins within a given cluster must have the same value. The remaining degrees of freedom are an overall sign for each cluster. Each sign has the same weight.

In the algorithm of Swendsen and Wang the new value of the sign is chosen among -1 and $+1$ with equal probability.

8.1 How to compute the clusters

The efficient determination of the clusters is a non-trivial problem. This problem also arises in the study of percolation. Beyond physics, the computation of clusters is used e.g. in image recognition.

In the literature mainly two methods are discussed:
-Hoshen-Kopelman, J.Hoshen and R.Kopelman, Phys.Rev.B 14 (1976) 3438
-"Ants in the labyrinth".

Let us discuss here the ants in the labyrinth method.

First we take a site x that is not assigned to a cluster yet. Next we consider all neighbours y of this lattice point. If $d_{<xy>} = 1$, the point y is added to the cluster. The new members of the cluster are noted in a list. Now we continue the search starting from the sites contained in the list. If there is no site left in the list, the construction of the cluster is completed.

Pseudo-C code for the construction of a cluster.

```
inew=1;
iold=0;
position[inew]=V*g05caf();
while(iold < inew)
  {
  iold++;
```

```
  for(all neighbours of position[iold])
    {
    if(d_{neighbour,position[iold]} = 1) && neighbour not in the cluster yet)
      {
      inew++;
      position[inew]=neighbour;
      Mark Neighbour as member of the cluster;
      }
  }
```

In addition to the Swendsen Wang algorithm there are variants of the cluster algorithm:

The most important one is the single cluster algorithm that has been proposed by U. Wolff:
COLLECTIVE MONTE CARLO UPDATING FOR SPIN SYSTEMS
Phys.Rev.Lett.62:361,1989.

Here a single point of the lattice is selected randomly, only the cluster that contains is computed. All spins within this cluster change sign (jargon: "are flipped").

Below you find a program for the simulation of the 3D Ising model with the single cluster algorithm. Note that the new $d_{<xy>}$ are completely independent of the old values of $d_{<xy>}$. Hence there is no need to store the $d_{<xy>}$. The can be computed at the time when they are needed for the construction of the cluster.

```
void cluster()
{
int i0,i1,i2,is0,is1,is2,iq0,iq1,iq2;
int iz1,iz2,ipp;

/* determine the stating point of the cluster */
i0=L0*g05caf();
i1=L1*g05caf();
i2=L2*g05caf();

iz1=-1 ; iz2=0 ;
spin[i0][i1][i2]=-spin[i0][i1][i2];

/*go ahead in the list cc[] until all entry are visited */
while(iz1<iz2)
  {
  ipp=spin[i0][i1][i2];

/* have a look at all six neighbours */
  is0=i0+1; if(is0==L0) is0=0;  /* periodic boundary conditions */
```

```
  FREEZE_DELETE(is0,i1,i2)
  iq0=i0-1; if(i0==0) iq0=L0-1; /* periodic boundary conditions */
  FREEZE_DELETE(iq0,i1,i2)
  is1=i1+1; if(is1==L1) is1=0;  /* periodic boundary conditions */
  FREEZE_DELETE(i0,is1,i2)
  iq1=i1-1; if(i1==0) iq1=L1-1; /* periodic boundary conditions */
  FREEZE_DELETE(i0,iq1,i2)
  is2=i2+1; if(is2==L2) is2=0;  /* periodic boundary conditions */
  FREEZE_DELETE(i0,i1,is2)
  iq2=i2-1; if(i2==0) iq2=L2-1; /* periodic boundary conditions */
  FREEZE_DELETE(i0,i1,iq2)

  i0=cc[++iz1];

/* to save space we put i0,i1,i2  into a single word
   To this end we use bit-wise manipulations:
*/
  i1=i0>>10;
  i2=i0>>20;
  i0=i0&1023;
  i1=i1&1023;
  }
}
```

Using the C-Preprocessor we insert:

```
#define FREEZE_DELETE(A,B,C)      \
if(spin[A][B][C]!=ipp)            \
  {                               \
  G05;                            \
  if(pd<xx)                       \
    {                             \
    spin[A][B][C]=ipp;            \
    cc[iz2++]=A|(B<<10)|(C<<20);  \
    }                             \
}
```

9 Cluster Algorithm beyond the Ising Model

The basic idea, is to pick out degrees of freedom in $-1, 1$. In a single step of the algorithm only these degrees of freedom are changed, while all other stay fixed.

In the following we shall see how this basic idea is implemented for the Landau-Ginzburg model on the lattice, $O(N)$-invariant nonlinear σ-models and interface models.

9.1 Landau-Ginzburg model on the lattice

Let us consider the Landau-Ginzburg model with a one component field. The Hamiltonian is given by

$$\beta H = \sum_x \left(-2\kappa \sum_\mu \phi_x \phi_{x+\hat{\mu}} + \phi_x^2 + \lambda(\phi_x^2 - 1)^2 \right) \quad . \tag{121}$$

In the limit $\lambda \to \infty$ we get the Ising model. For other values $\lambda > 0$ the model is in the same universality class as the Ising model. I.e. the critical exponents do not depend on λ, they always assume the same values as in the Ising model.

The application of the cluster algorithm to this model was first discussed in EMBEDDED DYNAMICS FOR ϕ^4 THEORY, R.C. Brower, P. Tamayo, PHYS. REV. LETT. 62 (1989) 1087-1090.

As degree of freedom for the cluster algorithm one chooses the sign of the field ϕ. We see immediately that by changing the sign only the term

$$-2\kappa \sum_x \sum_\mu \phi_x \phi_{x+\hat{\mu}} \tag{122}$$

might change its value. If we write $\phi_x = s_x|\phi_x|$ we get as effective $\beta_{<xy>} = 2\kappa|\phi_x||\phi_y|$ for the s. Now we can insert this $\beta_{<xy>}$ into equation (117) for $p(d)$:

$$p(d_{<xy>} = 0) = \exp(-\beta_{<xy>}(1+s_x s_y)) = \exp(-2\kappa|\phi_x||\phi_y|(1+s_x s_y)) = \min\left(1, \exp(-4\kappa\phi_x\phi_y)\right) \quad . \tag{123}$$

In order to get an ergodic algorithm we have to use local Metropolis sweeps in addition to the cluster updates.

A proposal for a new value of ϕ_x could look like $\phi'_x = \phi_x + s(drand48() - 0.5)$. s is a free parameter of the algorithm. As rule of thumb one might chose s such that the acceptance rate is about 50%.

9.2 O(N)-invariant non-linear σ models

COLLECTIVE MONTE CARLO UPDATING FOR SPIN SYSTEMS, Ulli Wolff, Phys.Rev.Lett.62:361,1989.

These models are characterised by the Hamiltonian

$$H = -J \sum_{<xy>} \vec{s}_x \vec{s}_y + \vec{h} \sum \vec{s}_x \tag{124}$$

where $\vec{s}_x$ is a unit vector in R^N. In the following we set $J = 1$.

For the cluster algorithm, the trick is again to chose some $-1, 1$ degrees of freedom. To this end we chose randomly a direction in R^N. In a single step of the cluster algorithm only the component parallel to this direction might change its sign. We get

$$p(d_{<xy>} = 0) = \min\left(1, \exp(-2\beta s_x^{(p)} s_y^{(p)})\right) \tag{125}$$

If one choses a new direction for each cluster-update, the algorithm becomes ergodic; no additional Metropolis updates are needed.

9.3 Interface models

STOCHASTIC CLUSTER ALGORITHMS FOR DISCRETE GAUSSIAN (SOS) MODELS. By H.G. Evertz (Florida State U., SCRI), M. Hasenbusch (Kaiserslautern U.), M. Marcu (Hamburg U. and Tel Aviv U.), K. Pinn (Hamburg U. and Münster U., ITP), S. Solomon (Hebrew U.), Phys.Lett.B254:185-191,1991.

In the case of interface models, the field variable assumes integer values h_x. The idea is that h_x is the number of atoms that is stapled on a 2D plane.

In the literature various Hamiltonians are discussed. e.g. (Discrete Gaussian SOS)

$$H = \sum_{<xy>} (h_x - h_y)^2 \tag{126}$$

or (Absolute Value SOS)

$$H = \sum_{<xy>} |h_x - h_y| \tag{127}$$

It is common to all H's, that roughness, i.e. the fluctuation of the h, increases the value of H. Hence at low temperatures, almost all h_x are the same; i.e. the interface is smooth.

As the temperature increases, the interface becomes rough. First this roughness is restricted to small length scales.

As a certain temperature is passed, the interface becomes rough at all length scales. This is the so called roughening transition.

For the cluster algorithm we need again degrees of freedom in $-1, 1$. Here we get them by reflecting h_x at a certain reference height m:

$$h_x = m + s_x |h_x - m| \tag{128}$$

h_x has to stay integer. Therefore m has to be either integer or half-integer.

How do we get m?
A lattice point x is chosen randomly. Then one choses $m = h_x \pm 0.5$ with equal probability for $+1/2$ and $-1/2$. For integer m one choses $m = h_x \pm 1$.

Please note, that the probability to chose a certain m, is the same before and after the cluster update. (which is needed to proof detailed balance).

An important generalization of the cluster algorithm is the THE LOOP ALGORITHM; It can be used for the efficient simulation of some quantum mechanical models. See: H.G. Evertz (Graz U.) Adv.Phys.52:1,2003, e-Print Archive: cond-mat/9707221

9.4 Where does the cluster algorithm fail?

Unfortunaltely not all hopes related with the cluster algorithm became reality. For a large range of important models no good implementation of the cluster algorithm was found. In particular there is no cluster algorithm for lattice QCD (For efforts in this direction see e.g. QUANTUM SPINS AND QUANTUM LINKS: THE D THEORY APPROACH TO FIELD THEORY. By U.J. Wiese (MIT, LNS and MIT),. Jul 1998. 14pp. Talk given at 16th International Symposium on Lattice Field Theory (LATTICE 98), Boulder, CO, 13-18 Jul 1998. Published in Nucl.Phys.Proc.Suppl.73:146-158,1999 e-Print Archive: hep-lat/9811025
and refs. therein)

In condensed matter physics, spin-glasses are hard to simulate ; also here no good cluster algorithm was found. It turned out that the cluster algorithm fails, when there is frustration.

Frustration means, that the product of the couplings $J_{<xy>}$ around some closed loop on the lattice is negative. This is the case for spin classes by construction.

10 Metropolis, Heatbath and over-relaxation for the classical Heisenberg model.

(The The classical Heisenberg model is defined by

$$H = -J \sum_{<xy>} \vec{s}_x \vec{s}_y - \vec{h} \sum_x \vec{s}_x \tag{129}$$

Where $\vec{s}_x$ is a unit vector in R^3. The model can be simulated very efficiently using the cluster algorithm. However here we like to use this simple model to introduce further local algorithms that can also be used in the simulation of lattice QCD.

Briefly on the physics of the model:
In 3 dimensions there is a second order phase transition. Recent Mote Carlo studies of the model are:

Finite size effects on measures of critical exponents in d=3 O(N) models
H. G. Ballesteros, L. A. Fernandez, V. Martin-Mayor, A. Munoz Sudupe (Universidad Complutense de Madrid), cond-mat/9606203, Phys.Lett. B387 (1996) 125-131

The inverse of the critical temperature is ($h = 0,\ J = 1$): $\beta_c = 0.693002(12)$.

Critical Exponents of the Classical Heisenberg Ferromagnet
Christian Holm, Wolfhard Janke
(FU-Berlin and JGU Mainz), hep-lat/9605024, Phys. Rev. Lett. 17 (1997) 2265

CRITICAL EXPONENTS AND EQUATION OF STATE OF THE THREE-DIMENSIONAL HEISENBERG UNIVERSALITY CLASS. By Massimo Campostrini (Pisa U. INFN, Pisa), Martin Hasenbusch (NIC, Zeuthen DESY), Andrea Pelissetto (Rome U. INFN, Rome), Paolo Rossi, Ettore Vicari (Pisa U. INFN, Pisa),. DESY-01-154, IFUP-TH-2001-33, 2002. 40pp. Published in Phys.Rev.B65:144520,2002 e-Print Archive: cond-mat/0110336

The exponents assume values different from the Ising ones:
$\gamma = 1.3960(9)$, $\nu = 0.7112(5)$, $\eta = 0.0375(5)$, $\alpha = -0.1336(15)$, $\beta = 0.3689(3)$, and $\delta = 4.783(3)$.

In two dimensions the model does not undergo a phase transition. For all finite temperatures there is a vanishing magnetisation (in the thermodynamic limit). The model is asymptotically free. The discussion of this property goes beyond the scope of the manuscript; We just remark that QCD shares this property. For this reason the 2D Heisenberg Model is often used as "toy model" in theoretical high energy physics.

10.1 Metropolis

First we discuss, how the Metropolis algorithm can be applied to the Heisenberg model. Indeed there are various possible choices.
Maybe the following is the simplest:

In an elementary step of the algorithm only two of the three components are changed. The proposal is generated by a rotation among these two components.

Say we take the 1 and 2 component:

$$\begin{aligned} s'_1 &= \cos(\alpha)\ s_1 + \sin(\alpha)\ s_2 \\ s'_2 &= -\sin(\alpha)\ s_1 + \cos(\alpha)\ s_2 \ . \end{aligned} \tag{130}$$

Where α is a random number that is uniformly distributed in $[-\alpha_{max}, \alpha_{max}]$. Here of course also other distributions can be used, as long as $p(-\alpha) = p(\alpha)$. The acceptance propability looks as in case of the Ising model

$$A(s', s) = \min\left(1, \exp(-\beta(H(s') - H(s)))\right) \ . \tag{131}$$

In order to get an ergodic algorithm, we also have to use $\{1, 3\}$ and $\{2, 3\}$ pairs of components for the update.

In contrast to the Ising model we have the "step-size" α_{max} as free parameter of the algorithm. The integrated autocorrelation time τ depends on α_{max}. If one chooses α_{max} very small, there is a large acceptance rate, however the change from

s to s' are small. On the other hand, the acceptance rate becomes small if α_{max} becomes large. The optimal case is somewhere between. As rule of thumb, one should choose α_{max} such that the acceptance rate is about 50%.

10.2 Heatbath

The idea of the heatbath algorithm is to chose the spin at a given lattice point x, for fixed spins elsewhere, with a probability distribution proportional to the Boltzmann factor. In contrast to the Metropolis algorithm, there is no correlation with the old value of the spin at this lattice site. The probability distribution is given by

$$P(\vec{s}_x) \propto \delta(\vec{s}_x^2 - 1) \exp(\beta \vec{S}_N \vec{s}_x) \tag{132}$$

where $S_N = \sum_{y.nn.x} \vec{s}_x$.

How can one generate $\vec{s}_x$ with the desired probability distribution?

We split the spin $\vec{s}_x$ in a component $s_{p,x}$ parallel to $\vec{S}_N$ and in two components perpendicular to $\vec{S}_N$: $s_{s1,x}$, $s_{s2,x}$.

The distribution for the $s_{p,x}$ component is given by the integration over the remaining components:

$$\begin{aligned}
\tilde{P}(s_p) &= \int_{-\infty}^{+\infty} ds_{s1} \int_{-\infty}^{+\infty} ds_{s2} \;\; P(\vec{s}_x) \\
&\propto \int_{-\infty}^{+\infty} ds_{s1} \int_{-\infty}^{+\infty} ds_{s2} \; \delta(\vec{s}_x^2 - 1) \exp(\beta \vec{S}_N \vec{s}_x) \\
&= \exp(\beta |S_N| s_p) \;\; \int_{-\infty}^{+\infty} ds_{s1} \int_{-\infty}^{+\infty} ds_{s2} \; \delta(\vec{s}_x^2 - 1) \;\; .
\end{aligned} \tag{133}$$

For $c > 0$ holds:

$$\int_{-\infty}^{+\infty} \mathrm{d}x \;\; \delta(x^2 - c) = 2 \;\int_0^{+\infty} \mathrm{d}y \;\; \frac{1}{2\sqrt{y}} \,\delta(y - c) = \frac{1}{\sqrt{c}} \quad . \tag{134}$$

Where we have substituted $y = x^2$. The 2 in front of the integral comes from the roots of x^2.

This result can now be used:

$$\begin{aligned}
&\int_{-\infty}^{+\infty} ds_{s1} \int_{-\infty}^{+\infty} ds_{s2} \;\; \delta\left(s_{s1}^2 \; + \; s_{s2}^2 \; - \; (1 - s_p^2)\right) \\
= &\int_{-\sqrt{1-s_p^2}}^{+\sqrt{1-s_p^2}} \; ds_{s1} \;\; \frac{1}{\sqrt{1 - s_p^2 - s_{s1}^2}} \quad .
\end{aligned} \tag{135}$$

The value of this integral does not depend on s_p, as we can see from the substitution

$$y = \frac{s_{s1}}{\sqrt{1 - s_p^2}} \tag{136}$$

In general, for $N > 1$:

$$
\begin{aligned}
& \int dx_1 ... \int dx_{N-1}\ \delta(x_1^2\ + ... +\ x_{N-1}^2\ +\ x_N^2 - 1) \\
\propto\ & \int \text{angle} \int dr\ r^{-N+2}\ \delta(r^2\ +\ x_N^2 - 1) \\
\propto\ & \int dr\ r^{N-2}\ \delta(r^2\ +\ x_N^2 - 1) \\
\propto\ & \int dy\ y^{(N-2)/2-1/2}\ \delta(y\ +\ x_N^2 - 1) \\
=\ & (1 - x_N^2)^{(N-3)/2} \quad .
\end{aligned} \tag{137}
$$

I.e. $N = 3$ is the simplest case. We generate s_p with

$$
s_p = \ln\Big(z\ +\ (z^{-1} - z)\ r\ \Big)/\tilde{\beta} \quad . \tag{138}
$$

Where $z = \exp(-\tilde{\beta})$ and $\tilde{\beta} = \beta|\vec{S}_N|$. r is a random number that has a uniform distribution in $(0, 1]$.

Let us convince ourself that this generates the correct distribution:

$$
\begin{aligned}
\frac{\mathrm{d}s_p}{\mathrm{d}r} &= \frac{1}{\tilde{\beta}}\ \frac{z^{-1} - z}{z\ +\ (z^{-1} - z)r} \\
&= \frac{z^{-1} - z}{\tilde{\beta}}\ \frac{1}{z\ +\ (z^{-1} - z)r} \\
&= \frac{z^{-1} - z}{\tilde{\beta}}\ \exp(-\tilde{\beta}s_p)
\end{aligned} \tag{139}
$$

The last step follows from (138), since

$$
\exp(\tilde{\beta}s_p) = z\ +\ (z^{-1} - z)\ r \quad . \tag{140}
$$

Recall that

$$
\tilde{P}(s_p) = \left|\frac{\mathrm{d}r}{\mathrm{d}s_p}\right| P(r) \quad . \tag{141}
$$

Hence

$$
\tilde{P}(s_p) = \frac{\tilde{\beta}}{z^{-1} - z}\ \ \exp(\tilde{\beta}s_p) \tag{142}
$$

as desired.

The generation of the remaining two components is simple: We take a random number α that is uniformly distributed in $(0, 2\pi]$. Then the perpendicular components are given by

$$
\begin{aligned}
s_{s1} &= \sqrt{1 - s_p^2}\cos(\alpha) \\
s_{s2} &= \sqrt{1 - s_p^2}\sin(\alpha)
\end{aligned} \tag{143}
$$

For illustration a fortran77 program:

```
xx=neigh(1)**2+neigh(2)**2+neigh(3)**2
        xx=beta*sqrt(xx)
yy=1D0/xx

u(1)=beta*neigh(1)*yy
u(2)=beta*neigh(2)*yy
u(3)=beta*neigh(3)*yy

z=exp(-xx)
t=log(z+(1D0/z-z)*xjames())*yy

if(abs(u(1)).gt.0.25D0) then
  a=sqrt((1D0-t**2)/(1D0-u(3)**2))
  v1(1)= u(2)*a
  v1(2)=-u(1)*a
  v1(3)=0D0
else
  a=sqrt((1D0-t**2)/(1D0-u(1)**2))
          v1(1)=0D0
          v1(2)= u(3)*a
  v1(3)=-u(2)*a
endif

v2(1)=u(2)*v1(3)-u(3)*v1(2)
v2(2)=u(3)*v1(1)-u(1)*v1(3)
v2(3)=u(1)*v1(2)-u(2)*v1(1)


ph=2D0*3.1415926535897932D0*xjames()
        co=cos(ph)
        si=sin(ph)

        sp(1,i1,i2,i3)=t*u(1)+co*v1(1)+si*v2(1)
        sp(2,i1,i2,i3)=t*u(2)+co*v1(2)+si*v2(2)
        sp(3,i1,i2,i3)=t*u(3)+co*v1(3)+si*v2(3)
```

10.3 Overrelaxation

The idea of overrelaxation stems from the iterative solution of systems of linear equations with sparse matrices. Such problems arise e.g. by discretising partial differential equations. Traditional methods are the Jacobi and the Gauss-Seidel method. (see textbooks on numerical mathematics, or numerical recipes, Press et al).

A simple improvement of these methods is overrelaxation. The idea is to overshoot the Gauss-Seidel step.

The simplest realisation in the case of O(N)-invariant non-linear σ-models is, to choose the new spin such that the value of the Hamiltonian stays exactly constant, while there is a maximal distance between the old and the new value of the spin. This can be achived by reflecting the spin at the sum of its neighbours:

$$\vec{s}_x\,' = 2\frac{(\vec{S}_N\vec{s}_x)}{\vec{S}_N^2}\vec{S}_N - \vec{s}_x \qquad (144)$$

To see that this algorithm fulfils detailed balance, we consider it as a special case of the Metropolis algorithm. The probability to get from s_x to s'_x is the same as vice versa. Since H does not change, the acceptance probability is 1.

Since H is not changed by overrelaxation updates, the algorithm can not be ergodic on its own. Hence we have to combine overrelaxation sweeps with Metropolis or Heatbath sweeps.

A complete cycle of the update has the following form:
1 heatbath sweep followed by M overrelaxation sweeps.
The parameter M should be chosen such that $\tau\times$ (CPU-Zeit pro Zyklus) becomes minimal.

Analytic calculation for the Gaussian model and numerical experiments for other models have shown that

$$M_{opt} \propto \xi \qquad (145)$$

for $L >> \xi$ or

$$M_{opt} \propto L \qquad (146)$$

for $T \approx T_c$.

Using this optimal choice, $z \approx 1$ can be achived.

10.4 Demons

The use of "demons" in Monte Carlo simulations was first proposed by M.Creutz.
M. Creutz, Phys. Rev. Lett. 50 (1983) 1411.

The name suggests something mysterious. Acctually it is a quite simple trick. One adds additional degrees of freedom to the system that do not couple with the physical degrees of freedom. In the Hamiltonion that looks like:

$$H_{comb}(s,d) = H_{spin}(s) + H_{demon}(d) \ , \qquad (147)$$

where $H_{demon}(d)$ is given by

$$H_{demon}(d) = \sum_{i=1}^{N_d} d_i \ , \qquad (148)$$

where d_i is a positive real number. Any number of demons can be chosen. First we show that for observables that depend only on the spins, the expectation value is the same for $H_{comb}(s,d)$ and $H_{spin}(s)$.

$$
\begin{aligned}
\frac{\int \mathrm{D}s \int \mathrm{D}d \;\; \exp\left(-\beta H_{spin}(s) - \beta H_{demon}(d)\right) \; A(s)}{Z_{comb}} &= \\
\frac{\int \mathrm{D}s \;\; \exp(-\beta H_{spin}(s)) \; A(s) \quad \int \mathrm{D}d \;\; \exp(-\beta H_{demon}(d))}{Z_{spin} \; Z_{demon}} &= \\
\frac{\int \mathrm{D}s \;\; \exp(-\beta H_{spin}(s)) \; A(s) \quad Z_{demon}}{Z_{spin} \; Z_{demon}} &= \\
< A > \quad . & \qquad (149)
\end{aligned}
$$

The demons allow for new types of updates. Let us take the case of a single demon. An elementary (Metropolis-like) step using the demon looks as follows: As before we compute a proposal s'_x for the spin at the site x. We compute the corresponding change in the Hamiltonian $\Delta H_{spin} = H'_{spin} - H_{spin}$ of the spin model. In the standard Metropolis we compute the acceptance rate from it. In the demon case instead we check whether

$$d' = d - \Delta H_{spin} \geq 0 \quad . \tag{150}$$

If this condition is satisfied, we replace s_x by s'_x and d by d'. Otherwise we keep the old values.

If only such updates are used, the value of H_{comb} stays constant. One simulates the microcanonical ensemble of the combined system. In the limit $N_d \rightarrow \infty$ one recovers the canonical ensemble of the spin system.

A simple way to get the canonical ensemble for the combined system is to alternate the demon updates with standard Metropolis updates. One can also perform a heatbath of the demon to change the value of the Hamiltonian:

$$d' = -\ln(drand48())/\beta \tag{151}$$

Demons can be combined with various algorithms. E.g. M.Creutz has proposed a demonic version of the cluster algorithm: MICROCANONICAL CLUSTER MONTE CARLO, Phys.Rev.Lett.69:1002-1005,1992, hep-lat-9204005

10.5 Hybrid Monte Carlo Algorithm

See also the pages 421-425 in the book "Quantum fields on a lattice" by G.Münster und I.Montvay.

The Hybrid Monte Carlo (HMC) algorithm goes back to
S. Duane, A.D. Kennedy, B.J. Pendleton and D. Roweth, Phys.Lett.B 195 (1987) 216.

For simplicity let us discuss the HMC at the example of the $O(2)$ nonlinear sigma model.

$$S(\{s\}) = -\beta \sum_{<xy>} s_x s_y = -\beta \sum_{<xy>} \cos(\phi_x - \phi_y) \tag{152}$$

Here we have chosen the more field theory like notation $S = \beta H$. The weights for the configurations are given by $\exp(-S)$. The trick now is quite similar to the demon method of M. Creutz:

$$H(\{\phi\},\{p\}) = S(\{\phi\}) + \frac{1}{2}\sum_x p_x^2 \tag{153}$$

where p_x is a real number. In the HMC, the p_x are called conjugate momenta of ϕ_x; However note that they are just auxilliary variables that have no physical meaning at all. The idea of adding the conjugate momenta is to allow for an evolution in Monte Carlo time t that keeps the $H(\{s\},\{p\})$ constant:

$$\frac{\mathrm{d}\phi_x}{\mathrm{d}t} = \frac{\partial H(\{\phi\},\{p\})}{\partial p_x} = p_x \tag{154}$$

$$\frac{\mathrm{d}p_x}{\mathrm{d}t} = -\frac{\partial H(\{\phi\},\{p\})}{\partial \phi_x} \tag{155}$$

for all sites x.

In general the equations of motion can not be solved exactly. Therefore one has to resort to a numerical solution of the partial differential equation. (Remember: Euler method; Higher order schemes; e.g. Runge Kutta).

The numerical integration of the equations of motion lead to systematic errors due to a finite step size δt in the Monte Carlo time. These can be controlled by performing the simulation for various values of δt.

A more radical solution of this problem is provided by the HMC algorithm. Here the numerical solution of the equation of motion is used as proposal in a Metropolis step.

A complete update cycle of the HMC looks as follows:

- Heatbath of the momenta p_x
- Numerical evolution of fields according the equations of motion for some finite time interval.
- Accept/reject step with $A = \min[1, \exp(-H' + H)]$,
  where H is the Hamiltonian for the fields at the beginning of the evolution and H' at the end.

Let us discuss in more detail the first two steps:

In the heatbath step the momenta are generated with a probability distribution proportional to $\exp(-0.5 p_x^2)$. This can be achieved with the so called Box-Muller

method:
(See e.g.)
http://mathworld.wolfram.com/Box-MullerTransformation.html

$$p_1 = \sqrt{-2\ln x_1}\cos(2\pi x_2) \tag{156}$$

$$p_2 = \sqrt{-2\ln x_1}\sin(2\pi x_2) \tag{157}$$

where x_1 and x_2 are statistically independent and are uniformly distributed in the interval 0 to 1. Then p_1 and p_2 have the desired Gaussian distribution, and are statistically independent.

For the equations of motion, two properties have to be satisfied:

1) area conservation
2) reversibility
The simplest scheme that fulfils these properties is the so called leap-frog scheme:

The total length of a trajectory is τ; it is divided into N intervals of the size $\delta\tau$. First a half-step is performed for the momentum

$$p_x(0.5\delta\tau) = p_x(0) - 0.5\left.\frac{H(\{\phi\},\{p\})}{\partial\phi_x}\right|_{t=0}\delta\tau \tag{158}$$

Then follow $N-1$ steps of size $\delta\tau$:

$$\phi(n\delta\tau) = \phi([n-1]\delta\tau) + p([n-0.5])\delta\tau \tag{159}$$

$$p([n+0.5]\delta\tau) = p([n-0.5]\delta\tau) - \left.\frac{H(\{\phi\},\{p\})}{\partial\phi}\right|_{t=n\delta\tau}\delta\tau \tag{160}$$

Finally a full step for ϕ

$$\phi(N\delta\tau) = \phi([N-1]\delta\tau) + p([N-0.5])\delta\tau \tag{161}$$

followed by a half step for the momentum

$$p(N\delta\tau) = p([N-0.5]\delta\tau) - 0.5\left.\frac{H(\{\phi\},\{p\})}{\partial\phi}\right|_{t=N\delta\tau}\delta\tau \tag{162}$$

The integration error for the complete trajectory is $O(\delta\tau^2)$. the role of ϕ and p can be interchanged. As a rule of thumb one often uses $\tau = 1$. The acceptance rate depends on $\delta\tau$. In the limit $\delta\tau \to 0$, the Hamiltonian is conserved, hence the acceptance rate is 1. The optimal acceptance rate is about 80%. On the issue of the acceptance rate see in particular:

S. Gupta, A. Irbäck, F. Karsch and B. Petersson, Phys. Lett. B 242 (1990) 437.
T. Takaishi, Comput.Phys.Commun. 133 (2000) 6, hep-lat/9909134.

For higher order schemes and generalizations of other types see:
J.C. Sexton and D.H. Weingarten, Nucl. Phys. B380 (1992) 665.

A check for the correctness of the implementation is

$$\langle \exp(-H' + H) \rangle = 1 \tag{163}$$

which holds in general for Metropolis like algorithms.

For $L >> \xi$ fluctuations in the Hamiltonian in different regions of the lattice add up almost like statistically independent. Therefore $\Delta H \propto sqrt(V)$ where V is the number of sites. In order to keep the acceptance rate constant, one has to decrease the step size of the leap-frog integration as $\delta\tau \propto V^{1/4}$.

Most recent developements in lattice QCD applications see e.g. and refs. therein: hep-lat/0506011 , HMC algorithm with multiple time scale integration and mass preconditioning, C. Urbach, K. Jansen, A. Shindler, U. Wenger

11 First order phase transitions

Most transitions in nature are of first order. First order means that (in the thermodynamic limit) first derivatives of the free energy density are not continuous at the transition; E.g. there is a latent heat.

Phase transitions of first order appear in many physical contexts. Examples are melting of ice to water or the evaporation of water to steam. Phase transitions of first order also play an important role in cosmology.

As simple examples let us discuss q-state Potts models in 3 dimensions. The Hamiltonian is

$$H = -J \sum_{<xy>} \delta_{\sigma_x,\sigma_y} \quad , \tag{164}$$

where $\sigma \in \{1, 2, ..., q\}$. For $q = 2$ we get the Ising model. In three dimensions, for $q \geq 3$ there is a phase transition of first order. In two dimensions for $q \geq 5$.

A recent large scale Monte Carlo study of the 3-state Potts model is
Three-Dimensional 3-State Potts Model Revisited With New Techniques
Wolfhard Janke (JGU Mainz), Ramon Villanova (UPF Barcelona)
hep-lat/9612008
Nucl.Phys. B489 (1997) 679-696
Here one finds $\beta_c = 0.550565(10)$ for the three state Potts model.

Quantities that characterize a first order phase transtion are the latent heat

$$\Delta E = \lim_{T \searrow T_c} E - \lim_{T \nearrow T_c} E \quad , \tag{165}$$

where E is the energy density in the thermodynamic limit, the correlation lengths ξ in the two phases and the interface tension between the phases. The interface tension of a three dimensional system is defined by

$$\sigma := \lim_{D\to\infty} \lim_{L\to\infty} \frac{1}{L^2} \left(F_{ug} - \frac{1}{2} F_{uu} - \frac{1}{2} F_{gg} \right) \tag{166}$$

where we consider a system of the size $D \times L^2$. In x_2 and x_3 direction there are periodic boundary conditions, while in x_1 direction we have Dirichlet boundary conditions. These boundaries are e.g. ordered (g) or disordered (u). By choosing a disordered boundary on one side of the lattice and disordered on the other, an interface is forced into the system. The temperature is T_c. F denotes the free energy.

A "weak" first order phase transition is characterized by a small latent heat, a small interface tension and large correlation length. If the latent heat and the interface tension vanish, and the correlation lengths diverge, the transition is of second order. In the simulation it is very difficult to discriminate weak first order from second order transitions.

Note however that on the finite lattice we can not get any non-analytic behaviour of the energy density.

As a first exercise, we simulate the 3-state Potts model on a 16^3 lattice using the Metropolis algorithm. To get a first impression, we have plotted the evolution of the energy in figure 10.

We see first signs of metastabilities. The system stays for large Monte Carlo times in a given phase.

The separation of the two phases can be seen even better in a histogram of the energy density. In figure 11 the histogram of the energy density is plotted for the lattice sizes 8^3, 12^3 und 16^3 at β_c.

We see that with increasing system size a clear double peak structure forms. This is a clear sign for a first order phase transition. Sometimes also in the case of a second order transition such a double peak can be seen. For the first order transition it is characteristic that this double peak becomes more and more pronounced as the system size increases.

A phenomenological explanation is that the peaks correspond to configurations, where one phase occupies the whole lattice. The minimum in between the peaks corresponds to configurations where one part can be assigned to one phase and the other part of the lattice to the other phase. In the case of a cubic lattice with periodic boundary conditions, the interface between the phases with minimal area are two planes of the size L^2. In fig. 15 we have sketched this situation.

Based on this scenario we can understand, how the double peak structure becomes more pronounced with increasing L: The ratio of the minimum between the peaks and the height of the peaks behaves as $\exp(-2L^2\sigma)$.

There are corrections to this behaviour; The discussion of these go beyond the scope of the present manuscript. For more see the literature e.g. The volume J.Mod.Phys.C 3 (1992).

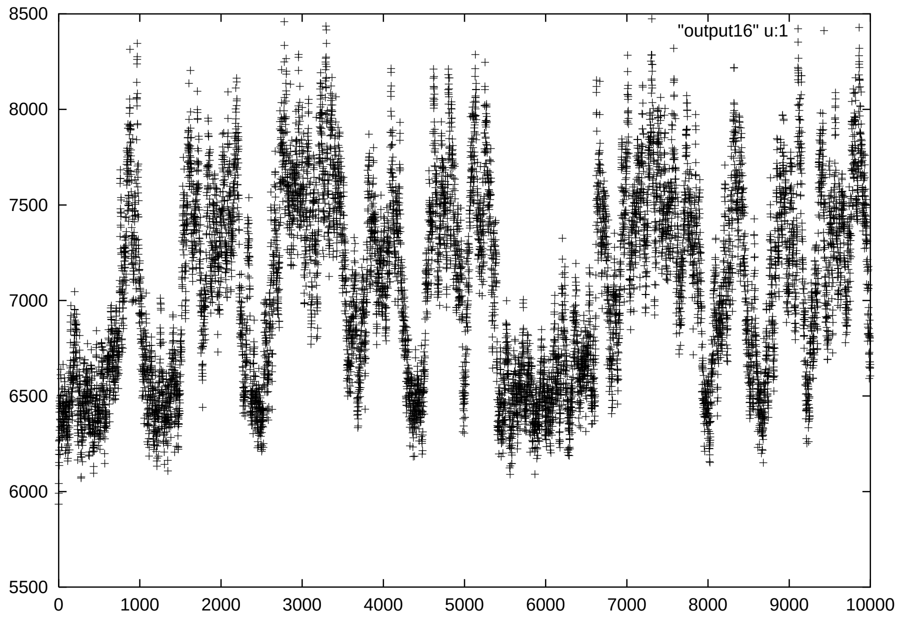


Figure 10:

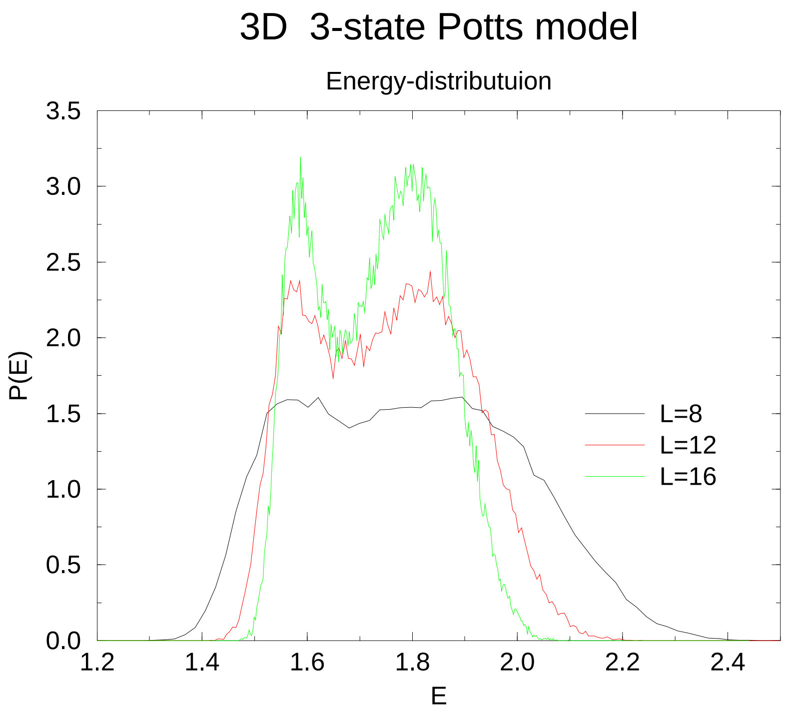


Figure 11:

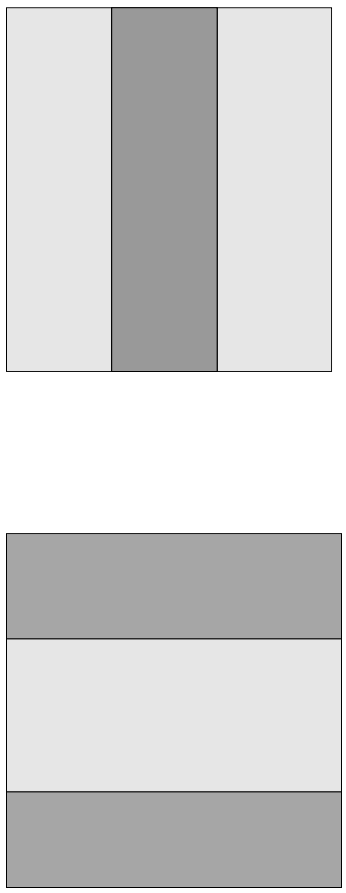


Figure 12:
Schetch of a configuration, where two phases co-exist.

11.1 Determination of the transition temperature

The transition temperature is characterized by the fact that the free energy densities of the different phases are the same.

For finite systems that means that at β_c all phases have the same weight. In the case of Potts models one has to take into account that there are q different ordered phases.

The nontrivial problem is to assign configurations to the phases. As simple criterion is the energy of a configuration. If $E < E_0$, it is assigned to the ordered phase and $E > E_0$ to the disordered one. It remains to chose an E_0.

One possible choice for E_0 is the minumum of $P(E)$ between the peaks.

```
      program betac
        integer ene(200000)
        open(10,file="output")
        read(10,*) L0,L1,L2,betas
        write(*,*) "the number of measurements"
        read*, imax
        write(*,*) "give the number of thermalization steps"
        read*, idis
        imax=imax-idis
        do i=1,idis
           read(10,*) ii
        enddo
        do i=1,imax
           read(10,*) ene(i)
        enddo
        write(*,*) "give E_0"
        read*, itrenn

C       determine beta_c

        db1=0.
        gew1=gewicht(ene,imax,db1,itrenn)
        db2=0.001
        gew2=gewicht(ene,imax,db2,itrenn)

        do i=1,1000
          dsave=db2
          db2=db1+(db2-db1)*(4.-gew1)/(gew2-gew1)
          gew1=gew2
          gew2=gewicht(ene,imax,db2,itrenn)
          db1=dsave
          write(*,*) db2,gew2
```

```
          if(abs(db1-db2).lt.0.0000001) goto 134
        enddo
134     continue

        write(*,*) "betac= ",betas+db2

        end

        real function gewicht(ene,imax,db,itrenn)
        integer ene(200000)
          g1=0.
          g2=0.
          do i=1,imax
            if(ene(i).gt.itrenn)  then
              g2=g2+exp(db*ene(i))
            else
              g1=g1+exp(db*ene(i))
            endif
          enddo
          gewicht=g2/g1
        end
```

For the 4 state Potts model I found $\beta_c \approx 0.6285$, using this method.

11.2 Generating the configurations with a weight different from the Boltzmann factor

In order to reduce systematic errors in the determination of β_c we have to increase the lattice size. Unfortunately, the statistical error for the relative weight of the phases is determined by the number of tunnelings between the phases. Since the probability for mixed phase configurations goes down exponentially with L^2 also the tunneling rate rapidly goes down with increasing lattice size.

A way out of this problem is provided by the proposal of B. Berg and T. Neuhaus Multicanonical Ensemble: A New Approach to Simulate First-order Phase Transitions,
hep-lat/9202004, Phys. Rev. Lett. 68 (1992) 9.

Instead of generating the configurations with the Boltzmann factor as weight we use same modified (artificial) weight:

$$P(X) \propto \exp(-\beta H(X)) \times P_M(H(X)) \quad . \tag{167}$$

Where $P_M(H(X))$ should be chosen such that the minimum between the peaks is lifted. $P_M(H(X))$ has to be determined numerically.

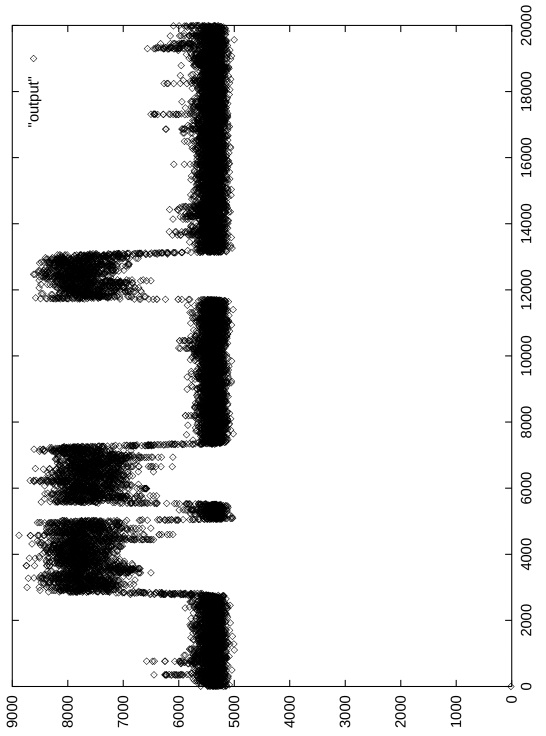


Figure 13: M
onte Carlo time evolution of the energy. Metropolis algorithm, 16^3 lattice 4 state potts model at $\beta = 0.6380$.

The Metropolis algorithm has to be adopted to the modified weight. The proposal for a new value of the spin is done in the same way as before. The modified weight has to be taken into account in the acceptance probability:

$$A(X', X) = \min\left(1\ ,\ \ \exp(-\beta(H(X') - H(X))\frac{P_M(H(X'))}{P_M(H(X))}\right) \quad . \tag{168}$$

Part of a C-program for the Metropolis algorithm with modified weight

```
/* decide whether to accept the proposal */
enerp=ener+dif;
if((p[dif+6]*pm(enerp)/pm(ener))>drand48())
  {
  ener=enerp;
  spin[i0][i1][i2]=pro;
  }
```

To determine pm(ener) a corresponding function has to be written.

```
double pm(int ene)
{
double xx;
xx=1.;
if((ene>5600)&&(ene<=5900))
  {
  xx=1.+19.*(ene-5600)/300. ;
  }
if((ene>5900)&&(ene<=7000))
  {
  xx=20.;
  }
if((ene>7000)&&(ene<=7300))
  {
  xx=1.+19.*(7300-ene)/300.;
  }
return xx;
}
```

To evaluate the expectation values of observables, we have to recover the Boltzmann factor. That looks similar to reweighting in β. If the configurations are generated with $\exp(-\beta H(X)) \times P_M(H(X))$ as relative weight, the estimate of the expectation value of the observable A is given by

$$< A > \approx \frac{\sum_i P_M(H_i)^{-1}\ \ A_i}{\sum_i P_M(H_i)^{-1}} \quad . \tag{169}$$

Figure 14:

Figure 15:

A. Billoire, T. Neuhaus and B. Berg:
A DETERMINATION OF INTERFACE FREE ENERGIES,
hep-lat/9307017, Nucl. Phys. B413 (1994) 795

Related methods:

Simulated Tempering: A New Monte Carlo Scheme,
by Enzo Marinari and Giorgio Parisi,
hep-lat/9205018, Europhys. Lett. 19 (1992) 451-458

Here the temperature (or inverse temperature) is used as dynamical variable. In addition to "standard" update-steps there are updates of the inverse temperature β.

Parallel Tempering:
Optimized Monte Carlo Methods
Enzo Marinari
cond-mat/9612010, Lectures given at the 1996 Budapest Summer School on Monte Carlo Methods.

11.3 Microcanonical ensemble

A differnt approach is the microcanonical simulation with a single demon. One uses the demon

$$H_{comb} \;=\; H_{potts} \;+\; H_{demon} \tag{170}$$

with

$$H_{demon} \;=\; d \;\;, \tag{171}$$

where d is a positive integer, including 0. In the canonical ensemble it is easy to compute the expectation value of the demon:

$$< d >_{kanonisch} (\beta) \;=\; \frac{\sum_{d=0}^{\infty} \; \exp(-\beta \; d) \; d}{\sum_{d=0}^{\infty} \; \exp(-\beta \; d)} \;\;. \tag{172}$$

The larger the lattice, the better plays the spin system the role of a heatbath for the demon. Hence the distribution of the demon is well discribed by a Boltzmann distribution. Therefore we can use eq. (172) to measure the temperature using the demon.

In particular we can determine the temperature as a function of the energy:

a) Start the system with a given value of the energy. This can be achieved e.g. by an ordered spin configuration, and all energy is contained in the demon.

b) Simulate the combined system with the method discussed in chapter 10. After thermalisation measure the demon. From the expectation value of the demon determine the temperature, using eq. (172).

A phase transition of first order is characterised by a constant $\beta(E)$ between E_+ and E_-.

12 The Renormalisation Group

The Renormalisation Group (RG) is the basis of the modern understanding of continuous phase transitions. The renormalisation group explains the appearance of power laws at second order transitions. It allows to derive relations among the critical exponents.

The Renormalisation Group is also the theoretical basis of finite size scaling. For a recent review see:
M.E. Fisher, Renormalization group: Its basis and formulation in statistical physics, Reviews of Modern Physics 70 (1998) 653.
The RG was pioneered by K.G. Wilson: (See z.B. K.G. Wilson and J. Kogut Phys. Rep. 12 (1974) 75).
Textbooks:
Kerson Huang, Statistical Mechanics, second edition
Giorgio Parisi, Statistical Field Theory, Addison Wesley.
Claude Itzykson and Jean-Michel Drouffe, Statistical Field Theory, Volume I, Cambridge University Press.
John Cardy, Scaling and Renormalization in Statistical Physics
C. Domb: The Critical Point, Taylor&Francis
R.J. Creswick, H.A. Farach, C.P. Poole: Introduction to Renormalization Group Methods in Physics, John Wiley and sons
J.J. Binney, N.J.Dowrick, A.J. Fisher, and M.E.J. Newman, The theory of critical phenomena, an introduction to the Renormalization group, Oxford Science publishing.
Nigel Goldenfeld, Lectures on Phase Transitions and the Renormalization Group, Addison-Wesley.

Here I like to give a brief introduction into the RG in real space. Then I will show how RG-Transformations can be performed using Monte Carlo simulations.

As example we make use of the Ising model once more. An explicit example of an RG-transformation in real space is the Blockspin transformation. The idea goes back to L.P. Kadanoff, Physics 2 (1966) 263.

The intension is to reduce the degrees of freedom of the system, while keeping the long range physics intact. In the case of the blockspin transformation, a b^d

sublattice is mapped onto one point of the blocked lattice. On each site of the blocked lattice lives a so called blockspin.

In the case of the Ising model often the so called majority rule is used. If the majority of the spins in the block is 1, the blockspin is equal to 1. If the majority is -1 than the Blockspin is equal to -1. Is the number of sites in the block even, it is possible to have as many spins -1 as 1 in the block. Then, with propability $1/2$ the blockspin is set to 1 or -1.

An other possibility is to chose the block spin not out of $\{-1, 1\}$ but real, e.g.

$$S_X = \frac{1}{b^y} \sum_{x \in X} s_x \quad . \tag{173}$$

Where X denotes a site on the blocked lattice. The exponent y will be explained later. This kind of blocking is called linear, since S_X is a linear function of s_x.

The blockspin transformation induces a weight for the configurations of the block spins.

$$B(S) := \sum_s T(S,s) \exp(-H(s)) \quad . \tag{174}$$

Where $T(S,s)$ gives the block transformation. S is the blockspin configuration and s is the configuration on the original lattice. Here we have changed our convention for the Hamiltonian. β is absorbed into H.

$T(S,s)$ should satisfy the following conditions:

$$\sum_S T(S,s) = 1 \tag{175}$$

and

$$T(S,s) \geq 0 \quad . \tag{176}$$

This allows us to interpret T as a probability (density).

An important property of the transformation is that it exactly keeps the value of the partition function:

$$\begin{aligned} Z' &= \sum_S B(S) \\ &= \sum_S \sum_s T(S,s) \exp(-H(s)) \\ &= \sum_s (\sum_S T(S,s)) \exp(-H(s)) \\ &= \sum_s \exp(-H(s)) = Z \quad . \end{aligned} \tag{177}$$

The correlation length is transformed as

$$\xi' = \xi / b \quad . \tag{178}$$

We recall that the correlation length is determined by the decay of the correlation function. For the correlation function holds:

$$G_{block}(b\,x) \propto G(x) \quad . \tag{179}$$

Also the Binder cumulant stays constant, at least for the linear blocking.

When we write the Boltzmann factor of the blocked system as

$$B(S) = \exp(-H'(S)) \quad , \tag{180}$$

We see that the blocking induces a mapping of the Hamiltonian:

$$H \to H' \tag{181}$$

Let us consider as simple example the 1-dimensional Ising model:
A simple RG-transformation is the so called decimation: We sum over every second spin. Let us study $h = 0$ for simplicity:

$$H \;=\; -\,K \sum_{x=1}^{L} s_x\, s_{x+1} \quad . \tag{182}$$

We use periodic boundary conditions; i.e. $L+1$ are identified 1.
The decimation reduces the number of spins by half: $L' = L/2$. The mapping of the site x of the original lattice and the sites X of the blocked lattice is given by $X = x/2$. The transformation kernel T is given by

$$T(S,s) \;=\; \prod_X \; \delta_{S_X, s_{2X}} \tag{183}$$

Plugging this into eq. (174), we get

$$\begin{aligned}
B(S) \;&=\; \sum_s T(S,s)\; \exp(-H(s)) \\
&=\; \sum_s \prod_X \;\; \delta_{S_X,s_{2X}} \;\; \exp(K \sum_{x=1}^{L} \; s_x\, s_{x+1}) \\
&=\; \sum_{s,\text{x ungerade}} \;\; \prod_X \;\; \exp(K\; s_{2X+1}\; (S_X + S_{X+1})) \\
&=\; \prod_X \sum_{s_{2X+1}=\pm 1} \exp(K\; s_{2X+1}\; (S_X + S_{X+1})) \\
&=\; \prod_X [(\exp(K\; (S_X + S_{X+1})) + \exp(-K\; (S_X + S_{X+1}))] \\
&=\; C^{L'} \exp(K' \;\sum_X S_X S_{X+1}) \quad .
\end{aligned} \tag{184}$$

With $C = 2\;\exp(1/2)\;\cosh(2K)$ and $K' = \frac{1}{2}\;\ln(\cosh(2K))$ C and K' we get

$$\begin{aligned}
C\exp(+K') \;&=\; \exp(2\,K) \;+\; \exp(-2\,K) \\
C\exp(-K') \;&=\; 2 \quad .
\end{aligned} \tag{185}$$

For large K one gets $K' \approx K - \frac{1}{2}\ln(2)$. K becomes smaller under the transformation. Further iterating the tranfrormation, the coupling goes to zero. This way we can see that for any finite K, the 1-D Ising model is disordered. I.e. there is no phase transition.

It is a particular feature of the 1-D model that the Hamiltonian keeps its simple form under the RG-Transformation.

Already in two dimensions things become more complicated. Let us start with a Hamiltonian that contains nearest neighbour interactions only. In order to perform the decimation, we divide the lattice like a checker board. The transformation is done by summing over the spins living on the black sites.

The Hamiltonfunction of the decimated system contains not only the term

$$\sum_{<XY>} S_X S_Y \quad , \tag{186}$$

but also

$$\sum_{[XY]} S_X S_Y \quad , \tag{187}$$

wher $[XY]$ denotes pairs of next to nearest neighbours and

$$\sum_{<XYUV>} S_X S_Y S_U S_V \quad , \tag{188}$$

where $< XYUV >$ are the site belonging to a unit square on the lattice.

For this more complicated Hamiltion it is not possible any more to perform an other decimation step exactly. The number of terms in the Hamiltonian becomes infinite.

In general (also for other transformation rules) the H' will not keep the simple form of the Hamiltonian we have started with. In the case of a blockspin in $\{-1, 1\}$ we get terms of the type

$$\sum_{[x,y]} s_x s_y \tag{189}$$

where now x and y may have any distance. However there are good arguments and results from examples that the strength of the coupling decays exponentially with the distance. But there are also terms with 4 spins as

$$\sum_{[x,y,u,v]} s_x s_y s_u s_v \tag{190}$$

6, 8 and so on. In the case of a non-vanishing external field, there are also terms with one, three, five ... spins. While there is in principle an infinite number of couplings, a finite subset of these couplings should be sufficient to give correctly the physics (i.e. the correlation length) up to tiny corrections.

Associated with each of these terms there is a coupling constant K_α:

$$H = -\sum_{\alpha} K_\alpha S_\alpha \quad . \tag{191}$$

where S_α denontes a terms like (189) or (190) in the Hamiltonian. Using this parametrization, we can associate the Hamiltonian with a vector in the coupling space.

This way the RG transformation becomes a mapping in the space of couplings K_α:

$$\vec{K} \to \vec{K}' \ . \tag{192}$$

Continuous phase are characterised by non-trivial fixed points $\vec{K}^*$. A flow, that starts form the critical surface, runs into the fixed point $\vec{K}^*$.

In order to understand the power laws in the neighbourhood of a continuous transition, it is sufficient to study the Taylor-series of the RG transformation in the neighbourhood of the fixed point $\vec{K}^*$:

$$\vec{K}' - \vec{K}^* = T \ \ (\vec{K} - \vec{K}^*) + ... \tag{193}$$

This requires that the RG-transformation is an analytic function at the fixed point $\vec{K}^*$. This can not be proven in general, however there are rather solid arguments in favour of this fact.

It is instructive to study the eigenvalues and eigenvectors of the transformation kernel T:

$$\lambda_i \Phi_i = \Phi_i T \ . \tag{194}$$

On defines "scaling fields

$$u_i = \Phi_i(\vec{K} - \vec{K}^*) \tag{195}$$

and RG-exponents:

$$b^{y_i} = \lambda_i \ . \tag{196}$$

The RG-exponents are classified as relevant, marginal or irrelevant:
relevant:

$$y_i > 0 \ . \tag{197}$$

I.e. the scaling field increases under RG-transformations.
marginal:

$$y_i = 0 \ . \tag{198}$$

irrelevant:

$$y_i < 0 \ . \tag{199}$$

I.e. the scaling field decreases under RG-transformations.

In the Ising model in 2 and 3 dimensions or in the $O(N)$-invariant non-linear σ-models ($N > 1$) in 3 D there are 2 relevant scaling fields: A temperature-like scaling field u_t and a scaling field u_h that is related with an external field. All other scaling fields are irrelevant. The exponents ν , γ, α, β , δ , η that we have introduced so far can be obtained from the two RG-exponents y_t and y_h.

We start with RG-transformations close to β_c and $h = 0$; we perform n_0 RG-transformations. n_0 is such that the irrelevant scaling fields become negligible, but also u_t and u_h are still small enough that the RG-transformation can be linearly approximated.

Since we have performed a finite number of RG-transformations, u_t and u_h can be written as a Taylor-series in $t = (\beta - \beta_c)/\beta_c$ and h:

$$u_t^{(n_0)} = c_t\, t + O(t^2, h^2) \tag{200}$$

and

$$u_h^{(n_0)} = c_h\, h + O(ht) \ . \tag{201}$$

The upper index n_0 says that the value of u_t and u_h after n_0 RG-transformations is meant. For symmetry reason only terms $O(t^2, h^2)$ and $O(ht)$ appear at the second order.

To derive power laws, we have to consider both the behaviour of scaling fields and of the observables under RG-transformations. As a first quantity let us consider the correlation length at vanishing external field $h = 0$:

After n_0 (and more) transformations ξ depends only (up to tiny corrections due to irrelevant scaling fields) on u_t. After the initial n_0 RG-transfromations, we perform n further ones:

$$b^{-n}\, \xi(b^{n y_t} u_t^{(n_0)}) \ = \ \xi(u_t^{(n_0)}) \ . \tag{202}$$

If we choose a fixed value

$$\tilde{u}_t \ = \ b^{n y_t} u_t^{(n_0)} \ , \tag{203}$$

also $\xi(b^{n y_t} u_t^{(n_0)})$ is fixed. Solving equation (203) with respect to b^n and plug in eq. (202) we get

$$\xi(u_t^{(n_0)}) \ = \ const\ (u_t^{(n_0)})^{-1/y_t} \ . \tag{204}$$

Using eq. (200) we get the desired result:

$$\xi(t) \ = \ f\ t^{-1/y_t}(1 + \ ... \ + \ ct \ ...) \ . \tag{205}$$

Hence $\nu = 1/y_t$. The correction term ct stems from the $O(t^2)$ in eq. (200). Taking into account the scaling field u_3, where y_3 is the largest irrelevant RG exponent, we get

$$\xi(t) \ = \ f\ t^{-1/y_t}(1 + \ at^\theta \ + \ ct \ ...) \ . \tag{206}$$

where $\theta = -y_3/y_t = \omega\nu$.

Other power laws can be derived from the transformation behaviour of the free energy density:

$$f(t,h) \ = \ \frac{1}{V} \ln Z(t,h) \tag{207}$$

e.g.:

$$< m > (t,h) \ = \ \frac{\partial f(t,h)}{\partial h} \tag{208}$$

or

$$\chi(t,h) \;=\; \frac{\partial^2 f(t,h)}{\partial h^2} \quad . \tag{209}$$

The free energy desity transforms as (recall that the partition function stays constant):

$$f' \;=\; b^d \, f \;\; . \tag{210}$$

Now it is convenient to split the free energy density f into a singular and a non-singular part:

$$f \;=\; f_{ns} \;+\; f_s \;\; . \tag{211}$$

The part f_{ns} is needed because of the multiplicative constant in the RG transformation. Analogous to eq. (202) one gets:

$$b^{-nd} f_s(b^{ny_t} u_t^{(n_0)}, b^{ny_h} u_h^{(n_0)}) \;=\; f_s(u_t^{(n_0)}, u_h^{(n_0)}) \;\; . \tag{212}$$

Again we keep $\tilde{u}_t \;=\; b^{ny_t} u_t^{(n_0)}$ fixed and solve with respect to b^n.

We obtain:

$$f_s(u_t, u_h) \;=\; const \; u_t^{d/y_t} \; \tilde{\Phi}(u_h \; u_t^{-y_h/y_t}) \;\; . \tag{213}$$

Inserting h and t:

$$f_s(t,h) \;\propto\; t^{d/y_t} \; \Phi(h \; t^{-y_h/y_t}) \tag{214}$$

Takeing the derivative with respect to h at $h = 0$ we get the power laws for the magnetisation and the magnetic susceptibility.

$$\beta = \frac{d - y_h}{y_t} \tag{215}$$

and

$$\gamma = \frac{2y_h - d}{y_t} \quad . \tag{216}$$

As a simple example for Finite Size Scaling we discuss the behaviour of the Binder cumulant at vanishing external field $h = 0$. By construction, the Binder cumulant is a function of L and u_t.

$$U(b^{-n}L, b^{ny_t} u_t) = U(L, u_t) \;\; . \tag{217}$$

We see immediately that at the critical point ($u_t = 0$) the Binder cumulant U does not depend on the lattice size L. We take the derivative of U with respect to t and hence with respect to u_t at the critical point. We get:

$$b^{ny_t} U'(L,0) = U'(b^n L, 0) \tag{218}$$

The derivative grows with the system size as

$$U'(L,0) \propto L^{y_t} \;\; . \tag{219}$$

For further relations among the exponents see the literature, e.g.: J.Cardy, Scaling and Renormalization in Statistical Physics pages 43-46.

12.1 RG-Transformations using Monte Carlo simulations

There are various methods to explicitly compute RG-transformations in some approximation. Most analytic calulations are done in momentum space. Often perturbation theory, starting from the free field theory is employed.

In real-space, RG-transformations can be exactly computed easily for 1-dimensional models. However already in the case of two-dimensional models, (even in the case of the Ising model) approximations and a truncation of the coupling space is needed.

In the early days of the Renormalization group, there are a number of papers in which RG-transformations are computed for the 2D Ising model. To this end, either exact calculations for finite (small) systems or a cumulant-expansion had been used. In introduction to such methods can be found in chapter 5.9 of
J.J. Binney et al. :
The Theory of Critical Phenomena, An Introduction to the Renormalization Group. Oxford Science publications.

Also the Monte Carlo simulation can be employed as computational tool for the RG-transformation. In the literature one finds essentiall two approaches:

- Calculation of the matrix T
- Calculation of the full RG-transformation

12.1.1 Computation the matrix T

In the literature this is called "Monte Carlo Renormalization Group"

References are:

S.-K. Ma, Phys. Rev. Lett. 37 (1976) 461.

More recent refs. for the MCRG are:

C.F. Baillie, R. Gupta, K.A. Hawick, and G.S. Pawley, Phys. Rev. B 45 (1992) 10438, and references

Rajan Gupta, Pablo Tamayo
Critical Exponents of the 3-D Ising Model
cond-mat/9601048

H. W. J. Blöte, J. R. Heringa, A. Hoogland, E. W. Meyer, T. S. Smit
Monte Carlo Renormalization of the 3-D Ising model: Analyticity and Convergence
cond-mat/9602020

Example: 3 D Ising model
The model is simulated at β_c with the standard nearest neighbour Hamiltonian (exception: Blöte et al., who also have next to nearest and next to next to nearest couplings) on on a lattice of the size L^3. The configurations that are generated are

blocked sucessively to lattices of the size $L/2$, $L/4$, ... and so on, until a minimum lattice size L_{min} is reached. I.e., the larger the original lattice, the more block-transformations can be performed and hence the better $\vec{K}^*$ can be approached.

As transformation, in most of the works in the literature, the majority rule is used.

Let us assume that the Hamiltonian has the form

$$H = -\sum_{\alpha} K_\alpha S_\alpha \tag{220}$$

In the literature (e.g. Gupta, Tamayo) Ansatze with 53 even and 46 odd terms are used.

Lets add an other index to K_α to indicate the level of blocking: $K_\alpha^{(i)}$.

In order to determine T, we have to determine the partial derivatives

$$T_{\alpha,\beta}^{(i)} = \frac{\partial K_\alpha^{(i+1)}}{\partial K_\beta^{(i)}} \tag{221}$$

This can be be done using certain expectation values (See S.-K. Ma) We define

$$A_{\alpha,\beta}^{(i)} := \frac{\partial < S_\alpha^{(i+1)} >}{\partial K_\beta^{(i)}} =< S_\alpha^{(i+1)} S_\beta^{(i)} > \; - \; < S_\alpha^{(i+1)} >< S_\beta^{(i)} > \tag{222}$$

and

$$B_{\alpha,\beta}^{(i)} := \frac{\partial < S_\alpha^{(i+1)} >}{\partial K_\beta^{(i+1)}} =< S_\alpha^{(i+1)} S_\beta^{(i+1)} > \; - \; < S_\alpha^{(i+1)} >< S_\beta^{(i+1)} > \quad . \tag{223}$$

Hence T can be obtained by solving

$$A = T \; B \quad . \tag{224}$$

The eigenvalues y_t and y_h can be obtained by diagonalization of T. The most important problem of this approach is that the space of couplings K has to be truncated. It is not fully understood how the the space of coupling can be systematically enlarged to to get y_t and y_h with smaller and smaller sytematic errors. (Proposal of Blöte et al.: Importance index that takes into account the number of spins and their mutual distance.)

12.1.2 Calculation of a RG transformation

There are various ways to perform the RG-transformation $\vec{K} \to \vec{K}'$ using Monte Carlo simulations. Here I like to discuss the most simple method. This method is based on demons.

References:

M. Creutz, A. Gocksch and M. Ogilvie, Phys.Rev.Lett. 53 (1984) 875.

M. Hasenbusch, K. Pinn, and C. Wieczerkowski
Canonical Demon Monte Carlo Renormalization Group
Phys. Lett. B338 (1994) 308, hep-lat/9406019.

The demon that haere used here take real values in the interval $[0, d_{max}]$. The expectation value of the demon as a function of the coupling can be easily evaluated:

$$< d >= \int_0^{d_{max}} \mathrm{d}d \exp(-Kd) d = \frac{\exp(-d_{max}K)(-d_{max}K - 1) + 1}{K^2} \quad . \tag{225}$$

Knowing $< d >$ from the simulation, we can solve the equation above with respect to K.

The demons have to be equlibrated with the blocked system. At the level of the Hamiltonian this looks as follwows:

$$H_{comb} = -\sum_\alpha K'_\alpha (S_\alpha - d_\alpha) \quad . \tag{226}$$

To exchange energy between the blocked sytem and the demons, we perform the following update:

1) First a new spin s' is proposed.

2) The corresponding change ΔS_α in all terms of the Hamiltonian is calculated der Hamiltonfunktion berechnet. It follows $d'_\alpha = d_\alpha + \Delta S_\alpha$.

3) If for all α, $0 < d'_\alpha < d_{\alpha,max}$ is satisfied, the proposal for s' and d'_α are accepted.

Acknowledgements. I like to thank Francesco Basile for proof-reading of the manuscript.