

INTRODUCTION TO MARKOV CHAIN MONTE CARLO SIMULATIONS AND THEIR STATISTICAL ANALYSIS

Bernd A. Berg

*Department of Physics
Florida State University
Tallahassee, Florida 32306-4350, USA*
and
*School of Computational Science
Florida State University
Tallahassee, Florida 32306-4120, USA*
E-mail: berg@csit.fsu.edu

This article is a tutorial on Markov chain Monte Carlo simulations and their statistical analysis. The theoretical concepts are illustrated through many numerical assignments from the author's book [7] on the subject. Computer code (in Fortran) is available for all subjects covered and can be downloaded from the web.

Contents

1	Introduction	2
2	Probability Distributions and Sampling	3
3	Random Numbers and Fortran Code	4
3.1	How to Get and Run the Fortran Code	5
4	Confidence Intervals and Heapsort	6
5	The Central Limit Theorem and Binning	8
6	Gaussian Error Analysis for Large and Small Samples	11
6.1	χ^2 Distribution, Error of the Error Bar, F-Test	15
6.2	The Jackknife Approach	16
7	Statistical Physics and Potts Models	17
8	Sampling and Re-weighting	19
9	Importance Sampling and Markov Chain Monte Carlo	21
9.1	Metropolis and Heat Bath Algorithm for Potts Models	23
9.2	The $O(3)$ σ Model and the Heat Bath Algorithm	25

9.3 Example Runs	26
10 Statistical Errors of Markov Chain Monte Carlo Data	30
10.1 Autocorrelations	31
10.2 Integrated Autocorrelation Time and Binning	33
10.3 Illustration: Metropolis Generation of Normally Distributed Data	34
11 Self-Consistent versus Reasonable Error Analysis	37
12 Comparison of Markov Chain MC Algorithms	38
13 Multicanonical Simulations	40
13.1 How to Get the Weights?	43
14 Multicanonical Example Runs ($2d$ Ising and Potts Models)	44
14.1 Energy and Specific Heat Calculation	46
14.2 Free Energy and Entropy Calculation	48
14.3 Time Series Analysis	49
References	51

1. Introduction

Markov chain Monte Carlo (MC) simulations started in earnest with the 1953 article by Nicholas Metropolis, Arianna Rosenbluth, Marshall Rosenbluth, Augusta Teller and Edward Teller [18]. Since then MC simulations have become an indispensable tool with applications in many branches of science. Some of those are reviewed in the proceedings [13] of the 2003 Los Alamos conference, which celebrated the 50th birthday of Metropolis simulations.

The purpose of this tutorial is to provide an overview of basic concepts, which are prerequisites for an understanding of the more advanced lectures of this volume. In particular the lectures by Prof. Landau are closely related.

The theory behind MC simulations is based on statistics and the analysis of MC generated data is applied statistics. Therefore, statistical concepts are reviewed first in this tutorial. Nowadays abundance of computational power implies also a paradigm shift with respect to statistics: Computationally intensive, but conceptually simple, methods belong at the forefront. MC simulations are not only relevant for simulating models of interest, but they constitute also a valuable tool for approaching statistics.

The point of departure for treating Markov chain MC simulations is the Metropolis algorithm for simulating the Gibbs canonical ensemble. The heat bath algorithm follows. To illustrate these methods our systems of choice are discrete Potts and continuous $O(n)$ models. Both classes of models are programmed for arbitrary dimensions ($d = 1, 2, 3, 4, \dots$). On the advanced

side we introduce multicanonical simulations, which cover an entire temperature range in a single simulation, and allow for direct calculations of the entropy and free energy.

In summary, we consider Statistics, Markov Chain Monte Carlo simulations, the Statistical Analysis of Markov chain data and, finally, Multicanonical Sampling. This tutorial is abstracted from the author's book on the subject [7]. Many details, which are inevitably omitted here, can be found there.

2. Probability Distributions and Sampling

A **sample space** is a set of points or elements, in natural sciences called **measurements** or **observations**, whose occurrence depends on chance. Carrying out independent repetitions of the same experiment is called **sampling**. The outcome of each experiment provides an event called data point. In N such experiments we may find the event A to occur with **frequency** n , $0 \leq n \leq N$. The **probability** assigned to the event A is a number $P(A)$, $0 \leq P(A) \leq 1$, so that

$$P(A) = \lim_{N \rightarrow \infty} \frac{n}{N}. \quad (1)$$

This equation is sometimes called the **frequency definition of probability**.

Let us denote by $P(a, b)$ the probability that $x^r \in [a, b]$ where x^r is a continuous **random variable** drawn in the interval $(-\infty, +\infty)$ with the **probability density** $f(x)$. Then,

$$P(a, b) = \int_a^b f(x) dx. \quad (2)$$

Knowledge of all probabilities $P(a, b)$ implies

$$f(x) = \lim_{y \rightarrow x^-} \frac{P(y, x)}{x - y} \geq 0. \quad (3)$$

The **(cumulative) distribution function** of the random variable x^r is defined as

$$F(x) = P(x^r \leq x) = \int_{-\infty}^x f(x) dx. \quad (4)$$

A particularly important case is the **uniform probability distribution** for random numbers between $[0, 1)$,

$$u(x) = \begin{cases} 1 & \text{for } 0 \leq x < 1; \\ 0 & \text{elsewhere.} \end{cases} \quad (5)$$

Remarkably, the uniform distribution allows for the construction of general probability distributions. Let

$$y = F(x) = \int_{-\infty}^x f(x') dx'$$

and assume that the inverse $x = F^{-1}(y)$ exists. For y^r being a uniformly distributed random variable in the range $[0, 1)$ it follows that

$$x^r = F^{-1}(y^r) \quad (6)$$

is distributed according to the probability density $f(x)$.

The **Gaussian** or **normal distribution** is of major importance. Its probability density is

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/(2\sigma^2)} \quad (7)$$

where σ^2 is the **variance** and $\sigma > 0$ the **standard deviation**. The Gaussian distribution function $G(x)$ is related to that of variance $\sigma^2 = 1$ by

$$\begin{aligned} G(x) &= \int_{-\infty}^x g(x') dx' = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x/\sigma} e^{-(x'')^2/2} dx'' \\ &= \frac{1}{2} + \frac{1}{2} \operatorname{erf} \left(\frac{x}{\sigma\sqrt{2}} \right). \end{aligned} \quad (8)$$

In principle we could now generate **Gaussian random numbers** according to Eq. (6). However, the numerical calculation of the inverse error function is slow and makes this an impractical procedure. Much faster is to express the product probability density of two independent Gaussian distributions in polar coordinates

$$\frac{1}{2\pi\sigma^2} e^{-x^2/(2\sigma^2)} e^{-y^2/(2\sigma^2)} dx dy = \frac{1}{2\pi\sigma^2} e^{-r^2/(2\sigma^2)} d\phi r dr,$$

and to use the relations

$$x^r = r^r \cos \phi^r \quad \text{and} \quad y^r = r^r \sin \phi^r. \quad (9)$$

3. Random Numbers and Fortran Code

According to Marsaglia and collaborators [17] a list of desirable properties for (pseudo) random number generators is:

- (i) *Randomness*. The generator should pass stringent tests for randomness.
- (ii) *Long period*.
- (iii) *Computational efficiency*.

- (iv) *Repeatability*. Initial conditions (seed values) completely determine the resulting sequence of random variables.
- (v) *Portability*. Identical sequences of random variables may be produced on a wide variety of computers (for given seed values).
- (vi) *Homogeneity*. All subsets of bits of the numbers are random.

Physicists have added a number of their applications as new tests (e.g., see [22] and references therein). In our program package a version of the random number generator of Marsaglia and collaborators [17] is provided. Our corresponding Fortran code consists of three subroutines:

`rmaset.f` to set the initial state of the random number generator.
`ranmar.f` which provides one random number per call.
`rmasave.f` to save the final state of the generator.

In addition, `rmafun.f` is a Fortran function version of `ranmar.f` and calls to these two routines are freely interchangeable. Related is also the subroutine `rmagau.f`, which generates two Gaussian random numbers.

The subroutine `rmaset.f` initializes the generator to mutually independent sequences of random numbers for distinct pairs of

$$-1801 \leq \text{iseed1} \leq 29527 \quad \text{and} \quad -9373 \leq \text{iseed2} \leq 20708 . \quad (10)$$

This property makes the generator quite useful for parallel processing.

3.1. How to Get and Run the Fortran Code

To **download** the Fortran code visit the website

<http://www.worldscibooks.com/physics/5602.html>

click the download link and follow the instructions given there. If the above link should be unavailable, visit the author's homepage which is presently located at

<http://www.hep.fsu.edu/~berg> .

After installation the directory tree shown in Fig. 1 is obtained. `ForLib` contains a library of functions and subroutines which is closed in the sense that no reference to non-standard functions or subroutines outside the library is ever made. Fortran programs are contained in the folder `ForProg` and procedures for interactive use in `ForProc`. It is **recommended** to leave the hyperstructure of program dependencies introduced between the levels of the STMC directory tree intact. Otherwise, complications may result which require advanced Fortran skills.

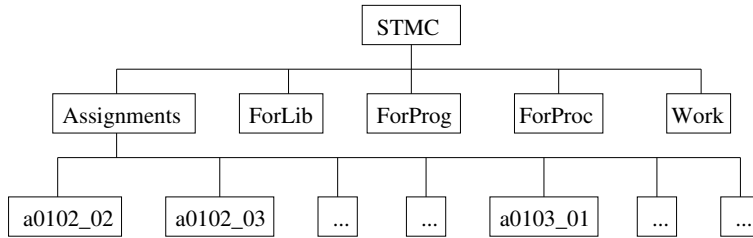


Fig. 1. The Fortran routines are provided and prepared to run in the tree structure of folders depicted in this figure. This tree unfolds from the downloaded file.

Assignment: Marsaglia random numbers. Run the program `mar.f` to reproduce the following results:

RANMAR INITIALIZED.	MARSAGLIA CONTINUATION.
idat, xr = 1 0.116391063	idat, xr = 1 0.495856345
idat, xr = 2 0.96484679	idat, xr = 2 0.577386141
idat, xr = 3 0.882970393	idat, xr = 3 0.942340136
idat, xr = 4 0.420486867	idat, xr = 4 0.243162394
extra xr = 0.495856345	extra xr = 0.550126791

Understand how to re-start the random number generator and how to perform different starts when the continuation data file `ranmar.d` does not exist. You find `mar.f` in `ForProg/Marsaglia` and it includes subroutines from `ForLib`. To compile properly, `mar.f` has to be located two levels down from a root directory `STMC`. The solution is given in the folder `Assignments/a0102.02`.

4. Confidence Intervals and Heapsort

Let a distribution function $F(x)$ and q , $0 \leq q \leq 1$ be given. One defines **q-tiles** (also called **quantiles** or **fractiles**) x_q by means of

$$F(x_q) = q. \quad (11)$$

The **median** $x_{\frac{1}{2}}$ is often (certainly not always) the **typical** value of the random variable x^r .

Example: For the normal distribution the precise probability content of the confidence intervals

$$[x_q, x_{1-q}] = [-n\sigma, n\sigma] \text{ for } n = 1, 2$$

is $p = 1 - 2q = 68.27\%$ for one σ and $p = 1 - 2q = 95.45\%$ for two σ .

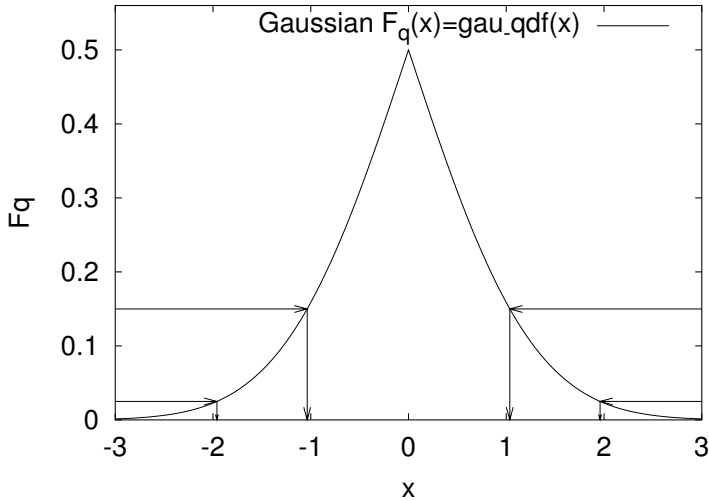


Fig. 2. Gaussian peaked distribution function and estimates of x_q for the 70% (approximately 1σ) and 95% (approximately 2σ) confidence intervals.

The peaked distribution function

$$F_q(x) = \begin{cases} F(x) & \text{for } F(x) \leq \frac{1}{2}, \\ 1 - F(x) & \text{for } F(x) > \frac{1}{2}. \end{cases} \tag{12}$$

provides a useful way to visualize probability intervals of a distribution. It is illustrated in Fig. 2 for the Gaussian distribution.

Sampling provides us with an empirical distribution function and in practice the problem is to estimate confidence intervals from the empirical data. Assume we generate n random numbers x_1, \dots, x_n independently according to a probability distribution $F(x)$. The n random numbers constitute a **sample**. We may re-arrange the x_i in increasing order. Denoting the smallest value by x_{π_1} , the next smallest by x_{π_2} , etc., we arrive at

$$x_{\pi_1} \leq x_{\pi_2} \leq \dots \leq x_{\pi_n} \tag{13}$$

where π_1, \dots, π_n is a permutation of $1, \dots, n$. Each of the x_{π_i} is called an **order statistic**. An estimator for the distribution function $F(x)$ is the **empirical distribution function**

$$\bar{F}(x) = \frac{i}{n} \quad \text{for } x_{\pi_i} \leq x < x_{\pi_{i+1}}, \quad i = 0, 1, \dots, n-1, n \tag{14}$$

with the definitions $x_{\pi_0} = -\infty$ and $x_{\pi_{n+1}} = +\infty$.

To calculate $\overline{F}(x)$ and the corresponding peaked distribution function, one needs an efficient way to **sort** n data values in ascending (or descending) order. This is provided by the **heapsort**, which relies on two steps: First the data are arranged in a heap, then the heap is sorted. A **heap** is a partial ordering so that the number at the top is larger or equal than the two numbers in the second row, provided at least three numbers x_i exist. More details are given in [7]. The computer time needed to succeed with this sorting process grows only like $n \log_2 n$, because there are $\log_2 n$ levels in the heap, see Knuth [15] for an exhaustive discussion of sorting algorithms.

5. The Central Limit Theorem and Binning

How is the sum of two independent random variables

$$y^r = x_1^r + x_2^r . \quad (15)$$

distributed? We denote their probability density of y^r by $g(y)$. The corresponding cumulative distribution function is given by

$$G(y) = \int_{x_1+x_2 \leq y} f_1(x_1) f_2(x_2) dx_1 dx_2 = \int_{-\infty}^{+\infty} f_1(x) F_2(y-x) dx$$

where $F_2(x)$ is the distribution function of the random variable x_2^r . We take the derivative and obtain the probability density of y^r

$$g(y) = \frac{dG(y)}{dy} = \int_{-\infty}^{+\infty} f_1(x) f_2(y-x) dx . \quad (16)$$

The probability density of a sum of two independent random variables is the **convolution of the probability densities** of these random variables.

Example: Sums of uniform random numbers, corresponding to the sums of an uniformly distributed random variable $x^r \in (0, 1]$:

(a) Let $y^r = x^r + x^r$, then

$$g_2(y) = \begin{cases} y & \text{for } 0 \leq y \leq 1, \\ 2-y & \text{for } 1 \leq y \leq 2, \\ 0 & \text{elsewhere.} \end{cases} \quad (17)$$

(b) Let $y^r = x^r + x^r + x^r$, then

$$g_3(y) = \begin{cases} y^2/2 & \text{for } 0 \leq y \leq 1, \\ (-2y^2 + 6y - 3)/2 & \text{for } 1 \leq y \leq 2, \\ (y-3)^2/2 & \text{for } 2 \leq y \leq 3, \\ 0 & \text{elsewhere.} \end{cases} \quad (18)$$

The convolution (16) takes on a simple form in **Fourier space**. In statistics the **Fourier transformation** of the probability density is known as **characteristic function**, defined as the expectation value of e^{itx^r} :

$$\phi(t) = \langle e^{itx^r} \rangle = \int_{-\infty}^{+\infty} e^{itx} f(x) dx . \tag{19}$$

A straightforward calculation gives

$$\phi(t) = \exp \left[-\frac{1}{2} \frac{\sigma_x^2}{N} t^2 \right] \tag{20}$$

for the characteristic function of the Gaussian probability density (7). The characteristic function is particularly useful for investigating sums of random variables, $y^r = x_1^r + x_2^r$:

$$\begin{aligned} \phi_y(t) &= \langle e^{(itx_1^r + itx_2^r)} \rangle \tag{21} \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} e^{itx_1} e^{itx_2} f_1(x_1) f_2(x_2) dx_1 dx_2 = \phi_{x_1}(t) \phi_{x_2}(t) . \end{aligned}$$

The characteristic function of a sum of random variables is the product of their characteristic functions. The result generalizes immediately to N random variables $y^r = x_1^r + \dots + x_N^r$. The characteristic function of y^r is

$$\phi_y(t) = \prod_{i=1}^N \phi_{x_i}(t) \tag{22}$$

and the probability density of y^r is the Fourier back-transformation of this characteristic function

$$g(y) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} dt e^{-ity} \phi_y(t) . \tag{23}$$

The **probability density of the sample mean** is obtained as follows: The arithmetic mean of y^r is $\bar{x}^r = y^r/N$. We denote the probability density of y^r by $g_N(y)$ and the probability density of the arithmetic mean by $\hat{g}_N(\bar{x})$. They are related by

$$\hat{g}_N(\bar{x}) = N g_N(N\bar{x}) . \tag{24}$$

This follows by substituting $y = N\bar{x}$ into $g_N(y) dy$:

$$1 = \int_{-\infty}^{+\infty} g_N(y) dy = \int_{-\infty}^{+\infty} g_N(N\bar{x}) 2d\bar{x} = \int_{-\infty}^{+\infty} \hat{g}_N(\bar{x}) d\bar{x} .$$

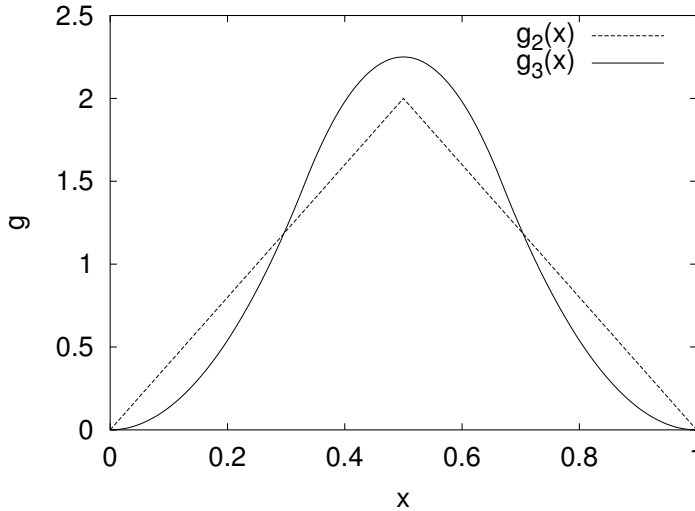


Fig. 3. Probability densities for the arithmetic means of two and three uniformly distributed random variables, $\hat{g}_2(\bar{x})$ and $\hat{g}_3(\bar{x})$, respectively.

Fig. 3 illustrates equation (24) for the sums of two (17) and three (18) uniformly distributed random variables. This suggests that sampling leads to convergence of the mean by reducing its variance. We use the characteristic function $\phi_y(t) = [\phi_x(t)]^N$ to understand the general behavior. The characteristic function for the corresponding arithmetic average is

$$\phi_{\bar{x}}(t) = \int_{-\infty}^{+\infty} d\bar{x} e^{it\bar{x}} \hat{g}_N(\bar{x}) = \int_{-\infty}^{+\infty} dy \exp\left(i \frac{t}{N} y\right) g_N(y) .$$

Hence,

$$\phi_{\bar{x}}(t) = \phi_y\left(\frac{t}{N}\right) = \left[\phi_x\left(\frac{t}{N}\right)\right]^N . \quad (25)$$

To simplify the equations we restrict ourselves to $\hat{x} = 0$. Let us consider a probability density $f(x)$ and assume that its moment exists, implying that the characteristic function is at least two times differentiable, so that

$$\phi_x(t) = 1 - \frac{\sigma_x^2}{2} t^2 + \mathcal{O}(t^3) . \quad (26)$$

The leading term reflects the normalization of the probability density and the first moment is $\phi'(0) = \hat{x} = 0$. The characteristic function of the mean becomes

$$\phi_{\bar{x}}(t) = \left[1 - \frac{\sigma_x^2}{2N^2}t^2 + \mathcal{O}\left(\frac{t^3}{N^3}\right) \right]^N = \exp\left[-\frac{1}{2}\frac{\sigma_x^2}{N}t^2\right] + \mathcal{O}\left(\frac{t^3}{N^2}\right).$$

This is the **central limit theorem**: The probability density of the arithmetic mean \bar{x}^r converges towards the Gaussian probability density with variance (compare Eq. (20))

$$\sigma^2(\bar{x}^r) = \frac{\sigma^2(x^r)}{N}. \tag{27}$$

Binning: The notion of binning introduced here should not be confused with histogramming. Binning means here that we group NDAT data into NBINS bins, where each binned data point is the arithmetic average of

$$\text{NBIN} = \lfloor \text{NDAT}/\text{NBINS} \rfloor \quad (\text{Fortran integer division})$$

data points in their original order. Preferably NDAT is a multiple of NBINS. The purpose of the binning procedure is twofold:

- (1) When the the central limit theorem applies, the binned data will become practically Gaussian, as soon as NBIN becomes large enough. This allows to apply Gaussian error analysis methods even when the original data are not Gaussian.
- (2) When data are generated by a Markov process subsequent events are correlated. For binned data these correlations are reduced and can in practical applications be neglected, once NBIN is sufficiently large compared to the autocorrelation time (see section 10).

6. Gaussian Error Analysis for Large and Small Samples

The central limit theorem underlines the importance of the normal distribution. Assuming we have a large enough sample, the arithmetic mean of a suitable expectation value becomes normally distributed and the calculation of the confidence intervals is reduced to studying the normal distribution. It has become the convention to use the **standard deviation** of the sample mean

$$\sigma = \sigma(\bar{x}^r) \quad \text{with} \quad \bar{x}^r = \frac{1}{N} \sum_{i=1}^N x_i^r \tag{28}$$

and its confidence intervals $[\hat{x} - n\sigma, \hat{x} + n\sigma]$ (the dependence of σ on N is suppressed). For a Gaussian distribution equation Eq. (8) yields the probability content p of the confidence intervals (28) to be

$$p = p(n) = G(n\sigma) - G(-n\sigma) = \frac{1}{\sqrt{2\pi}} \int_{-n}^{+n} dx e^{-\frac{1}{2}x^2} = \operatorname{erf}\left(\frac{n}{\sqrt{2}}\right). \quad (29)$$

In practice the roles of \bar{x} and \hat{x} are interchanged: One would like to know the likelihood that the **unknown** exact expectation value \hat{x} will be in a certain confidence interval around the measured sample mean. The relationship

$$\bar{x} \in [\hat{x} - n\sigma, \hat{x} + n\sigma] \iff \hat{x} \in [\bar{x} - n\sigma, \bar{x} + n\sigma] \quad (30)$$

solves the problem. Conventionally, these estimates are quoted as

$$\hat{x} = \bar{x} \pm \Delta\bar{x} \quad (31)$$

where the **error bar** $\Delta\bar{x}$ is often an **estimator** of the exact standard deviation.

An obvious estimator for the variance σ_x^2 is

$$(s'_x)^2 = \frac{1}{N} \sum_{i=1}^N (x_i^r - \bar{x}^r)^2 \quad (32)$$

where the prime indicates that we shall not be happy with it, because we encounter a **bias**. An estimator is said to be biased when its expectation value does not agree with the exact result. In our case

$$\langle (s'_x)^2 \rangle \neq \sigma_x^2. \quad (33)$$

An estimator whose expectation value agrees with the true expectation value is called **unbiased**. The bias of the definition (32) comes from replacing the exact mean \hat{x} by its estimator \bar{x}^r . The latter is a random variable, whereas the former is just a number. Some algebra [7] shows that the desired **unbiased estimator of the variance** is given by

$$(s_x^r)^2 = \frac{N}{N-1} (s'_x)^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i^r - \bar{x}^r)^2. \quad (34)$$

Correspondingly, the unbiased estimator of the variance of the sample mean is

$$(s_{\bar{x}}^r)^2 = \frac{1}{N(N-1)} \sum_{i=1}^N (x_i^r - \bar{x}^r)^2. \quad (35)$$

Gaussian difference test: In practice one is often faced with the problem to compare two different empirical estimates of some mean. How large must $D = \bar{x} - \bar{y}$ be in order to indicate a real difference? The quotient

$$d^r = \frac{D^r}{\sigma_D}, \quad \sigma_D = \sqrt{\sigma_x^2 + \sigma_y^2} \tag{36}$$

is normally distributed with expectation zero and variance one, so that

$$P = P(|d^r| \leq d) = G_0(d) - G_0(-d) = \operatorname{erf}\left(\frac{d}{\sqrt{2}}\right). \tag{37}$$

The **likelihood that the observed difference $|\bar{x} - \bar{y}|$ is due to chance** is defined to be

$$Q = 1 - P = 2G_0(-d) = 1 - \operatorname{erf}\left(\frac{d}{\sqrt{2}}\right). \tag{38}$$

If the assumption is correct, then Q is a uniformly distributed random variable in the range $[0, 1]$. Examples are collected in table 1. Often a 5% cut-off is used to indicate a real discrepancy.

Table 1. Gaussian difference tests (compile and run the program provided in ForProc/Gau_dif, which results in an interactive dialogue).

$\bar{x}_1 \pm \sigma_{\bar{x}_1}$	1.0 ± 0.1	1.0 ± 0.1	1.0 ± 0.1	1.0 ± 0.05	1.000 ± 0.025
$\bar{x}_2 \pm \sigma_{\bar{x}_2}$	1.2 ± 0.2	1.2 ± 0.1	1.2 ± 0.0	1.2 ± 0.00	1.200 ± 0.025
Q	0.37	0.16	0.046	0.000063	0.15×10^{-7}

Gosset’s Student Distribution: We ask the question: What happens with the Gaussian confidence limits when we replace the variance σ_x^2 by its estimator s_x^2 in statements like

$$\frac{|\bar{x} - \hat{x}|}{\sigma_{\bar{x}}} < 1.96 \text{ with } 95\% \text{ probability.}$$

For sampling from a Gaussian distribution the answer was given by Gosset, who published his article 1908 under the pseudonym *Student* in *Biometrika* [20]. He showed that the distribution of the random variable

$$t^r = \frac{\bar{x}^r - \hat{x}}{s_{\bar{x}}^r} \tag{39}$$

is given by the probability density

$$f(t) = \frac{1}{(N - 1) B(1/2, (N - 1)/2)} \left(1 + \frac{t^2}{N - 1}\right)^{-\frac{N}{2}}. \tag{40}$$

Here $B(x, y)$ is the beta function. The fall-off is a power law $|t|^{-N}$ for $|t| \rightarrow \infty$, instead of the exponential fall-off of the normal distribution. Some confidence probabilities of the Student distribution are (assignment a0203_01):

N \ S	1.0000	2.0000	3.0000	4.0000	5.0000
2	.50000	.70483	.79517	.84404	.87433
3	.57735	.81650	.90453	.94281	.96225
4	.60900	.86067	.94233	.97199	.98461
8	.64938	.91438	.98006	.99481	.99843
16	.66683	.93605	.99103	.99884	.99984
32	.67495	.94567	.99471	.99963	.99998
64	.67886	.95018	.99614	.99983	1.0000
INFINITY:	.68269	.95450	.99730	.99994	1.0000

For $N \leq 4$ we find substantial deviations from the Gaussian confidence levels, whereas up to two standard deviations reasonable approximations of Gaussian confidence limits are obtained for $N \geq 16$ data. If desired, the Student distribution function can always be used to calculate the exact confidence limits. When the central limit theorem applies, we can bin a large set of non-Gaussian data into 16 almost Gaussian data to reduce the error analysis to Gaussian methods.

Student difference test: This test is a generalization of the Gaussian difference test. It takes into account that only a finite number of events are sampled. As before it is assumed that the events are drawn from a normal distribution. Let the following data be given

$$\bar{x} \text{ calculated from } M \text{ events, i.e., } \sigma_{\bar{x}}^2 = \sigma_x^2/M \quad (41)$$

$$\bar{y} \text{ calculated from } N \text{ events, i.e., } \sigma_{\bar{y}}^2 = \sigma_y^2/N \quad (42)$$

and unbiased estimators of the variances are

$$s_{\bar{x}}^2 = s_x^2/M = \frac{\sum_{i=1}^M (x_i - \bar{x})^2}{M(M-1)} \quad \text{and} \quad s_{\bar{y}}^2 = s_y^2/N = \frac{\sum_{j=1}^N (y_j - \bar{y})^2}{N(N-1)}. \quad (43)$$

Under the **additional assumption** $\sigma_x^2 = \sigma_y^2$ the probability

$$P(|\bar{x} - \bar{y}| > d) \quad (44)$$

is determined by the Student distribution function in the same way as the probability of the Gaussian difference test is determined by the normal distribution.

Examples for the Student difference test for $\bar{x}_1 = 1.00 \pm 0.05$ from M data and $\bar{x}_2 = 1.20 \pm 0.05$ from N data are given in table 2. The Gaussian difference test gives $Q = 0.0047$. For $M = N = 512$ the Student Q value is practically identical with the Gaussian result, for $M = N = 16$ it has almost doubled. Likelihoods above a 5% cut-off, are only obtained for $M = N = 2$ (11%) and $M = 16, N = 4$ (7%). The latter result looks a bit surprising, because its Q value is smaller than for $M = N = 4$. The explanation is that for $M = 16, N = 4$ data one would expect the $N = 4$ error bar to be two times larger than the $M = 16$ error bar, whereas the estimated error bars are identical. This leads to the problem: Data are assumed to be sampled from the same normal distribution, when are two measured error bars consistent and when not?

Table 2. Student difference test for the data $\bar{x}_1 = 1.00 \pm 0.05$ and $\bar{x}_2 = 1.20 \pm 0.05$ (compile and run the program provided in `ForProc/Stud_dif`, which results in an interactive dialogue).

M	512	32	16	16	4	3	2
N	512	32	16	4	4	3	2
Q	0.0048	0.0063	0.0083	0.072	0.030	0.047	0.11

6.1. χ^2 Distribution, Error of the Error Bar, F-Test

The distribution of the random variable

$$(\chi^r)^2 = \sum_{i=1}^N (y_i^r)^2, \tag{45}$$

where each y_i^r is normally distributed, defines the **χ^2 distribution** with N degrees of freedom. The study of the variance $(s_x^r)^2$ of a Gaussian sample can be reduced to the χ^2 -distribution with $f = N - 1$ degrees of freedom

$$(\chi_f^r)^2 = \frac{(N - 1)(s_x^r)^2}{\sigma_x^2} = \sum_{i=1}^N \frac{(x_i^r - \bar{x}^r)^2}{\sigma_x^2}. \tag{46}$$

The probability density of χ^2 **per degree of freedom (pdf)** is

$$f_N(\chi^2) = Nf(N\chi^2) = \frac{a e^{-a\chi^2} (a\chi^2)^{a-1}}{\Gamma(a)} \quad \text{where } a = \frac{N}{2}. \tag{47}$$

The Error of the Error Bar: For normally distributed data the number of data alone determines the errors of error bars, because the χ^2 distribution is exactly known. Confidence intervals for variance estimates $s_x^2 = 1$ from NDAT data (assignment `a0204_01`) are:

		q	q	q	1-q	1-q
NDAT=2**K		.025	.150	.500	.850	.975
2	1	.199	.483	2.198	27.960	1018.255
4	2	.321	.564	1.268	3.760	13.902
8	3	.437	.651	1.103	2.084	4.142
16	4	.546	.728	1.046	1.579	2.395
32	5	.643	.792	1.022	1.349	1.768
1024	10	.919	.956	1.001	1.048	1.093
16384	14	.979	.989	1.000	1.012	1.022

The variance ratio test or F-test: We assume that two sets of normal data are given together with estimates of their variances: $(s_{x_1}^2, N_1)$ and $(s_{x_2}^2, N_2)$. We would like to test whether the ratio $F = s_{x_1}^2/s_{x_2}^2$ differs from $F = 1$ in a statistically significant way. The probability $(f_1/f_2) F < w$, where $f_i = N_i - 1$, $i = 1, 2$, is known to be

$$H(w) = 1 - B_I\left(\frac{1}{w+1}, \frac{1}{2}f_2, \frac{1}{2}f_1\right). \tag{48}$$

Examples are given in table 3. This allows us later to compare the efficiency of MC algorithms.

Table 3. Examples for the F-test (use the program in `ForProc/F_test` or the one in `ForProc/F_stud`).

$\Delta\bar{x}_1$	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
N_1	16	16	64	1024	2048	32	1024	16
$\Delta\bar{x}_2$	1.0	1.0	1.0	1.05	1.05	2.0	2.0	2.0
N_2	16	8	16	1024	2048	8	256	16
Q	1.0	0.36	0.005	0.12	0.027	0.90	0.98	0.01

6.2. The Jackknife Approach

Jackknife estimators allow to correct for the bias and the error of the bias. The method was introduced in the 1950s (for a review see [7]). It is **recommended as the standard** for error bar calculations. In unbiased situations the jackknife and the usual error bars agree. Otherwise the jackknife estimates are more reliable.

The unbiased estimator of the expectation value \hat{x} is

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

Bias problems may occur when one estimates a non-linear function of \hat{x} :

$$\hat{f} = f(\hat{x}) . \quad (49)$$

Typically, the bias is of order $1/N$:

$$\text{bias}(\bar{f}) = \hat{f} - \langle \bar{f} \rangle = \frac{a_1}{N} + \frac{a_2}{N^2} + O\left(\frac{1}{N^3}\right) \quad (50)$$

where a_1 and a_2 are constants. But for the biased estimator we lost the ability to estimate the variance $\sigma^2(\bar{f}) = \sigma^2(f)/N$ via the standard equation

$$s^2(\bar{f}) = \frac{1}{N} s^2(f) = \frac{1}{N(N-1)} \sum_{i=1}^N (f_i - \bar{f})^2 , \quad (51)$$

because $f_i = f(x_i)$ is not a valid estimator of \hat{f} . Further, it is in non-trivial applications almost always a bad idea to use linear error propagation formulas. Jackknife methods are not only easier to implement, but also more precise and far more **robust**.

The error bar problem for the estimator \bar{f} is conveniently overcome by using **jackknife estimators** \bar{f}^J , f_i^J , defined by

$$\bar{f}^J = \frac{1}{N} \sum_{i=1}^N f_i^J \quad \text{with} \quad f_i^J = f(x_i^J) \quad \text{and} \quad x_i^J = \frac{1}{N-1} \sum_{k \neq i} x_k . \quad (52)$$

The estimator for the variance $\sigma^2(\bar{f}^J)$ is

$$s_J^2(\bar{f}^J) = \frac{N-1}{N} \sum_{i=1}^N (f_i^J - \bar{f}^J)^2 . \quad (53)$$

Straightforward algebra shows that in the unbiased case the estimator of the jackknife variance (53) reduces to the normal variance (51). Notably only of order N (not N^2) operations are needed to construct the jackknife averages x_i^J , $i = 1, \dots, N$ from the original data.

7. Statistical Physics and Potts Models

MC simulations of systems described by the Gibbs canonical ensemble aim at calculating estimators of physical observables at a temperature T . In the following we choose units so that the Boltzmann constant becomes one, i.e. $\beta = 1/T$. Let us consider the calculation of the **expectation value** of an **observable** \mathcal{O} . Mathematically all systems on a computer are discrete,

because a finite word length has to be used. Hence, the expectation value is given by the sum

$$\widehat{\mathcal{O}} = \widehat{\mathcal{O}}(\beta) = \langle \mathcal{O} \rangle = Z^{-1} \sum_{k=1}^K \mathcal{O}^{(k)} e^{-\beta E^{(k)}} \quad (54)$$

$$\text{where } Z = Z(\beta) = \sum_{k=1}^K e^{-\beta E^{(k)}} \quad (55)$$

is the **partition function**. The index $k = 1, \dots, K$ labels the **configurations** of the system, and $E^{(k)}$ is the (internal) energy of configuration k . The configurations are also called **microstates**. To distinguish the configuration index from other indices, it is put in parenthesis.

We introduce generalized Potts models in an external magnetic field on d -dimensional hypercubic lattices with periodic boundary conditions (i.e., the models are defined on a torus in d dimensions). Without being overly complicated, these models are general enough to illustrate the essential features we are interested in. In addition, various subcases of these models are by themselves of physical interest.

We define the energy function of the system by

$$-\beta E^{(k)} = -\beta E_0^{(k)} + H M^{(k)} \quad (56)$$

where

$$E_0^{(k)} = -2 \sum_{\langle ij \rangle} \delta(q_i^{(k)}, q_j^{(k)}) + \frac{2dN}{q} \quad (57)$$

$$\text{with } \delta(q_i, q_j) = \begin{cases} 1 & \text{for } q_i = q_j \\ 0 & \text{for } q_i \neq q_j \end{cases} \quad \text{and } M^{(k)} = 2 \sum_{i=1}^N \delta(1, q_i^{(k)}).$$

The sum $\langle ij \rangle$ is over the nearest neighbor lattice sites and $q_i^{(k)}$ is called the **Potts spin** or **Potts state** of configuration k at site i . For the q -state Potts model $q_i^{(k)}$ takes on the values $1, \dots, q$. The external magnetic field is chosen to interact with the state $q_i = 1$ at each site i , but not with the other states $q_i \neq 1$. The case $q = 2$ becomes equivalent to the Ising ferromagnet. See F.Y. Wu [25] for a detailed review of Potts models.

For the **energy per spin** our notation is

$$e_s = E/N. \quad (58)$$

A factor of two and an additive constant are introduced in Eq. (57), so that e_s agrees for $q = 2$ with the conventional Ising model definition, and

$$\beta = \beta^{\text{Ising}} = \frac{1}{2} \beta^{\text{Potts}}. \quad (59)$$

For the $2d$ Potts models a number of exact results are known in the infinite volume limit, mainly due to work by Baxter [1]. The phase transition temperatures are

$$\frac{1}{2} \beta_c^{\text{Potts}} = \beta_c = \frac{1}{T_c} = \frac{1}{2} \ln(1 + \sqrt{q}), \quad q = 2, 3, \dots \quad (60)$$

At β_c the average energy per state is

$$e_s^c = E_0^c/N = \frac{4}{q} - 2 - 2/\sqrt{q}. \quad (61)$$

The phase transition is second order for $q \leq 4$ and first order for $q \geq 5$. The exact infinite volume **latent heats** Δe_s and **entropy jumps** Δs were also found by Baxter [1], while the interface tensions f_s were derived later (see [9] and references therein).

8. Sampling and Re-weighting

For the Ising model it is straightforward to **sample statistically independent configurations**. We simply have to generate N spins, each either up or down with 50% likelihood. This is called **random sampling**. In Fig. 4 a thus obtained histogram for the $2d$ Ising model **energy per spin** is depicted.

Note that it is very important to distinguish the energy measurements on single configurations from the expectation value. The expectation value \hat{e}_s is a single number, while e_s fluctuates. From the measurement of many e_s values one finds an estimator of the mean, \bar{e}_s , which fluctuates too.

The histogram entries at $\beta = 0$ can be re-weighted so that they correspond to other β values. We simply have to multiply the entry corresponding to energy E by $\exp(-\beta E)$. Similarly histograms corresponding to the Gibbs ensemble at some value β_0 can be re-weighted to other β values. Care has to be taken to ensure that the arguments of the exponential function do not become too large. This can be done by first calculating the mean energy and then implementing re-weighting with respect to the difference from the mean.

Re-weighting has a long history. For finite size scaling (FSS) investigations of second order phase transitions its usefulness has been stressed

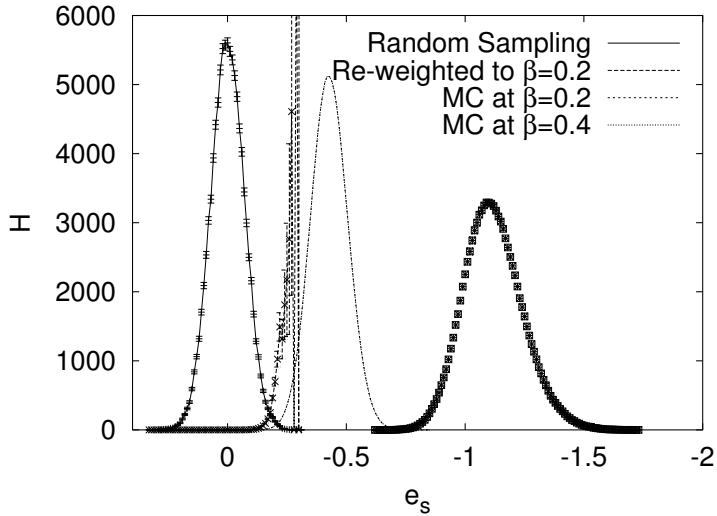


Fig. 4. Energy histograms of 100 000 entries each for the Ising model on a 20×20 lattice: Random Sampling gives statistically independent configurations at $\beta = 0$. Histograms at $\beta = 0.2$ and $\beta = 0.4$ are generated with Markov chain MC. Re-weighting of the $\beta = 0$ random configurations to $\beta = 0.2$ is shown to fail (assignments a0301_02 and a0303_02).

by Ferrenberg and Swendsen [12] (accurate determinations of peaks of the specific heat or of susceptibilities).

In Fig. 4 re-weighting is done from $\beta_0 = 0$ to $\beta = 0.2$. But, by comparison to the histogram from a Metropolis MC calculation at $\beta = 0.2$, the result is seen to be disastrous. The reason is easily identified: In the range where the $\beta = 0.2$ histogram takes on its maximum, the $\beta = 0$ histogram has not a single entry. Our random sampling procedure misses the important configurations at $\beta = 0.2$. Re-weighting to new β values works only in a range $\beta_0 \pm \Delta\beta$, where $\Delta\beta \rightarrow 0$ in the infinite volume limit.

Important Configurations: Let us determine the important contributions to the partition function. The partition function can be re-written as a sum over energies

$$Z = Z(\beta) = \sum_E n(E) e^{-\beta E} \quad (62)$$

where the unnormalized spectral density $n(E)$ is defined as the number of microstates k with energy E . For a fixed value of β the energy probability density

$$P_\beta(E) = c_\beta n(E) e^{-\beta E} \quad (63)$$

is peaked around the average value $\widehat{E}(\beta)$, where c_β is a normalization constant determined by $\sum_E P_\beta(E) = 1$.

Away from first and second order phase transitions, the width of the energy distribution is $\Delta E \sim \sqrt{V}$. This follows from the fact that the fluctuations of the $N \sim V$ lattice spins are essentially uncorrelated, so that the magnitude of a typical fluctuations is $\sim \sqrt{N}$. As the energy is an extensive quantity $\sim V$, we find that the re-weighting range is $\Delta\beta \sim 1/\sqrt{V}$, so that $\Delta\beta E \sim \sqrt{V}$ stays within the fluctuation of the system.

Interestingly, the re-weighting range increases at a second order phase transition point, because critical fluctuations are larger than non-critical fluctuations. Namely, one has $\Delta E \sim V^x$ with $1/2 < x < 1$ and the requirement $\Delta\beta E \sim V^x$ yields $\Delta\beta \sim V^{x-1}$.

For first order phase transitions one has a latent heat $\Delta V \sim V$, but this does not mean that the re-weighting range becomes of order one. In essence, the fluctuations collapse, because the two phases become separated by an interface. One is back to fluctuations within either of the two phases, *i.e.* $\Delta\beta \sim 1/\sqrt{V}$.

The important configurations at temperature $T = 1/\beta$ are at the energy values for which the probability density $P_\beta(E)$ is large. To sample them efficiently, one needs a procedure which generates the configurations with their Boltzmann weights

$$w_B^{(k)} = e^{-\beta E^{(k)}} \quad (64)$$

The number of configurations $n(E)$ and the weights combine then so that the probability to generate a configuration at energy E becomes precisely $P_\beta(E)$ as given by equation (63).

9. Importance Sampling and Markov Chain Monte Carlo

For the canonical ensemble **importance sampling** generates configurations k with probability

$$P_B^{(k)} = c_B w_B^{(k)} = c_B e^{-\beta E^{(k)}} \quad (65)$$

where the constant c_B is determined by the normalization condition $\sum_k P_B^{(k)} = 1$. The vector $(P_B^{(k)})$ is called the **Boltzmann state**. When configurations are stochastically generated with probability $P_B^{(k)}$, the **expectation value** becomes the **arithmetic average**:

$$\widehat{\mathcal{O}} = \widehat{\mathcal{O}}(\beta) = \langle \mathcal{O} \rangle = \lim_{N_K \rightarrow \infty} \frac{1}{N_K} \sum_{n=1}^{N_K} \mathcal{O}^{(k_n)} \quad (66)$$

Truncating the sum at some finite value of N_K , we obtain an **estimator of the expectation value**

$$\bar{\mathcal{O}} = \frac{1}{N_K} \sum_{n=1}^{N_K} \mathcal{O}^{(k_n)} . \quad (67)$$

Normally, we cannot generate configurations k directly with the probability (65), but they may be found as members of the equilibrium distribution of a dynamic process. A **Markov process** is a particularly simple dynamic process, which generates configuration k_{n+1} stochastically from configuration k_n , so that no information about previous configurations k_{n-1}, k_{n-2}, \dots is needed. The elements of the Markov process **time series** are the configurations. Assume that the configuration k is given. Let the transition probability to create the configuration l in one step from k be given by $W^{(l)(k)} = W[k \rightarrow l]$. The **transition matrix**

$$W = \left(W^{(l)(k)} \right) \quad (68)$$

defines the Markov process. Note, that this matrix is very big (never stored in the computer), because its labels are the configurations. To generate configurations with the desired probabilities, the matrix W needs to satisfy the following properties:

(i) **Ergodicity:**

$$e^{-\beta E^{(k)}} > 0 \text{ and } e^{-\beta E^{(l)}} > 0 \text{ imply :} \quad (69)$$

an integer number $n > 0$ exists so that $(W^n)^{(l)(k)} > 0$ holds.

(ii) **Normalization:**

$$\sum_l W^{(l)(k)} = 1 . \quad (70)$$

(iii) **Balance:**

$$\sum_k W^{(l)(k)} e^{-\beta E^{(k)}} = e^{-\beta E^{(l)}} . \quad (71)$$

Balance means: The Boltzmann state (65) is an eigenvector with eigenvalue 1 of the matrix $W = (W^{(l)(k)})$.

An **ensemble** is a collection of configurations for which to each configuration k a probability $P^{(k)}$ is assigned, $\sum_k P^{(k)} = 1$. The **Gibbs or Boltzmann ensemble** E_B is defined to be the ensemble with the probability distribution (65).

An **equilibrium ensemble** E_{eq} of the Markov process is defined by its probability distribution P_{eq} satisfying

$$W P_{eq} = P_{eq}, \text{ in components } P_{eq}^{(l)} = \sum_k W^{(l)(k)} P_{eq}^{(k)}. \quad (72)$$

Statement: Under the conditions (i), (ii) and (iii) the Boltzmann ensemble is the **only** equilibrium ensemble of the Markov process.

For a proof the readers is referred to [7]. There are many ways to construct a Markov process satisfying (i), (ii) and (iii). A stronger condition than balance (71) is

(iii') **Detailed balance:**

$$W^{(l)(k)} e^{-\beta E^{(k)}} = W^{(k)(l)} e^{-\beta E^{(l)}}. \quad (73)$$

Using the normalization $\sum_k W^{(k)(l)} = 1$ detailed balance implies balance (iii).

At this point we have succeeded to replace the canonical ensemble average by a time average over an artificial dynamics. Calculating averages over large times, like one does in real experiments, is equivalent to calculating averages of the ensemble. One distinguishes *dynamical universality classes*. The Metropolis and heat bath algorithms discussed in the following fall into the class of so called *Glauber dynamics*, model A in a frequently used classification [10]. Cluster algorithms [21] constitute another universality class.

9.1. Metropolis and Heat Bath Algorithm for Potts Models

The **Metropolis algorithm** can be used whenever one knows how to calculate the energy of a configuration. Given a configuration k , the Metropolis algorithm proposes a configuration l with probability

$$f(l, k) \text{ normalized to } \sum_l f(l, k) = 1. \quad (74)$$

The new configuration l is accepted with probability

$$w^{(l)(k)} = \min \left[1, \frac{P_B^{(l)}}{P_B^{(k)}} \right] = \begin{cases} 1 & \text{for } E^{(l)} < E^{(k)} \\ e^{-\beta(E^{(l)} - E^{(k)})} & \text{for } E^{(l)} > E^{(k)}. \end{cases} \quad (75)$$

If the new configuration is rejected, the old configuration has to be counted again. The **acceptance rate** is defined as the ratio of accepted changes

over proposed moves. With this convention we do not count a move as accepted when it proposes the at hand configuration.

The Metropolis procedure gives rise to the transition probabilities

$$W^{(l)(k)} = f(l, k) w^{(l)(k)} \quad \text{for } l \neq k \quad (76)$$

$$\text{and } W^{(k)(k)} = f(k, k) + \sum_{l \neq k} f(l, k) (1 - w^{(l)(k)}) . \quad (77)$$

Therefore, the ratio $(W^{(l)(k)}/W^{(k)(l)})$ satisfies detailed balance (73) if

$$f(l, k) = f(k, l) \quad \text{holds.} \quad (78)$$

Otherwise the probability density $f(l, k)$ is unconstrained. So there is an amazing flexibility in the choice of the transition probabilities $W^{(l)(k)}$. Also, the algorithm generalizes immediately to arbitrary weights.

The **heat bath algorithm** chooses a state q_i directly with the local Boltzmann distribution defined by its nearest neighbors. The state q_i can take on one of the values $1, \dots, q$ and, with all other states set, determines a value of the energy function (56). We denote this energy by $E(q_i)$ and the Boltzmann probabilities are

$$P_B(q_i) = \text{const } e^{-\beta E(q_i)} \quad (79)$$

where the constant follows from the normalization condition

$$\sum_{q_i=1}^q P_B(q_i) = 1 . \quad (80)$$

In equation (79) we can define $E(q_i)$ to be just the contribution of the interaction of q_i with its nearest neighbors to the total energy and absorb the other contributions into the overall constant. Here we give a generic code which works for arbitrary values of q and d (other implementations may be more efficient).

We calculate the cumulative distribution function of the heat bath probabilities

$$P_{HB}(q_i) = \sum_{q'_i=1}^{q_i} P_B(q'_i) . \quad (81)$$

The normalization condition (80) implies $P_{HB}(q) = 1$. Comparison of these cumulative probabilities with a uniform random number x^r yields the heat bath update $q_i \rightarrow q'_i$. Note that in the heat bath procedure the original value q_i^{in} does not influence the selection of q_i^{new} .

9.2. The $O(3)$ σ Model and the Heat Bath Algorithm

We give an example of a model with a continuous energy function. Expectation values are calculated with respect to the partition function

$$Z = \int \prod_i ds_i e^{-\beta E(\{s_i\})} . \tag{82}$$

The spins $\vec{s}_i = \begin{pmatrix} s_{i,1} \\ s_{i,2} \\ s_{i,3} \end{pmatrix}$ are normalized to $(\vec{s}_i)^2 = 1$ (83)

and the measure ds_i is defined by $\int ds_i = \frac{1}{4\pi} \int_{-1}^{+1} d \cos(\theta_i) \int_0^{2\pi} d\phi_i$, (84)

where the polar (θ_i) and azimuth (ϕ_i) angles define the spin s_i on the unit sphere. The energy is

$$E = - \sum_{\langle ij \rangle} \vec{s}_i \vec{s}_j , \tag{85}$$

where the sum goes over the nearest neighbor sites of the lattice and $\vec{s}_i \vec{s}_j$ is the dot product of the vectors. The $2d$ version of the model is of interest to field theorists because of its analogies with the four-dimensional Yang-Mills theory. In statistical physics the d -dimensional model is known as the **Heisenberg ferromagnet** (references can be found in [7]).

We would like to update a single spin \vec{s} . The sum of its $2d$ neighbors is

$$\vec{S} = \vec{s}_1 + \vec{s}_2 + \dots + \vec{s}_{2d-1} + \vec{s}_{2d} .$$

Hence, the contribution of spin \vec{s} to the energy is $2d - \vec{s}\vec{S}$. We propose a new spin \vec{s}' with the measure (84) by drawing two uniformly distributed random numbers

$$\begin{aligned} \phi^r &\in [0, 2\pi) \quad \text{for the azimuth angle and} \\ \cos(\theta^r) &= x^r \in [-1, +1) \quad \text{for the cosine of the polar angle.} \end{aligned}$$

This defines the probability function $f(\vec{s}', \vec{s})$ of the Metropolis process, which accepts the proposed spin \vec{s}' with probability

$$w(\vec{s} \rightarrow \vec{s}') = \begin{cases} 1 & \text{for } \vec{S}\vec{s}' > \vec{S}\vec{s}, \\ e^{-\beta(\vec{S}\vec{s} - \vec{S}\vec{s}')} & \text{for } \vec{S}\vec{s}' < \vec{S}\vec{s}. \end{cases}$$

If sites are chosen with the uniform probability distribution $1/N$ per site, where N is the total number of spins, it is obvious that the algorithm

fulfills detailed balance. It is noteworthy that the procedure remains valid when the spins are chosen in the systematic order $1, \dots, N$. Balance (71) still holds, whereas detailed balance (73) is violated (an exercise of Ref. [7]).

One would prefer to choose \vec{s}' directly with the probability

$$W(\vec{s} \rightarrow \vec{s}') = P(\vec{s}' ; \vec{S}) = \text{const } e^{\beta \vec{s}' \cdot \vec{S}} .$$

The **heat bath algorithm** creates this distribution. Implementation of it becomes feasible when the energy function allows for an explicit calculation of the probability $P(\vec{s}' ; \vec{S})$. This is an easy task for the $O(3)$ σ -model. Let

$$\alpha = \text{angle}(\vec{s}' , \vec{S}), \quad x = \cos(\alpha) \quad \text{and} \quad S = \beta |\vec{S}| .$$

For $S = 0$ a new spin \vec{s}' is simply obtained by random sampling. We assume in the following $S > 0$. The Boltzmann weight becomes $\exp(xS)$ and the normalization constant follows from

$$\int_{-1}^{+1} dx e^{xS} = \frac{2}{S} \sinh(S) .$$

Therefore, the desired probability is

$$P(\vec{s}' ; \vec{S}) = \frac{S}{2 \sinh(S)} e^{xS} =: f(x)$$

and the method of Eq. (6) can be used to generate events with the probability density $f(x)$. A uniformly distributed random number $y^r \in [0, 1]$ translates into

$$x^r = \cos \alpha^r = \frac{1}{S} \ln [\exp(+S) - y^r \exp(+S) + y^r \exp(-S)] . \quad (86)$$

Finally, one has to give \vec{s}' a direction in the plane orthogonal to S . This is done by choosing a random angle β^r uniformly distributed in the range $0 \leq \beta^r < 2\pi$. Then, $x^r = \cos \alpha^r$ and β^r completely determine \vec{s}' with respect to \vec{S} . Before storing \vec{s}' in the computer memory, we have to calculate coordinates of \vec{s}' with respect to a Cartesian coordinate system, which is globally used for all spins of the lattice. This amounts to a linear transformation.

9.3. Example Runs

Start and equilibration: Under repeated application of one of our updating procedures the probability of states will approach the Boltzmann distribution. However, initially we have to start with a microstate which may be far off the Boltzmann distribution. Suppression factors like 10^{-10000}

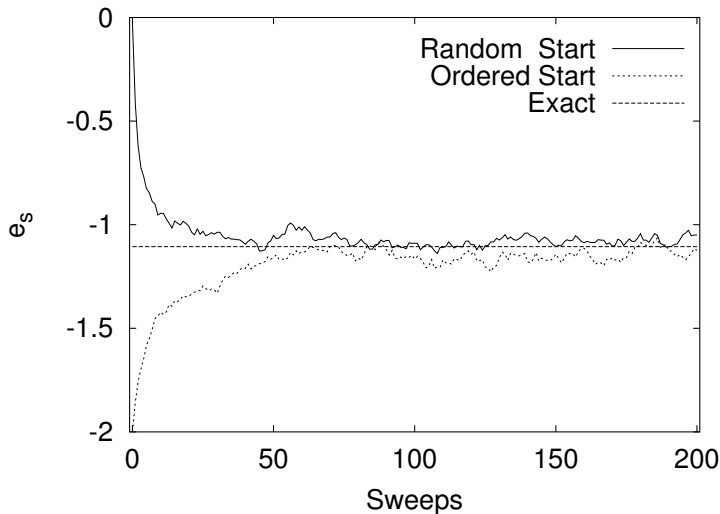


Fig. 5. Two Metropolis time series of 200 sweeps each for a $2d$ Ising model on a 80×80 lattice at $\beta = 0.4$ are shown. Random updating for which the positions of the spins are chosen with the uniform probability distribution was used. Measurements of the energy per spin after every sweep are plotted for ordered and disordered starts. The exact mean value $\hat{e}_s = -1.10608$ is also indicated (assignment a0303_01).

are possible. Although the weight of states decreases with $1/n$ where n is the number of steps of the Markov process, one should exclude the initial states from the equilibrium statistics. In practice this means we should allow for a certain number of sweeps n_{equi} to equilibrate the system. One **sweep** updates each spin once or once in the average.

Many ways to generate start configurations exist. Two natural and easy to implement choices are:

- (1) Generate a random configuration corresponding to $\beta = 0$. This defines a **random** or **disordered start** of a MC simulation.
- (2) Generate a configuration for which all Potts spins take on the same q -value. This is called an **ordered start** of a MC simulation.

Examples of initial time series are given in Fig. 5 and 6. Unless explicitly stated otherwise, we use here and in the following always **sequential updating**, for which the spins are touched in a systematic order.

Consistency Checks: For the $2d$ Ising model we can test against the exact finite lattice results of Ferdinand and Fisher [11]. We simulate a 20^2

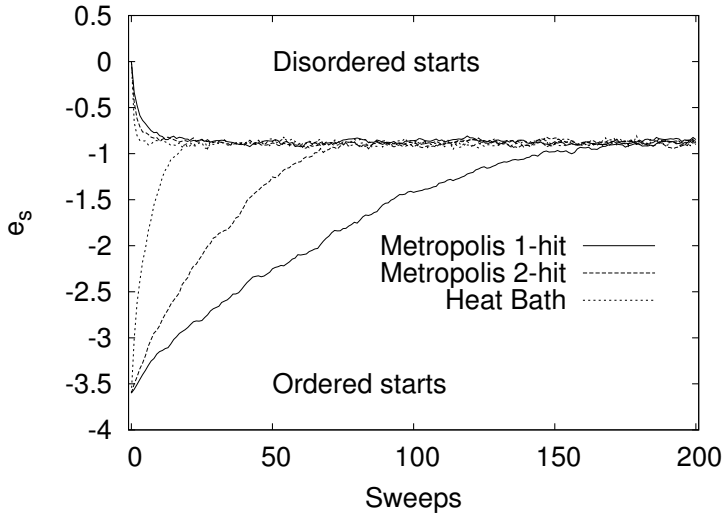


Fig. 6. $q = 10$ Potts model time series of 200 sweeps on a 80×80 lattice at $\beta = 0.62$. Measurements of the energy per spin after every sweep are plotted for ordered and disordered starts (assignment a0303_05).

lattice at $\beta = 0.4$, using a statistics of 10 000 sweeps for reaching equilibrium. The statistics for measurement is chosen to be 64 bins of 5 000 sweeps each. The number 64 is taken, because according to the student distribution the approximation to the Gaussian distribution is then excellent, and the binsize of 5 000 ($\gg 200$) is argued to be large enough to neglect correlations between the bins. A more careful analysis is the subject of our next section. With our statistics we find (assignment a0303_06)

$$\bar{e}_s = -1.1172 \text{ (14) (Metropolis) versus } \hat{e}_s = -1.117834 \text{ (exact) . (87)}$$

The Gaussian difference test gives a perfectly admissible value, $Q = 0.66$.

For the $2d$ 10-state Potts model at $\beta = 0.62$ we test our Metropolis versus our heat bath code on a 20×20 lattice. For the heat bath updating we use the same statistics as for the $2d$ Ising model. For the Metropolis updating we increase these numbers by a factor of four. This increase is done, because we expect the performance of Metropolis updating for the 10-state model to be worse than for the 2-state model: At low temperature the likelihood to propose the most probable (aligned) Potts spin is $1/2$ for the 2-state model, but only $1/10$ for the 10-state model, and $\beta = 0.62$ is sufficiently close to the ordered phase, so that this effect is expected to be of relevance. The results of our simulations are (assignment a0303_08)

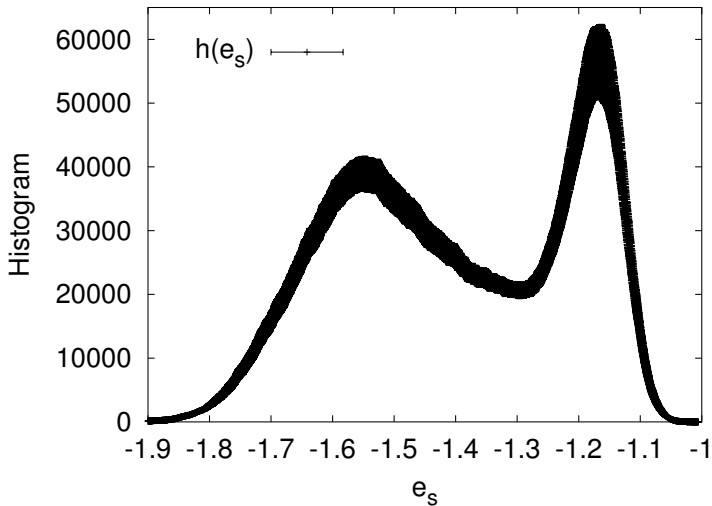


Fig. 7. Histogram of the energy per spin for the $3d$ 3-state Potts model on a 24^3 latticed at $\beta = 0.275229525$ (assignment a0303_10).

$e_s = -0.88709$ (30) (Metropolis) versus $e_s = -0.88664$ (28) (heat bath) and $Q = 0.27$ for the Gaussian difference test. Another perfectly admissible value.

To illustrate features of a first order phase transition for the $3d$ 3-state Potts model, we use the 1-hit Metropolis algorithm on a 24^3 lattice and simulate at $\beta = 0.275229525$. We perform 20 000 sweeps for reaching equilibrium, then $64 \times 10\,000$ sweeps with measurements. From the latter statistics we show in Fig. 7 the energy histogram and its error bars. The histogram exhibits a **double peak** structure, which is typically obtained when systems with first order transitions are simulated on finite lattices in the neighborhood of so called **pseudo-transition temperatures**. These are finite lattice temperature definitions, which converge with increasing system size towards the infinite volume transition temperature. Equal heights of the maxima of the two peaks is one of the popular definitions of a pseudo-transition temperature for first order phase transitions. Equal weights (areas under the curves) is another, used in the lecture by Prof. Landau. Our β value needs to be re-weighted to a slightly higher value to arrange for equal heights (assignment a0303_10). Our mean energy per spin, corresponding to the histogram of the figure is $e_s = -1.397$ (13). Due to the double peak structure of the histogram the error bar is relatively large. Still, the cen-

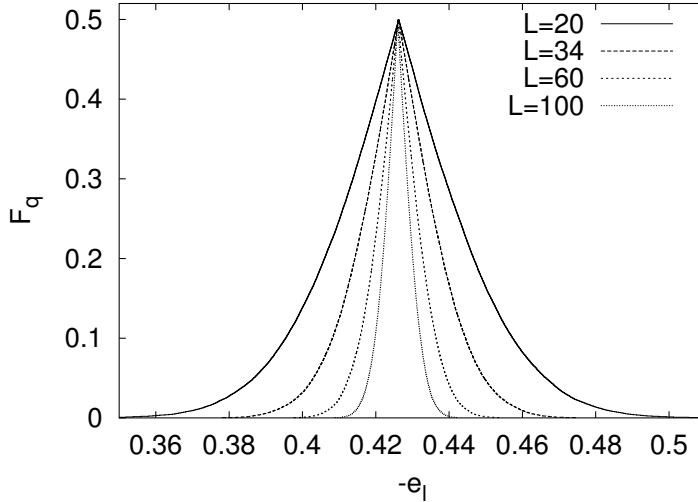


Fig. 8. Peaked distribution functions for the $O(3)$ σ -model mean energy per link on various lattices at $\beta = 1.1$ (assignment a0304_08).

tral limit theorem works and a Kolmogorov test shows that our statistics is large enough to create an approximately Gaussian distribution for the binned data (assignment a0303_11).

Self-Averaging Illustration for the $O(3)$ model: We compare in Fig. 8 the peaked distribution function of the mean energy per link e_l for different lattice sizes. The property of **self-averaging** is observed: The larger the lattice, the smaller the confidence range. The other way round, the peaked distribution function is very well suited to exhibit observables for which self-averaging does not work, as for instance encountered in spin glass simulations [5].

10. Statistical Errors of Markov Chain Monte Carlo Data

In large scale MC simulation it may take months, possibly years, to collect the necessary statistics. For such data a thorough error analysis is a must. A typical MC simulation falls into two parts:

- (1) **Equilibration:** Initial sweeps are performed to reach the equilibrium distribution. During these sweeps measurements are either not taken at all or they have to be discarded when calculating equilibrium expectation values.

(2) **Data Production:** Sweeps with measurements are performed. Equilibrium expectation values are calculated from this statistics.

A rule of thumb is: **Do not spend more than 50% of your CPU time on measurements!** The reason for this rule is that one cannot be off by a factor worse than two ($\sqrt{2}$ in the statistical error).

How many sweeps should be discarded for reaching equilibrium? In a few situations this question can be rigorously answered with the *Coupling from the Past* method (see the article by W. Kendall in this volume). The next best thing to do is to measure the integrated autocorrelation time and to discard, after reaching a visually satisfactory situation, a number of sweeps which is larger than the integrated autocorrelation time. In practice even this can often not be achieved.

Therefore, it is re-assuring that it is sufficient to pick the number of discarded sweeps approximately right. With increasing statistics the contribution of the non-equilibrium data dies out like $1/N$, where N is the number of measurements. This is eventually swallowed by the statistical error, which declines only like $1/\sqrt{N}$. The point of discarding the equilibrium configurations is that the factor in front of $1/N$ can be large.

There can be far more involved situations, like that the Markov chain ends up in a metastable configuration, which may even stay unnoticed (this tends to happen in complex systems like spin glasses or proteins).

10.1. Autocorrelations

We like to estimate the expectation value \hat{f} of some physical observable. We assume that the system has reached equilibrium. How many MC sweeps are needed to estimate \hat{f} with some desired accuracy? To answer this question, one has to understand the autocorrelations within the Markov chain.

Given is a **time series** of N measurements from a Markov process

$$f_i = f(x_i), \quad i = 1, \dots, N, \quad (88)$$

where x_i are the configurations generated. The label $i = 1, \dots, N$ runs in the temporal order of the Markov chain and the elapsed time (measured in updates or sweeps) between subsequent measurements f_i, f_{i+1} is always the same. The estimator of the expectation value \hat{f} is

$$\bar{f} = \frac{1}{N} \sum f_i. \quad (89)$$

With the notation

$$t = |i - j|$$

the definition of the **autocorrelation function** of the observable \hat{f} is

$$\hat{C}(t) = \hat{C}_{ij} = \langle (f_i - \langle f_i \rangle) (f_j - \langle f_j \rangle) \rangle = \langle f_i f_j \rangle - \langle f_i \rangle \langle f_j \rangle = \langle f_0 f_t \rangle - \hat{f}^2 \quad (90)$$

where we used that translation invariance in time holds for the equilibrium ensemble. The asymptotic behavior for large t is

$$\hat{C}(t) \sim \exp\left(-\frac{t}{\tau_{\text{exp}}}\right) \quad \text{for } t \rightarrow \infty, \quad (91)$$

where τ_{exp} is called **(exponential) autocorrelation time** and is related to the second largest eigenvalue λ_1 of the transition matrix by $\tau_{\text{exp}} = -1/\ln \lambda_1$ under the assumption that f has a non-zero projection on the corresponding eigenstate. Superselection rules are possible so that different autocorrelation times reign for different operators.

The variance of f is a special case of the autocorrelations (90)

$$\hat{C}(0) = \sigma^2(f). \quad (92)$$

Some algebra [7] shows that the variance of the estimator \bar{f} (89) for the mean and the autocorrelation functions (90) are related by

$$\sigma^2(\bar{f}) = \frac{\sigma^2(f)}{N} \left[1 + 2 \sum_{t=1}^{N-1} \left(1 - \frac{t}{N} \right) \hat{c}(t) \right] \quad \text{with } \hat{c}(t) = \frac{\hat{C}(t)}{\hat{C}(0)}. \quad (93)$$

This equation ought to be compared with the corresponding equation for uncorrelated random variables $\sigma^2(\bar{f}) = \sigma^2(f)/N$. The difference is the factor in the bracket of (93), which defines the **integrated autocorrelation time**

$$\tau_{\text{int}} = \left[1 + 2 \sum_{t=1}^{N-1} \left(1 - \frac{t}{N} \right) \hat{c}(t) \right]. \quad (94)$$

For correlated data the variance of the mean is larger by a factor τ_{int} than the corresponding **naive variance** for uncorrelated data:

$$\tau_{\text{int}} = \frac{\sigma^2(\bar{f})}{\sigma_{\text{naive}}^2(\bar{f})} \quad \text{with } \sigma_{\text{naive}}^2 = \frac{\sigma^2(f)}{N}. \quad (95)$$

In most simulations one is interested in the limit $N \rightarrow \infty$ and equation (94) becomes

$$\tau_{\text{int}} = 1 + 2 \sum_{t=1}^{\infty} \hat{c}(t). \quad (96)$$

The numerical estimation of the integrated autocorrelation time faces difficulties. Namely, the variance of the $N \rightarrow \infty$ estimator of τ_{int} diverges:

$$\bar{\tau}_{\text{int}} = 1 + 2 \sum_{t=1}^{\infty} \bar{c}(t) \quad \text{and} \quad \sigma^2(\bar{\tau}_{\text{int}}) \rightarrow \infty, \quad (97)$$

because for large t each $\bar{c}(t)$ adds a constant amount of noise, whereas the signal dies out like $\exp(-t/\tau_{\text{exp}})$. To obtain an estimate one considers the t -dependent estimator

$$\bar{\tau}_{\text{int}}(t) = 1 + 2 \sum_{t'=1}^t \bar{c}(t') \quad (98)$$

and looks out for a **window** in t for which $\bar{\tau}_{\text{int}}(t)$ is flat.

To give a simple example, let us assume that the autocorrelation function is governed by a single exponential autocorrelation time

$$\hat{C}(t) = \text{const} \exp\left(-\frac{t}{\tau_{\text{exp}}}\right). \quad (99)$$

In this case we can carry out the sum (96) for the integrated autocorrelation function and find

$$\tau_{\text{int}} = 1 + 2 \sum_{t=1}^{\infty} e^{-t/\tau_{\text{exp}}} = 1 + \frac{2 e^{-1/\tau_{\text{exp}}}}{1 - e^{-1/\tau_{\text{exp}}}}. \quad (100)$$

For a large exponential autocorrelation time $\tau_{\text{exp}} \gg 1$ the approximation

$$\tau_{\text{int}} = 1 + \frac{2 e^{-1/\tau_{\text{exp}}}}{1 - e^{-1/\tau_{\text{exp}}}} \cong 1 + \frac{2 - 2/\tau_{\text{exp}}}{1/\tau_{\text{exp}}} = 2\tau_{\text{exp}} - 1 \cong 2\tau_{\text{exp}} \quad (101)$$

holds.

10.2. Integrated Autocorrelation Time and Binning

Using binning the integrated autocorrelation time can also be estimated via the variance ratio. We bin the time series (88) into $N_{bs} \leq N$ bins of

$$N_b = \text{NBIN} = \left\lfloor \frac{N}{N_{bs}} \right\rfloor = \left\lfloor \frac{\text{NDAT}}{\text{NBINS}} \right\rfloor \quad (102)$$

data each. Here $\lfloor \cdot \rfloor$ stands for Fortran integer division, *i.e.*, $N_b = \text{NBIN}$ is the largest integer $\leq N/N_{bs}$, implying $N_{ba} \cdot N_b \leq N$. It is convenient to choose the values of N and N_{bs} so that N is a multiple of N_{bs} . The binned data are the averages

$$f_j^{N_b} = \frac{1}{N_b} \sum_{i=1+(j-1)N_b}^{jN_b} f_i \quad \text{for } j = 1, \dots, N_{bs}. \quad (103)$$

For $N_b > \tau_{\text{exp}}$ the autocorrelations are essentially reduced to those between nearest neighbor bins and even these approach zero under further increase of the binsize.

For a set of N_{bs} binned data $f_j^{N_b}$, ($j = 1, \dots, N_{bs}$) we may calculate the mean with its naive error bar. Assuming for the moment an infinite time series, we find the integrated autocorrelation time (95) from the following ratio of sample variances

$$\tau_{\text{int}} = \lim_{N_b \rightarrow \infty} \tau_{\text{int}}^{N_b} \quad \text{with} \quad \tau_{\text{int}}^{N_b} = \left(\frac{s_{f_j^{N_b}}^2}{s_f^2} \right). \quad (104)$$

In practice the $N_b \rightarrow \infty$ limit will be reached for a sufficiently large, finite value of N_b . The statistical error of the τ_{int} estimate (104) is, in the first approximation, determined by the errors of $s_{f_j^{N_b}}^2$. The typical situation is then that, due to the central limit theorem, the binned data are approximately Gaussian, so that the **error of $s_{f_j^{N_b}}^2$ is analytically known** from the χ^2 distribution. Finally, the fluctuations of s_f^2 of the denominator give rise to a small correction which can be worked out [7].

Numerically most accurate estimates of τ_{int} are obtained for the finite binsize N_b which is just large enough that the binned data (103) are practically uncorrelated. While the Student distribution shows that the confidence intervals of the error bars from 16 uncorrelated normal data are reasonable approximations to those of the Gaussian standard deviation, about 1000 independent data are needed to provide a decent estimate of the corresponding variance (at the 95% confidence level with an accuracy of slightly better than 10%). It makes sense to work with error bars from 16 binned data, but the error of the error bar, and hence a reliable estimate of τ_{int} , requires far more data.

10.3. *Illustration: Metropolis Generation of Normally Distributed Data*

We generate normally distributed data according to the Markov process

$$x' = x + 2a x^r - a \quad (105)$$

where x is the event at hand, x^r a uniformly distributed random number in the range $[0, 1)$, and the real number $a > 0$ is a parameter which relates to the efficiency of the algorithm. The new event x' is accepted with the

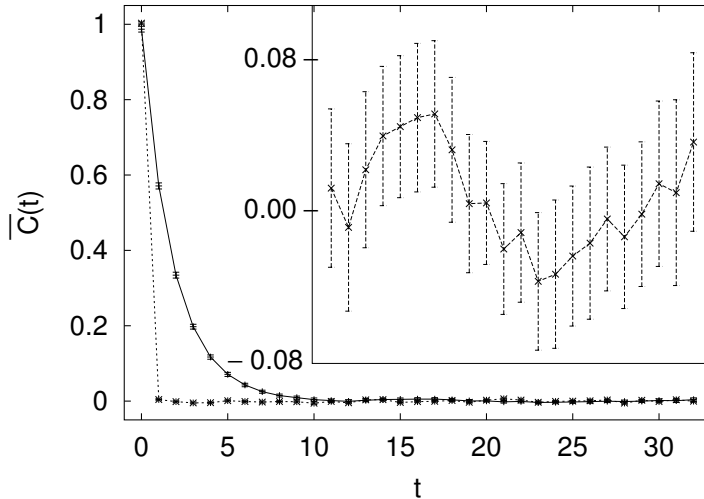


Fig. 9. The autocorrelation function (90) of a Metropolis time series for the normal distribution (upper data) in comparison with those of our Gaussian random number generator (lower data). For $t \geq 11$ the inlay shows the autocorrelations on an enlarged ordinate. The straight lines between the data points are just to guide the eyes. The curves start with $\overline{C}(0) \approx 1$ because the variance of the normal distribution is one.

Metropolis probability

$$P_{\text{accept}}(x') = \begin{cases} 1 & \text{for } x'^2 \leq x^2; \\ \exp[-(x'^2 - x^2)/2] & \text{for } x'^2 > x^2. \end{cases} \tag{106}$$

If x' is rejected, the event x is counted again. The Metropolis process introduces an autocorrelation time in the generation of normally distributed random data.

We work with $N = 2^{17} = 131072$ data and take $a = 3$ for the Markov process (105), what gives an acceptance rate of approximately 50%. The autocorrelation function of this process is depicted in Fig. 9 (assignment a0401_01). The integrated autocorrelation time (assignment a0401_02) is shown in Fig. 10. We compare the $\tau_{\text{int}}^{N_b}$ estimators with the direct estimators $\tau_{\text{int}}(t)$ at

$$t = N_b - 1. \tag{107}$$

With this relation the estimators agree for binsize $N_b = 1$ and for larger N_b the relation gives the range over which we combine data into either

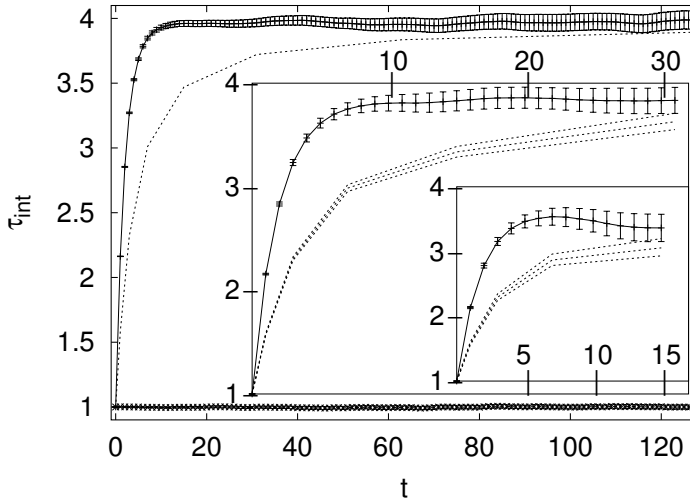


Fig. 10. The upper curves in the figure and its inlays display the estimators obtained by direct calculation. The lowest curve is for the Gaussian random number generator. The remaining curves are binning procedure estimators of the integrated autocorrelation time with one standard deviation bounds. The main figure relies on 2^{21} data and depicts estimators up to $t = 127$. The first inlay relies on 2^{17} data and depicts estimators up to $t = 31$. The second inlay relies on 2^{14} data and depicts estimators up to $t = 15$.

one of the estimators. The approach of the binning procedure towards the asymptotic τ_{int} value is slower than that of the direct estimate of τ_{int} .

For our large $\text{NDAT} = 2^{21}$ data set $\tau_{int}(t)$ reaches its plateau before $t = 20$. All the error bars within the plateau are strongly correlated. Therefore, it is not recommended to make an attempt to combine them. Instead, it is safe to pick an appropriate single value and its error bar as the final estimate:

$$\tau_{int} = \tau_{int}(20) = 3.962 \pm 0.024 \text{ from } 2^{21} = 2,097,152 \text{ data.} \quad (108)$$

The binning procedure, on the other hand, shows an increase of $\tau_{int}^{N_b}$ all the way to $N_b = 2^7 = 128$, where the estimate with the one confidence level error bounds is

$$3.85 \leq \tau_{int}^{128} \leq 3.94 \text{ from } 2^{14} = 16,384 \text{ bins from } 2^{21} \text{ data.} \quad (109)$$

How many data are needed to allow for a meaningful estimate of the integrated autocorrelation time?

For a statistics of $\text{NDAT} = 2^{17}$ the autocorrelation signal disappears for $t \geq 11$ into the statistical noise. Still, there is clear evidence of the hoped

for window of almost constant estimates. A conservative choice is to take $t = 20$ again, which now gives

$$\tau_{\text{int}} = \tau_{\text{int}}(20) = 3.86 \pm 0.11 \text{ from } 2^{17} \text{ data.} \tag{110}$$

Worse is the binning estimate, which for the 2^{17} data is

$$3.55 \leq \tau_{\text{int}}^{32} \leq 3.71 \text{ from } 2^{12} = 4,096 \text{ bins from } 2^{17} = 131,072 \text{ data.} \tag{111}$$

Our best value (108) is no longer covered by the two standard deviation zone.

For the second inlay the statistics is reduced to $\text{NDAT} = 2^{14}$. With the integrated autocorrelation time rounded to 4, this is 4096 times τ_{int} . For binsize $N_b = 2^4 = 16$ we are then down to $N_{bs} = 1024$ bins, which are needed for accurate error bars of the variance. To work with this number we limit, in accordance with equation (107), our $\tau_{\text{int}}(t)$ plot to the range $t \leq 15$. Still, we find a quite nice window of nearly constant $\tau_{\text{int}}(t)$, namely all the way from $t = 4$ to $t = 15$. By a statistical fluctuation (assignment a0401_03) $\tau_{\text{int}}(t)$ takes its maximum value at $t = 7$ and this makes $\tau_{\text{int}}(7) = 3.54 \pm 0.13$ a natural candidate. However, this value is inconsistent with our best estimate (108). The true $\tau_{\text{int}}(t)$ increases monotonically as function of t , so we know that the estimators have become bad for $t > 7$. The error bar at $t = 7$ is too small to take care of our difficulties. One may combine the $t = 15$ error bar with the $t = 7$ estimate. In this way the result is

$$\tau_{\text{int}} = 3.54 \pm 0.21 \text{ for } 2^{14} = 16,384 \text{ data,} \tag{112}$$

which achieves consistency with (108) in the two error bar range. For binsize $N_b = 16$ the binning estimate is

$$2.93 \leq \tau_{\text{int}}^{16} \leq 3.20 \text{ from } 2^{10} = 1,024 \text{ bins from } 2^{14} \text{ data.} \tag{113}$$

Clearly, the binsize $N_b = 16$ is too small for an estimate of the integrated autocorrelation time. We learn that one needs a binsize of at least ten times the integrated autocorrelation time τ_{int} , whereas for its direct estimate it is sufficient to have t about four times larger than τ_{int} .

11. Self-Consistent versus Reasonable Error Analysis

By visual inspection of the time series, one may get an impression about the length of the out-of-equilibrium part of the simulation. On top of this one should still choose

$$\text{nequi} \gg \tau_{\text{int}} , \tag{114}$$

to allow the system to settle down. That is a first reason, why it appears necessary to control the integrated autocorrelation time of a MC simulation. A second reason is that we have to control the error bars of the equilibrium part of our simulation. Ideally the error bars are calculated as

$$\Delta \bar{f} = \sqrt{\sigma^2(\bar{f})} \quad \text{with} \quad \sigma^2(\bar{f}) = \tau_{\text{int}} \frac{\sigma^2(f)}{N} . \quad (115)$$

This constitutes a **self-consistent error analysis** of a MC simulation.

However, the calculation of the integrated autocorrelation time may be out of reach. Many more than the about twenty independent data are needed, which according to the Student distribution are sufficient to estimate mean values with reasonably reliable error bars.

In practice, one has to be content with what can be done. **Often this means to rely on the binning method.** We simply calculate error bars of our ever increasing statistics with respect to a fixed number of

$$\text{NBINS} \geq 16 . \quad (116)$$

In addition, we may put 10% of the initially planned simulation time away for reaching equilibrium. *A-posteriori*, this can always be increased. Once the statistics is large enough, our small number of binned data become effectively independent and our error analysis is justified.

How do we know that the statistics has become large enough? In practical applications there can be indirect arguments, like FSS estimates, which tell us that the integrated autocorrelation time is in fact (much) smaller than the achieved bin length. This is no longer self-consistent, as we perform no explicit measurement of τ_{int} , but it is a **reasonable error analysis**.

12. Comparison of Markov Chain MC Algorithms

Is the 1-hit Metropolis algorithm more efficient with sequential updating or with random updating? For $2d$ Ising lattices at $\beta = 0.4$ Fig. 11 illustrates that sequential updating wins. This is apparently related to the fact that random updating may miss out on some spins for some time, whereas sequential updating touches each spin with certainty during one sweep.

Figures 12 and 13 illustrate $2d$ Ising model simulations off and on the critical point. Off the critical point, at $\beta = 0.4$, the integrated autocorrelation time increases for $L = 5, 10$ and 20 . Subsequently, it decreases to approach for $L \rightarrow \infty$ a finite asymptotic value. On the critical point, at $\beta = \beta_c = \ln(1 + \sqrt{2})/2$, **critical slowing down** is observed, an increase

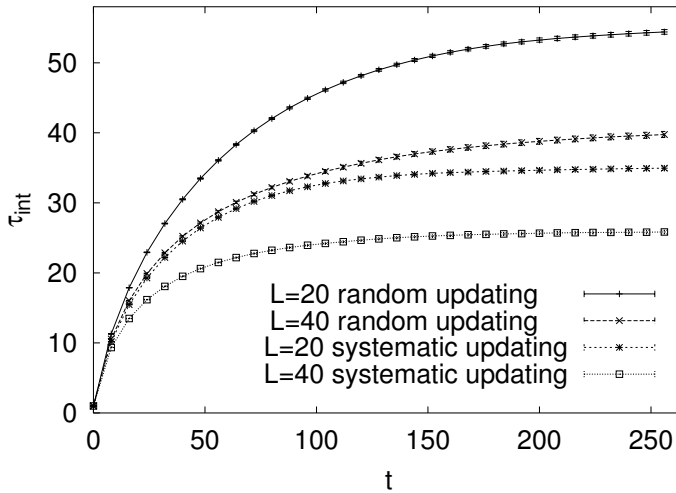


Fig. 11. Comparison of the integrated autocorrelation time of the Metropolis process with random updating versus sequential updating for the $d = 2$ Ising model at $\beta = 0.4$ (assignment a0402_01 B). The ordering of the curves is identical with the ordering of the labels in the figure.

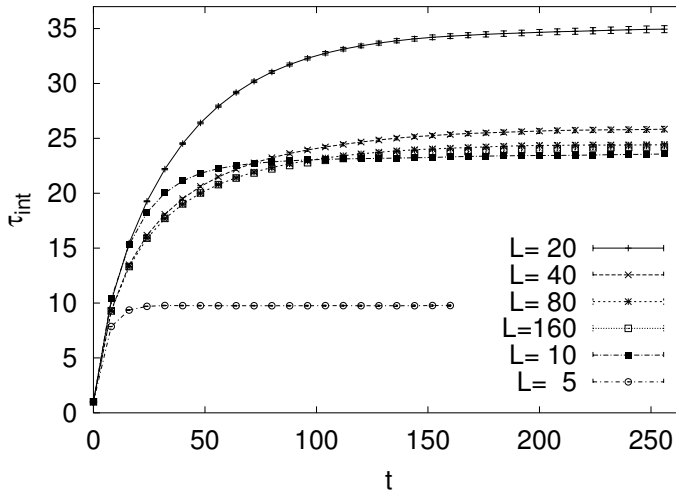


Fig. 12. One-hit Metropolis algorithm with sequential updating; Lattice size dependence of the integrated autocorrelation time for the $d = 2$ Ising model at $\beta = 0.4$ (assignment a0402_01 A). The ordering of the curves is identical with the ordering of the labels in the figure.

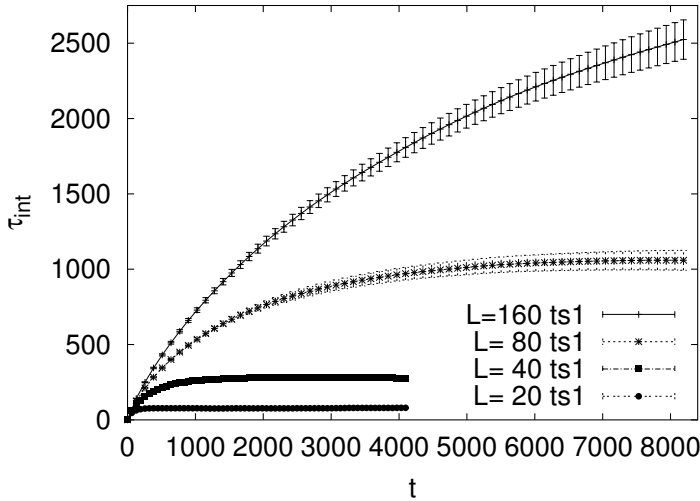


Fig. 13. One-hit Metropolis algorithm with sequential updating; Lattice size dependence of the integrated autocorrelation time for the $d = 2$ Ising model at its critical temperature (assignment a0402_02 D). The ordering of the curves is identical with the ordering of the labels in the figure.

$\tau_{\text{int}} \sim L^z$ with lattice size, where $z \approx 2.17$ is the **dynamical critical exponent** of the $2d$ Ising model. Estimates of z are compiled in the book by Landau and Binder [16].

Using another MC dynamics the critical slowing down can be overcome. Fig. 14 shows the major improvements for Swendsen-Wang [21] (SW) and Wolff [24] (W) cluster updating.

Finally, Fig. 15 exhibit the improvements of heat bath over Metropolis updating for the 10-state $d = 2$ Potts model at $\beta = 0.62$.

13. Multicanonical Simulations

One of the questions which ought to be addressed before performing a large scale computer simulation is “What are suitable weight factors for the problem at hand?” So far we used the Boltzmann weights as this appears natural for simulating the canonical ensemble. However, a broader view of the issue is appropriate.

Conventional, canonical simulations calculate expectation values at a fixed temperature T and can, by re-weighting techniques, only be extrapolated to a vicinity of this temperature. For multicanonical simulations this is

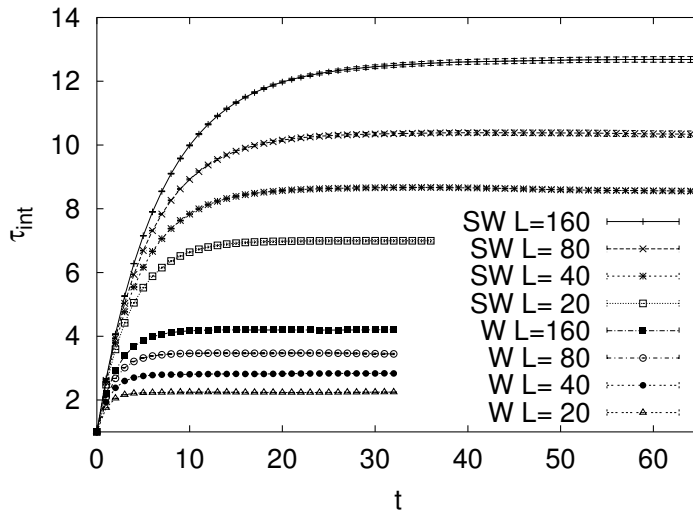


Fig. 14. Cluster algorithms: Estimates of integrated autocorrelation times from simulations of the $d = 2$ Ising model at the critical temperature $\beta_c = 0.44068679351$ (assignment a0503_05).

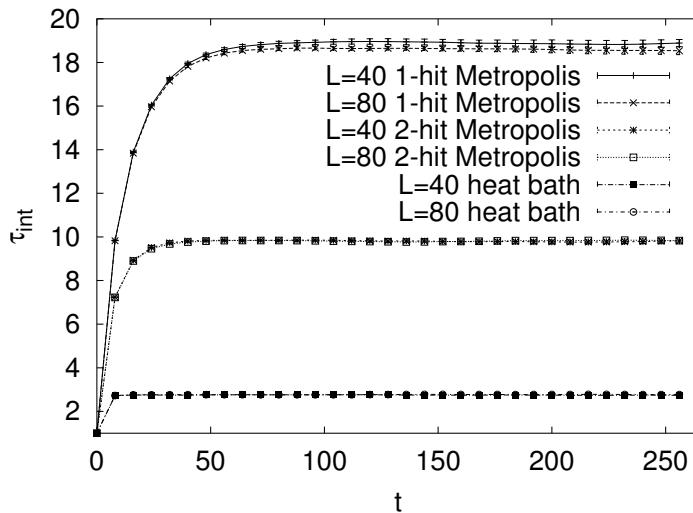


Fig. 15. Systematic updating: Comparison of the integrated autocorrelation times of the 1-hit and 2-hit Metropolis algorithms and the heat bath algorithm for the 10-state Potts model on $L \times L$ lattices at $\beta = 0.62$ (assignment a0402_06). The $L = 40$ and $L = 80$ curves lie almost on top of one another.

different. A single simulation allows to obtain equilibrium properties of the Gibbs ensemble over a range of temperatures. Of particular interest are two situations for which canonical simulations do not provide the appropriate implementation of importance sampling:

- (1) The physically important configurations are rare in the canonical ensemble.
- (2) A rugged free energy landscape makes the physically important configurations difficult to reach.

MC calculation of the interface tension of a first order phase transition provide an example for the first situation. Let $N = L^d$ be the lattice size. For a first order phase transition **pseudo-transition temperatures** $\beta^c(L)$ exist so that the energy distributions $P(E) = P(E; L)$ become double peaked and the maxima at $E_{\max}^1 < E_{\max}^2$ are of equal height $P_{\max} = P(E_{\max}^1) = P(E_{\max}^2)$. In-between the maximum values a minimum is located at some energy E_{\min} . Configurations at E_{\min} are exponentially suppressed like

$$P_{\min} = P(E_{\min}) = c_f L^p \exp(-f^s A) \quad (117)$$

where f^s is the interface tension and A is the minimal area between the phases, $A = 2L^{d-1}$ for an L^d lattice, c_f and p are constants (computations of p have been done in the capillary-wave approximation). The interface tension can be determined by Binder's histogram method [8]. One has to calculate the quantities

$$f^s(L) = -\frac{1}{A(L)} \ln R(L) \quad \text{with} \quad R(L) = \frac{P_{\min}(L)}{P_{\max}(L)} \quad (118)$$

and to make a FSS extrapolation of $f^s(L)$ for $L \rightarrow \infty$.

For large systems a canonical MC simulation will practically never visit configurations at energy $E = E_{\min}$ and estimates of the ratio $R(L)$ will be very inaccurate. The terminology **supercritical slowing down** was coined to characterize such an exponential deterioration of simulation results with lattice size.

Multicanonical simulations [3] approach this problem by sampling, in an appropriate energy range, with an **approximation** to the weights

$$w_{1/n}(E^{(k)}) = \frac{1}{n(E^{(k)})} = \exp \left[-b(E^{(k)}) E + a(E^{(k)}) \right] \quad (119)$$

where $n(E)$ is the number of states of energy E . The function $b(E)$ defines the inverse **microcanonical temperature** and $a(E)$ the **dimensionless**,

microcanonical free energy. The function $b(E)$ has a relatively smooth dependence on its arguments, which makes it a useful quantity when dealing with the weight factors.

Instead of the canonical energy distribution $P(E)$, one samples a new multicanonical distribution

$$P_{mu}(E) = c_{mu} n(E) w_{mu}(E) \approx c_{mu} . \quad (120)$$

The desired canonical probability density is obtained by re-weighting

$$P(E) = \frac{c_{\beta}}{c_{mu}} \frac{P_{mu}(E)}{w_{mu}(E)} e^{-\beta E}. \quad (121)$$

This relation is rigorous, because the weights $w_{mu}(E)$ used in the simulation are exactly known. Accurate estimates of the interface tension (118) become possible.

The multicanonical method requires two steps:

- (1) Obtain a working estimate $\hat{w}_{mu}(k)$ of the weights $\hat{w}_{1/n}(k)$. Working estimate means that the approximation to (119) has to be good enough to ensure movement in the desired energy range.
- (2) Perform a Markov chain MC simulation with the fixed weights $\hat{w}_{mu}(k)$. The thus generated configurations constitute the multicanonical ensemble. Canonical expectation values are found by re-weighting to the Gibbs ensemble and jackknife methods allow reliable error estimates.

It is a strength of computer simulations that one can generate artificial (not realized by nature) ensembles, which enhance the probabilities of rare events one may be interested in, or speed up the dynamics. Nowadays Generalized Ensembles (umbrella, multicanonical, 1/k, ...) have found many applications. Besides for first order phase transitions they are in particular useful for complex systems such as biomolecules, where they accelerate the dynamics. For a review see [14].

13.1. How to Get the Weights?

To get the weights is at the heart of the method. Some approaches are:

- (1) Overlapping, constrained (microcanonical) MC simulations. A potential problem is to fulfill ergodicity.
- (2) FSS Estimates. This appears to be best when it works, but there may be no FSS theory for the system at hand.
- (3) General Purpose Recursions. Problem: They tend to deteriorate with increasing lattice size (large lattices).

The Multicanonical Recursion (a variant of [4]): The multicanonical parameterization of the weights is

$$w(a) = e^{-S(E_a)} = e^{-b(E_a) E_a + a(E_a)},$$

where (for ϵ being the smallest energy stepsize)

$$b(E) = [S(E + \epsilon) - S(E)] / \epsilon \quad \text{and} \quad a(E - \epsilon) = a(E) + [b(E - \epsilon) - b(E)] E.$$

The recursion reads then (see [6] for details):

$$b^{n+1}(E) = b^n(E) + \hat{g}_0^n(E) [\ln H^n(E + \epsilon) - \ln H^n(E)] / \epsilon$$

$$\hat{g}_0^n(E) = g_0^n(E) / [g^n(E) + \hat{g}_0^n(E)],$$

$$g_0^n(E) = H^n(E + \epsilon) H^n(E) / [H^n(E + \epsilon) + H^n(E)],$$

$$g^{n+1}(E) = g^n(E) + g_0^n(E), \quad g^0(E) = 0.$$

The Wang-Landau Recursion [23]: Updates are performed with estimators $g(E)$ of the density of states

$$p(E_1 \rightarrow E_2) = \min \left[\frac{g(E_1)}{g(E_2)}, 1 \right].$$

Each time an energy level is visited, the estimator of $g(E)$ is updated according to

$$g(E) \rightarrow g(E) f$$

where, initially, $g(E) = 1$ and $f = f_0 = e^1$. Once the desired energy range is covered, the factor f is refined:

$$f_1 = \sqrt{f}, \quad f_{n+1} = \sqrt{f_{n+1}}$$

until some value very close to one like $f = 1.00000001$ is reached. Afterwards the usual multicanonical production runs may be carried out.

14. Multicanonical Example Runs (2d Ising and Potts Models)

Most illustrations of this section are from Ref. [6].

For an Ising model on a 20×20 lattice the multicanonical recursion is run in the range

$$\text{namin} = 400 \leq \text{iact} \leq 800 = \text{namax}. \quad (122)$$

The recursion is terminated after a number of so called tunneling events. A **tunneling event** is defined as an updating process which finds its way from

$$\text{iact} = \text{namin} \text{ to } \text{iact} = \text{namax} \text{ and back} . \tag{123}$$

This notation comes from applications to first order phase transitions. An alternative notation for tunneling event is **random walk cycle**. For most applications 10 tunneling events lead to acceptable weights.

For the Ising model example run we find the requested 10 tunneling events after 787 recursions and 64,138 sweeps (assignment a0501_01). In assignment a0501_02 a similar example run is performed for the 2d 10-state Potts model.

Performance: If the multicanonical weighting would remove all relevant free energy barriers, the behavior of the updating process would become that of a free **random walk**. Therefore, the theoretically optimal performance for the second part of the multicanonical simulation is

$$\tau_{\text{tun}} \sim V^2 . \tag{124}$$

Recent work about first order transitions by Neuhaus and Hager [19] shows that the multicanonical procedure removes only the leading free energy barrier, while at least one subleading barrier causes a residual supercritical slowing done. Up to certain medium sized lattices the behavior $V^{2+\epsilon}$ gives a rather good effective description. For large lattices exponential slowing down dominates again. The slowing down of the weight recursion with the volume size is expected to be even (slightly) worse than that of the second part of the simulation.

Re-Weighting to the Canonical Ensemble: Let us assume that we have performed a multicanonical simulation which covers the energy histograms for a temperature range

$$\beta_{\min} \leq \beta = \frac{1}{T} \leq \beta_{\max} . \tag{125}$$

Given the multicanonical time series, where $i = 1, \dots, n$ labels the generated configurations, the formula

$$\bar{\mathcal{O}} = \frac{\sum_{i=1}^n \mathcal{O}^{(i)} \exp[-\beta E^{(i)} + b(E^{(i)}) E^{(i)} - a(E^{(i)})]}{\sum_{i=1}^n \exp[-\beta E^{(i)} + b(E^{(i)}) E^{(i)} - a(E^{(i)})]} . \tag{126}$$

replaces the multicanonical weighting of the simulation by the Boltzmann factor. The denominator differs from the partition function Z by a constant factor which drops out against the same constant factor in the numerator.

For discrete systems it is sufficient to keep histograms when only functions of the energy are calculated. For an operator $\mathcal{O}^{(i)} = f(E^{(i)})$ equation (126) simplifies to

$$\bar{f} = \frac{\sum_E f(E) h_{mu}(E) \exp[-\beta E + b(E) E - a(E)]}{\sum_E h_{mu}(E) \exp[-\beta E + b(E) E - a(E)]} \quad (127)$$

where $h_{mu}(E)$ is the histogram sampled during the multicanonical production run and the sums are over all energy values for which $h_{mu}(E)$ has entries.

The computer implementation of these equations requires care. The differences between the largest and the smallest numbers encountered in the exponents can be really large. We can avoid large numbers by dealing only with logarithms of sums and partial sums. For $C = A + B$ with $A > 0$ and $B > 0$ we can calculate $\ln C = \ln(A + B)$ from the values $\ln A$ and $\ln B$, without ever storing either A or B or C (see [7] for more details):

$$\begin{aligned} \ln C &= \ln \left[\max(A, B) \left(1 + \frac{\min(A, B)}{\max(A, B)} \right) \right] \\ &= \max(\ln A, \ln B) + \ln \{ 1 + \exp[\min(\ln A, \ln B) - \max(\ln A, \ln B)] \} \\ &= \max(\ln A, \ln B) + \ln \{ 1 + \exp[-|\ln A - \ln B|] \} . \end{aligned} \quad (128)$$

14.1. Energy and Specific Heat Calculation

We are now ready to produce multicanonical data for the energy per spin of the $2d$ Ising model on a 20×20 lattice (assignment `a0501_03`). The same numerical data allow to calculate the **specific heat** defined by

$$C = \frac{d\hat{E}}{dT} = \beta^2 (\langle E^2 \rangle - \langle E \rangle^2) . \quad (129)$$

The comparison of the multicanonical specific heat data with the exact curve of Ferdinand and Fisher [11] is shown in Fig. 16 (error bars rely on the jackknife method).

The energy histogram of this multicanonical simulation together its canonically re-weighted descendants at $\beta = 0$, $\beta = 0.2$ and $\beta = 0.4$ is shown in Fig. 17. The normalization of the multicanonical histogram is adjusted so that it fits into the same figure with the three re-weighted histograms.

It is assignment `a0501_06` to produce similar data for the $2d$ 10-state Potts model and to re-weighted the multicanonical energy histogram to the canonical distribution at $\beta = 0.71$, which is close to the pseudo-transition

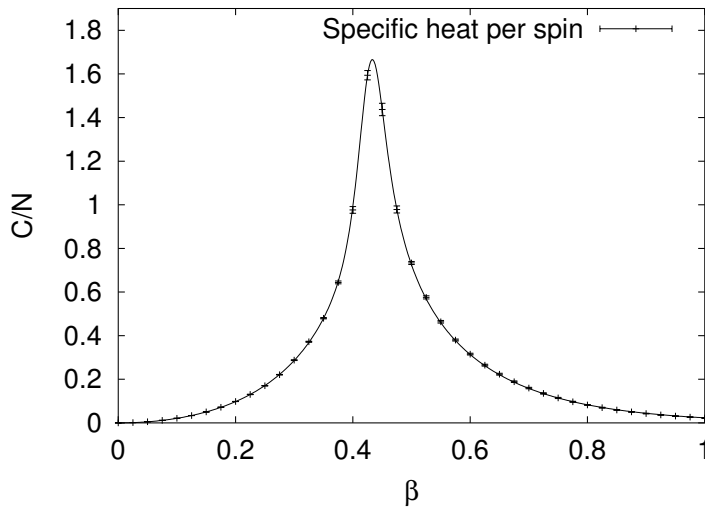


Fig. 16. Specific heat per spin for the Ising model on a 20×20 lattice: Multicanonical data versus exact results of Ferdinand and Fisher. This figure was first published in [6].

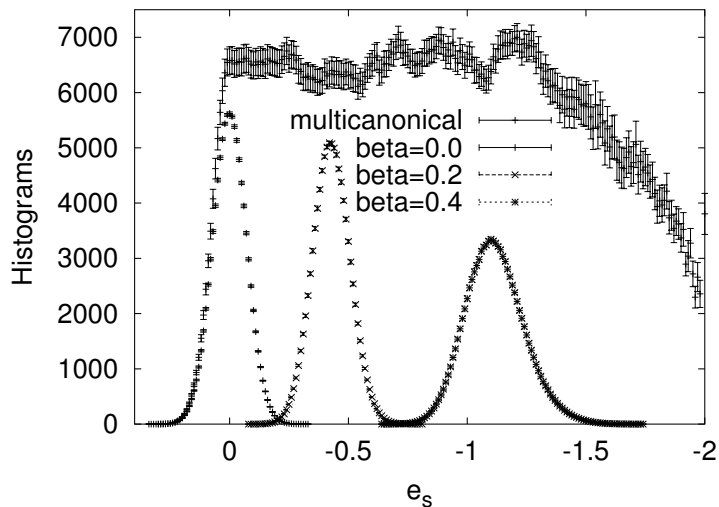


Fig. 17. Energy histogram from a multicanonical simulation of the Ising model on a 20×20 lattice together with canonically re-weighted histograms (assignment a0501_04). This figure was first published in [6].

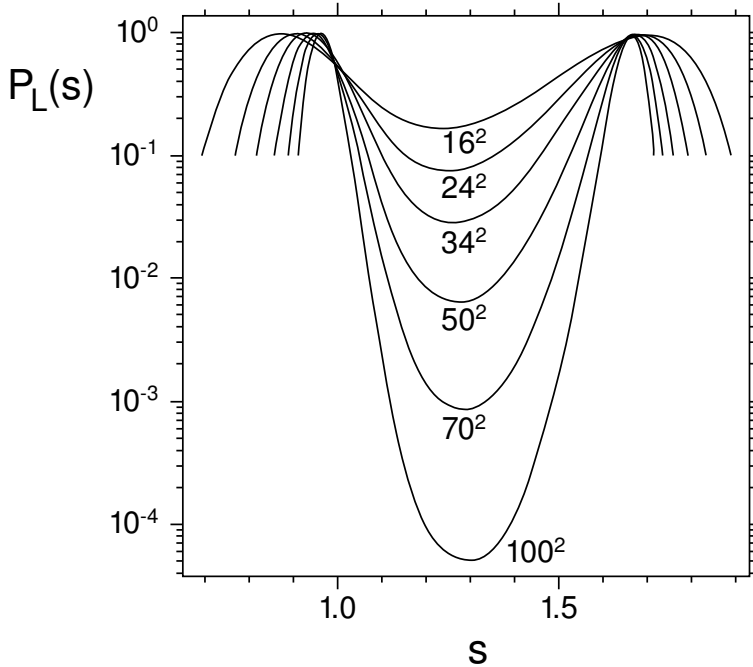


Fig. 18. Energy histogram, $e_s = -2s + 2$, for the $2d$ 10-state Potts models on various lattice sizes (re-drawn after Ref. [3] from where the notation for s comes).

temperature. The multicanonical method allows then to estimate the interface tension of the transition by following the minimum to maximum ratio $R(L)$ of Eq. (118) over many orders of magnitude [3] as is shown in Fig. 18.

14.2. Free Energy and Entropy Calculation

At $\beta = 0$ the Potts partition function is $Z = q^N$. Therefore, multicanonical simulations allow for proper normalization of the partition function, if $\beta = 0$ is included in the temperature range. The properly normalized partition function allows to calculate the **Helmholtz free energy**

$$F = -\beta^{-1} \ln(Z) \quad (130)$$

and the **entropy**

$$S = \frac{F - E}{T} = \beta(F - E) \quad (131)$$

of the canonical ensemble. Here E is the expectation value of the internal energy and the last equal sign holds because of our choice of units for the

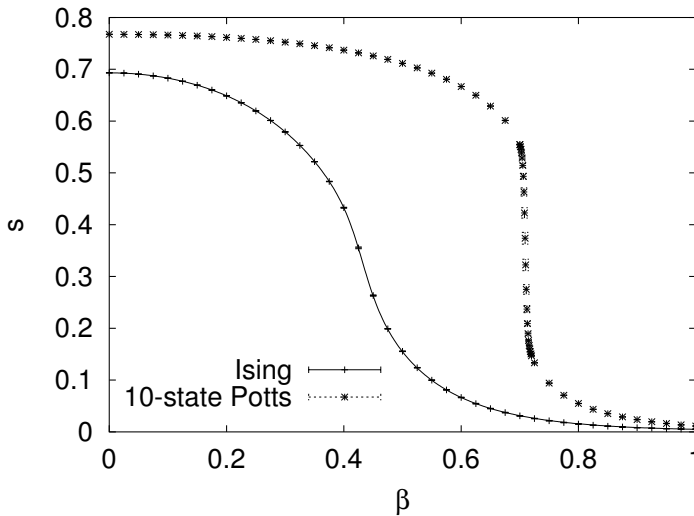


Fig. 19. Entropies per spin, $s = S/N$, from multicanonical simulations of the Ising and 10-state Potts models on a 20×20 lattice (assignments a0501_03 and a0501_05). The full line is the exact result of Ferdinand and Fischer for the Ising model.

temperature. For the $2d$ Ising model as well as for the $2d$ 10-state Potts model, we show in Fig. 19 multicanonical estimates of the entropy density per site

$$s = S/N . \tag{132}$$

For the $2d$ Ising model one may also compare directly with the number of states. Up to medium sized lattices these integers can be calculated to all digits by analytical methods [2]. However, MC results are only sensitive to the first few (not more than six) digits and, therefore, one finds no real advantages over using other physical quantities.

14.3. Time Series Analysis

Typically, one prefers in continuous systems time series data over keeping histograms, because one avoids then discretization errors [7]. Even in discrete systems time series data are of importance, as one often wants to measure more physical quantities than just the energy. Then RAM storage limitations may require to use a time series instead of histograms. To illustrate this point, we use the Potts magnetization.

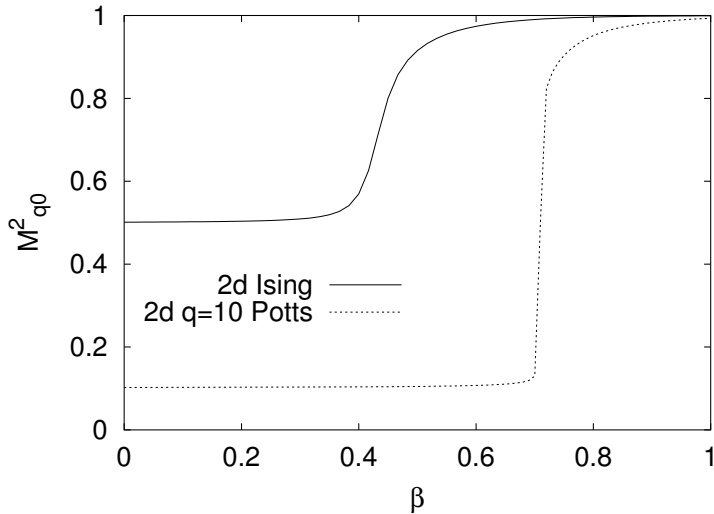


Fig. 20. The Potts magnetization squared per lattice site for the $q = 2$ and $q = 10$ Potts models on a 20×20 lattice (assignments a0501_08 and a0501_09).

In assignments a0501_08 and a0501_09 we create the same statistics on 20×20 lattices as before, including time series measurements for the energy and for the Potts magnetization. For energy based observables the analysis of the histogram and the time series data give consistent results.

For zero magnetic field, $H = 0$, the expectation value of the Potts magnetization on a finite lattice is simply

$$M_{q0} = \langle \delta_{q_i, q_0} \rangle = \frac{1}{q}, \quad (133)$$

independently of the temperature. For the multicanonical simulation it is quite obvious that even at low temperatures each Potts state is visited with probability $1/q$. In contrast to this, the expectation value of the magnetization squared

$$M_{q0}^2 = q \left\langle \left(\frac{1}{N} \sum_{i=1}^N \delta_{q_i, q_0} \right)^2 \right\rangle \quad (134)$$

is a non-trivial quantity. At $\beta = 0$ its value is $M_{q0}^2 = q(1/q)^2 = 1/q$, whereas it approaches 1 for $N \rightarrow \infty$, $\beta \rightarrow \infty$. For $q = 2$ and $q = 10$ Fig. 20 shows our numerical results and we see that the crossover of M_{q0}^2 from $1/q$ to 1 happens in the neighborhood of the critical temperature. A FSS analysis

would reveal that a singularity develops at β_c , which is in the derivative of M_{q0}^2 for the second order phase transitions ($q \leq 4$) and in M_{q0}^2 itself for the first order transitions ($q \geq 5$).

Acknowledgments

I thank Professor Louis Chen and the IMS staff for their kind hospitality. While visiting the IMS I greatly benefitted from discussions with Professors Wolfhard Janke, David Landau, Robert Swendsen and Jian-Sheng Wang.

References

1. R.J. Baxter, *Potts Models at the Critical Temperature*, J. Phys. C **8** (1973), L445–L448.
2. P.D. Beale, *Exact Distribution of Energies in the Two-Dimensional Ising Model*, Phys. Rev. Lett. **76** (1996), 78–81.
3. B.A. Berg and T. Neuhaus, *Multicanonical Ensemble: A New Approach to Simulate First-Order Phase Transitions*, Phys. Rev. Lett. **68** (1992), 9–12.
4. B.A. Berg, *Multicanonical Recursions*, J. Stat. Phys. **82** (1996), 323–342.
5. B.A. Berg, A. Billoire and W. Janke, *Spin Glass Overlap Barriers in Three and Four Dimensions*, Phys. Rev. B **61** (2000), 12143–12150.
6. B.A. Berg, *Multicanonical Simulations Step by Step*, Comp. Phys. Commun. **153** (2003), 397–406.
7. B.A. Berg, *Markov Chain Monte Carlo Simulations and Their Statistical Analysis*, World Scientific, Singapore, 2004. Information on the web at <http://www.hep.fsu.edu/~berg>.
8. K. Binder, *The Monte Carlo Calculation of the Surface Tensions for Two- and Three-Dimensional Lattice-Gas Models*, Phys. Rev. A **25** (1982), 1699–1709.
9. C. Borgs and W. Janke, *An Explicit Formula for the Interface Tension of the 2D Potts Model*, J. Phys. I France **2** (1992), 2011–2018.
10. P.M. Chaikin and T.C. Lubensky, *Principles of condensed matter physics*, Cambridge University Press, 1997, table 8.6.1, p.467.
11. A.E. Ferdinand and M.E. Fisher, *Bounded and Inhomogeneous Ising Models. I. Specific-Heat Anomaly of a Finite Lattice*, Phys. Rev. **185** (1969), 832–846.
12. A. Ferrenberg and R. Swendsen, *New Monte Carlo Technique for Studying Phase Transitions*, Phys. Rev. Lett. **61** (1988), 2635–2638; **63** (1989), 1658.
13. J. Gubernatis (editor), *The Monte Carlo Method in the Physical Sciences: Celebrating the 50th Anniversary of the Metropolis Algorithm*, AIP Conference Proceedings, Volume 690, Melville, NY, 2003.
14. U.H. Hansmann and Y. Okamoto, *The Generalized-Ensemble Approach for Protein Folding Simulations*, Ann. Rev. Comp. Phys. **6** (1999), 129–157.
15. D.E. Knuth, *Fundamental Algorithms*, Vol.1 of *The Art of Computer Programming*, Addison-Wesley, Reading, MA, 1968.

16. D.P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics*, Cambridge University Press, Cambridge, 2000.
17. G. Marsaglia, A. Zaman and W.W. Tsang, *Toward a Universal Random Number Generator*, *Stat. Prob.* **8** (1990), 35–39.
18. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller, *Equation of State Calculations by Fast Computing Machines*, *J. Chem. Phys.* **21** (1953), 1087–1092.
19. T. Neuhaus and J.S. Hager, *2d Crystal Shapes, Droplet Condensation and Supercritical Slowing Down in Simulations of First Order Phase Transitions*, *J. Stat. Phys.* **113** (2003), 47–83.
20. Student, *The Probable Error of a Mean*, *Biometrika* **6** (1908), 1–25.
21. R.H. Swendsen and J.-S. Wang, *Nonuniversal Critical Dynamics in Monte Carlo Simulations*, *Phys. Rev. Lett.* **58** (1987), 86–88.
22. I. Vattulainen, T. Ala-Nissila and K. Kankaala, *Physical Models as Tests for Randomness*, *Phys. Rev. E* **52** (1995), 3205–3214.
23. F. Wang and D.P. Landau, *Efficient, Multiple-Range Random Walk Algorithm to Calculate the Density of States*, *Phys. Rev. Lett.* **86** (2001), 2050–2053.
24. U. Wolff, *Collective Monte Carlo Updating for Spin Systems*, *Phys. Rev. Lett.* **62** (1989), 361–363.
25. F.Y. Wu, *The Potts Model*, *Rev. Mod. Phys.* **54** (1982), 235–268.