Manuel Pfeifenberger, BSc

Nested sampling algorithm for Potts models

MASTER THESIS

For obtaining the academic degree
Diplom-Ingenieur

Master Programme of
Technical Physics

Graz University of Technology

Supervisor:
Univ.-Prof. Dipl.-Phys. Dr.rer.nat. Wolfgang von der Linden
Institute of Theoretical and Computational Physics

Graz, March 2015

To my loving parents Roswitha and Walter
and of course Olivia

EIDESSTATTLICHE ERKLÄRUNG

AFFIDAVIT

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit/Diplomarbeit/Dissertation identisch.

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis/diploma thesis/doctoral dissertation.

05.03.2015

Datum / Date

Unterschrift / Signature

Abstract

The evaluation of high dimensional integrals or sums, as occurring in statistical analyses or in statistical physics problems, poses a challenging task for standard Monte Carlo methods, especially if the integrand or summand exhibits a multimodal structure. An example for such a problem is the calculation of the partition function of the Potts model. The energy distribution of the Potts model with $q > 4$, exhibits a well separated phase space, associated with a first order phase transition. A novel, promising way to deal with this issue is the nested sampling algorithm by John Skilling.

Nested sampling and the more established thermodynamic integration are employed to evaluate the partition function of the Potts model. A correlation time analysis of both algorithms shows, that the severe slowing down of thermodynamic integration around the critical temperature does not occur for the nested sampling algorithm.

The scaling exponents of the computational time in dependence of the system size exhibit the same values for both methods, though nested sampling is about three orders of magnitude faster. Further a way to compute physical quantities, hence the derivations of the logarithmic partition function, via the nested sampling results, is presented. Results for the inner energy are compared to known results obtained from a multicanonical simulation. Eventually an approach for a parallel implementation of nested sampling is investigated. Here no significant enhancement in computational performance for the evaluation of the Potts model has been observed. However, the parallel nested sampling is able to use more walkers, which leads to a better sampling of the phase space, while the wall clock time is kept constant. Hence when dealing with problems involving a multiple peaked likelihood, the parallel implementation will become advantageous.

Kurzfassung

Das Berechnen hochdimensionaler Integrale oder Summen, kommt in statistischen Analysen und Problemen der statistischen Physik häufig vor. Es stellt, speziell im Fall multimodaler Integranden und Summanden, eine schwierige Aufgabe für Standard Monte Carlo Methoden dar. Ein Beispiel für ein derartiges Problem ist die Berechnung der Zustandssumme des Potts-Modells. Die Energie Verteilung dieses Modells besitzt für $q > 4$, einen stark separierten Phasenraum, im Zusammenhang mit einem Phasenübergang erster Ordnung. Einen neuartigen und vielversprechenden Weg, solche Aufgaben zu lösen, bietet die Nested Sampling Methode von John Skilling.

Nested Sampling und das etablierte thermodynamische Integrieren werden zur Berechnung der Zustandssumme des Potts-Modells herangezogen. Mittels einer Korrelationszeitanalyse beider Methoden ergibt sich, dass die starke Zunahme der Korrelationszeiten beim thermodynamischen Integrieren nahe der kritischen Temperatur, beim Nested Sampling Algorithmus nicht auftritt. Die Rechenzeit in Abhängigkeit von der Systemgröße skaliert für beide Methoden in etwa quadratisch. Die benötigte Zeit für Nested Sampling liegt allerdings ca. drei Größenordnungen unter der benötigten Zeit für das thermodynamische Integrieren.

Weiters wird angeführt, wie physikalische Größen, d.h. die Ableitungen der logarithmischen Zustandsfunktion, aus den Ergebnissen des Nested Sampling Algorithmus, berechnet werden können. Zusätzlich wird eine Möglichkeit präsentiert den Algorithmus zu parallelisieren. Die Implementierung des parallelen Algorithmus liefert allerdings für das Potts-Modell keine wesentliche Verbesserung. Jedoch ist es durch die parallele Version möglich mehr Walker zu verwenden als im nicht parallelen Fall, was bei konstanter, realer Rechenzeit ein besseres Explorieren des Phasenraums erlaubt. Speziell bei Problemen, welche eine Likelihood Funktion mit mehreren Peaks aufweisen, ist daher die parallele Implementierung vorteilhafter.

Acknowledgements

The task of writing a master thesis, includes many times of joy and frustration, many times of understanding and perplexity. But besides posing a compelling challenge, it is also a magnificent possibility to gain profound knowledge about a certain topic. After all, the completion of my thesis included the support of many people.

First of all I would like to express my gratitude towards Professor Dr. Wolfgang von der Linden, for being an excellent supervisor and for always having an open door for me. The weekly discussions and many helpful inputs, successfully lead me through the creation of this thesis.
I would also like to express my appreciation to Andreas Hirczy, for his support in any computational issue, and to Professor Dr. Hans Gerd Evertz, for stimulating discussions on diverse simulation techniques. A special thanks goes to Dr. Murray, who sent me the source code for the results, published in [18]. This has been an excellent orientation for the improvements and extensions developed during this work.
I would like to thank the people from the Institute of Theoretical and Computational Physics for constructive conversations and useful hints on various issues as well as for the pleasant working atmosphere.

I am very grateful for my loving parents Roswitha and Walter, who have always been encouraging and supportive throughout my studies. Their everlasting support assured me, that I can seek their advice and help, whenever I need it. Further I would also like to thank all my friends, for their support and the enjoyable time. Finally a very special acknowledgement goes to Olivia for her warm and patient love and for being there for me all the time. Her encouragement always makes me feel that anything is possible.

Contents

1 Introduction **1**

2 Potts model **3**

2.1 Fortuin-Kasteleyn random cluster model 5

3 Monte Carlo methods **8**

3.1 Markov Chain Monte Carlo 9

3.1.1 Metropolis Hastings algorithm 10

3.2 Cluster algorithms 12

3.2.1 Introduction of Clusters 12

3.2.2 The Swendsen–Wang algorithm 13

3.2.3 Cluster Identification 14

3.3 Evaluation of the partition function 14

4 Thermodynamic integration **15**

4.1 Thermodynamic integration for Potts models 16

4.1.1 Correlation time analysis 20

5 Multicanonical sampling **24**

5.1 Weight estimate 25

5.2 Evaluation of the partition function 26

6 Nested sampling **28**

6.1 Basics 28

6.2 Nested sampling algorithm 30

6.2.1 Degeneracy 34

6.3 Statistical analysis of the prior masses 35

6.3.1 Order statistic 36

6.3.2 Mean and variance for the estimator of Z 39

6.3.3 Samples of the prior mass . . . 41
6.3.4 Upper bound of variance . . . 43
6.4 Quadrature and truncation error . . . 43
6.5 Parallel implementation . . . 45
6.5.1 Parallel nested sampling algorithm . . . 47
6.5.2 OpenMP . . . 48
6.6 Nested sampling for Potts models . . . 48
6.6.1 Adaption of Nested sampling . . . 48
6.6.2 Brute force calculation of small Potts systems . . . 60
6.6.3 Correlations . . . 62

7 Results **65**
7.1 One dimensional Gaussian likelihood function . . . 65
7.2 Potts model with 10 colours . . . 73
7.3 Thermodynamic quantities . . . 81
7.3.1 Comparison of multicanonical sampling and nested sampling 83
7.4 Performance comparison of thermodynamic integration and nested sampling . . . 84
7.5 Analysis of the parallel nested sampling algorithm . . . 90

8 Conclusion **94**

9 Appendix **96**
9.1 Handling sums of large numbers . . . 96
9.2 Beta and Gamma function . . . 97

Bibliography **98**

List of abbreviations **101**

List of figures **102**

List of tables **105**

1 Introduction

With the massive rise of computer power and the coupled decrease in price over the last decades, the endless possibilities of computer simulations attract an enormous attention in science and industry. For Landau 'computer experiments' are meanwhile one of the three cornerstones, besides theory and experiments, in the attempt of describing nature [14].
One of the most important kind of 'computer experiments' are Monte Carlo algorithms, developed by the physicists Fermi, Ulam, Neumann and Metropolis in the 1940's [13]. Nowadays they are successfully used to solve problems which originate from various scientific disciplines. In principle Monte Carlo methods rely on simulating certain models in a stochastic manner instead of a determined analytical way. Although plenty different prosperous designs of this algorithms exist, problems with a complicated phase space structure, still depict a demanding challenge. Diverse methods for such problems, for example simulated annealing [14] or multicanonical simulations [4], [5], exist. Basically they are trying to flatten structures in phase space in different ways, to enhance the efficiency of the MC algorithm.

A conceptual completely new approach, named *nested sampling*, has been suggested by Skilling in 2004 [22]. It is an ingenious and promising way in estimating high dimensional, multimodal integrals and is based on a clever rearrangement of the integral. Since its development it has already found its way into various fields of research. Especially for solving statistical and Bayesian inference problems many applications already exist (e.g. [2], [6], [8], [17], [20]). A first application in the field of statistical physics was investigated by Murray et al. [18]. They presented a way to compute the partition function of the Potts model via nested sampling. The evaluation of the partition function of that model represents a difficult challenge for standard Monte Carlo algorithms. Especially the first order phase transition occurring for certain parameters, poses a severe difficulty. Further the Potts model provides, although its simple structure, a wide variety of

physically interesting properties. The analytical availability of certain quantities of the two dimensional system makes it an optimal playground for testing new approaches in simulation techniques. A first investigation of the nested sampling algorithm for this model has been conducted by Murray et al. [18].

In the present work the performance of nested sampling in evaluating the partition function of the Potts model is investigated in more detail. Further we present a thorough analysis of the application of nested sampling to this model. The performance of thermodynamic integration, an alternative way of computing the partition function, is used to benchmark nested sampling.
The second chapter gives a short introduction of the investigated Potts model. In chapter 3 the principle of Monte Carlo methods and the idea of cluster algorithms are explained.
The Monte Carlo variants employed by us to compute the partition of the Potts model are found in chapter 4. thermodynamic integration, 5. multicanonical simulation and 6. nested sampling. The first part explaines the basics of the algorithms and the second part deals with their application to the Potts model.
Being in the focus of our investigation nested sampling is treated in a more elaborate way. A parallel implementation of it is presented in section 6.5.
Results for a one dimensional Gaussian likelihood function and the Potts model, as well as a performance comparison of thermodynamic integration and nested sampling are compiled in chapter 7. Further it is depicted how to retrieve thermodynamic quantities from the nested sampling results. Subsequently an analysis of the capability of the parallel implementation is given. Finally the results are discussed and potentially future work is outlined.

2 Potts model

One of the most investigated models in statistical physics is the Ising model. On the one hand it shows a great simplicity, on the other hand it is able to show diverse, interesting phenomena like e.g. phase transitions. Landau even denotes this model as the "fruit fly" in this field of research ([14, p.16]). The Hamiltonian of this model reads as

$$H_{\text{Ising}} = -\sum_{\langle i,j \rangle} J_{ij}\, s_i\, s_j - h \sum_i s_i$$

Here i and j denote lattice indices and the s_i's represent the spin value at the indicated location. The Ising model restricts the spins to 2 possible states, also called colours. The first sum only includes the nearest neighbour spins (here and in the following denoted by $\langle i, j \rangle$). J_{ij} is the interaction strength between spin s_i and s_j. The system is called ferromagnetic if $J_{ij} > 0$, because the same spin orientation for all spins is energetically favoured, and anti-ferromagnetic if $J_{ij} < 0$ respectively. The second sum includes the coupling to an external field. The number of spins is denoted by N. Notice that the discussion in this thesis is restricted to square lattices with periodic boundary conditions pbc.

The extension of the Ising model to an arbitrary number of possible single spin states, is named *Potts model*. For an extensive review of this model see Wu [29]. The generic Potts model Hamiltonian has the following form:

$$H_{\text{Potts}} = -\sum_{\langle i,j \rangle} J_{ij}\, \delta_{s_i,s_j} - h \sum_i \delta_{s_i,1}$$

Due to the Kronecker delta δ_{s_i,s_j} neighbouring spins with the same colour are the only contribution to the first sum. Throughout this work we assume a homogeneous ferromagnetic system so the couplings J_{ij} are positive ($J > 0$) and independent of the indices i, j. Further we apply no external field ($h = 0$) and

periodic boundary conditions. Therefore our Hamiltonian of interest yields

$$\boxed{H_{Potts} = -J \sum_{\langle i,j \rangle} \delta_{s_i,s_j}} \tag{2.1}$$

An important point, regarding the connection of the Ising and the Potts model needs to be outlined. Looking at the Potts model with 2 colours, which corresponds to the Ising model, one notices that the Hamiltonians do not yield the same value. This leads to the relation

$$s_i s_j = 2\delta_{s_i,s_j} - 1 \tag{2.2}$$

A further used representation of the same Hamiltonian (e.g. in [18]) is given by

$$H_{Potts} = -J \sum_{\langle i,j \rangle} \left(\delta_{s_i,s_j} - 1\right) \tag{2.3}$$

This includes an constant shift of the energy of $-2\,N\,J$ and will be relevant when we are comparing results of the different methods later on.

The probability for the system to be in a certain state $\mathbf{s}$ is given by the Boltzmann weight

$$p(\mathbf{s}) = \frac{e^{-\beta\, H(\mathbf{s})}}{Z}$$

where β denotes the inverse temperature and the normalisation Z is the partition function.

$$Z = \sum_{\mathbf{s}} e^{\beta\, H(\mathbf{s})} \tag{2.4}$$

The sum runs over all possible spin configurations. The partition function contains all thermodynamic information of the system at hand.

High and low temperature limit

In the limit of high temperature ($\beta = 0$) each spin in the system becomes independent of the state of his neighbouring spins. Therefore the partition function in

this limit equals the number of all possible configurations.

$$\lim_{\beta \to 0} Z = q^N \tag{2.5}$$

In the limit of low temperature ($\beta \to \infty$) all the spins freeze into the same state and the system effectively got just q remaining configurations.

$$\lim_{\beta \to \infty} Z = q \tag{2.6}$$

Phase transition temperature

The inverse temperature at the phase transition for an infinite square lattice is known analytically and can be found e.g. in [5].

$$J\beta_{crit} = \ln(1 + \sqrt{q}) \tag{2.7}$$

Therefore the physically interesting $J\beta$ range will, also for systems with a high number of colours, be restricted to quite low values. For example for a system with $q = 1000$ we obtain $J\beta_{crit} = 3.485$.
The phase transition is of first order for $q > 4$ and of higher order for $q <= 4$ (see [5]).

2.1 Fortuin-Kasteleyn random cluster model

Instead of writing the Hamiltonian of the Potts model as a function of the spins it is also thinkable to establish a dependence from a different variable, which is still able to describe the same states. One model, which will be used in chapter 6.6 to adapt the nested sampling algorithm, is the Fortuin-Kasteleyn random cluster (FK) model.

The FK model can be obtained starting from the partition function of the Potts model

$$\begin{aligned} Z(\mathbf{s}) &= \sum_{\mathbf{s}} e^{-\beta H(s)} = \sum_{\mathbf{s}} e^{\beta J \sum_{<i,j>} (\delta_{s_i,s_j} - 1)} \\ &= \sum_{\mathbf{s}} \prod_{<i,j>} e^{\beta J (\delta_{s_i,s_j} - 1)} \end{aligned} \tag{2.8}$$

Here the sum again includes all possible spin configurations ($s_1 = 1...q$, $s_2 = 1...q$, ..., $s_N = 1...q$).

In the next step a single factor of the product is rewritten as

$$\begin{aligned} e^{-\beta J} e^{\beta J\, \delta_{s_i,s_j}} &= e^{-\beta J} e^{0} (1 - \delta_{s_i,s_j}) + e^{-\beta J} e^{\beta J} \delta_{s_i,s_j} \\ &= (1-p) + p\, \delta_{s_i,s_j} \quad \text{with } p = 1 - e^{-\beta J} \end{aligned} \tag{2.9}$$

Now a new variable for bonds d_{ij} is introduced, which equals 1 with probability p if neighbouring spins are equal. If the neighbouring spins are not in the same state $d_{ij} = 0$. This leads us to

$$e^{-\beta J} e^{\beta J\, \delta_{s_i,s_j}} = \sum_{\{d_{ij}\}=0,1} \left[(1-p)\, \delta_{d_{ij},0} + p\, \delta_{s_i,s_j}\, \delta_{d_{ij},1} \right]$$

p describes the probability that two spins with the same colour are connected by a bond. Then the partition function is given by

$$\begin{aligned} Z(\mathbf{s},\mathbf{d}) &= \sum_{\mathbf{s}} \prod_{<i,j>} \sum_{\{d_{ij}\}=0,1} \left[(1-p)\, \delta_{d_{ij},0} + p\, \delta_{s_i,s_j}\, \delta_{d_{ij},1} \right] \\ &= \sum_{\mathbf{s}} \prod_{<i,j>} \sum_{\{d_{ij}\}=0,1} \left[(1-p)(\delta_{d_{ij},0} + \frac{p}{1-p}\, \delta_{s_i,s_j}\, \delta_{d_{ij},1}) \right] \end{aligned}$$

Here the state space has been extended through the embedding of the bond variable. This joint model, called *Fortuin-Kasteleyn-Swendsen-Wang model*, was introduced in [9]. As pointed out in [18] marginalisation of $Z(\mathbf{s},\mathbf{d})$ over $\mathbf{d}$ gives again the *Potts model* $Z(\mathbf{s})$. Marginalisation over $\mathbf{s}$ leads to the *Fortuin-Kasteleyn random cluster model* $Z(\mathbf{d})$.
Next we can exchange the summation and excerpt the factor $(1-p)^E$. Therefore we get

$$Z(\mathbf{s},\mathbf{d}) = (1-p)^E \sum_{\{d_{ij}\}=0,1} \sum_{\mathbf{s}} \prod_{<i,j>} \left[\delta_{d_{ij},0} + \frac{p}{1-p}\, \delta_{s_i,s_j}\, \delta_{d_{ij},1} \right]$$

If now $d_{ij} = 1$ the factor $\frac{p}{1-p}$ yields a contribution if δ_{s_i,s_j} is fulfilled. Hence it contributes $D = \sum_{ij} d_{ij}$ times. For $d_{ij} = 0$ only the factor 1 contributes without any restriction on the spin orientation.

The number of possible spin configurations is accounted in the factor q^C, which is considering that the spins in areas connected by $d_{ij} = 0$ are constant. C is the number of this connected areas, which are from now on termed clusters.

Following this considerations the Hamiltonian for the FK model can eventually be expressed as

$$\begin{aligned} Z(\mathbf{d}) &= \sum_{\{d_{ij}\}=0,1} (1-p)^E \left(\frac{p}{1-p}\right)^{D(\mathbf{d})} q^{C(\mathbf{d})} \\ &= \sum_{\{d_{ij}\}=0,1} p^{D(\mathbf{d})}(1-p)^{E-D(\mathbf{d})}\, q^{C(\mathbf{d})} \end{aligned} \tag{2.10}$$

with

$$\begin{aligned} E &= \text{number of edges} \\ D(\mathbf{d}) &= \sum_{i,j} d_{ij} = \text{number of bonds} \\ C(\mathbf{d}) &= \text{number of clusters} \end{aligned}$$

In this model a cluster describes a region, where the spins have the same colour and are connected through bonds. But because bonds are only active with probability p the area with equal spin colour does not correspond to the clusters (see figure 2.1). This model and especially the clusters are the basis for the algorithms described in chapter 3.2.2.

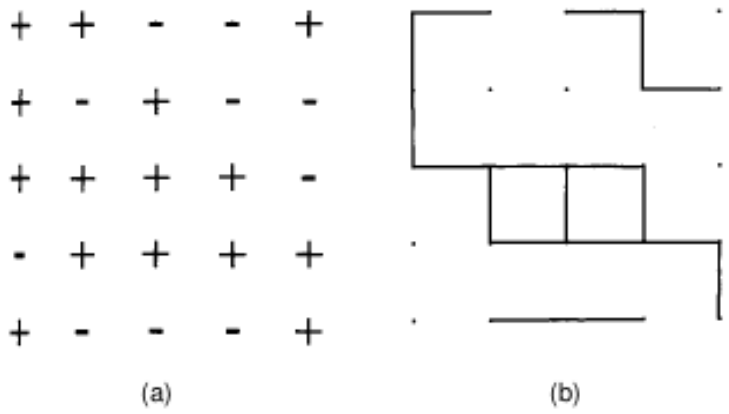


Figure 2.1: **(a)** Spin configuration of a $q = 2$ Potts model and **(b)** the respective representation in bond variables (taken from [14]). All spins within a cluster exhibit the same orientation.

3 Monte Carlo methods

In many problems in physics, expectation values or normalization constants usually need high dimensional integrals or sums to be evaluated. These integrals are mostly analytically intractable. Therefore either the integrand can be approximated by a function, which allows an analytical solution or an approximation for the integral as a whole can be computed by numerical integration. Or one uses sampling techniques like MC [27].
A good and compact overview of MC methods can be found in [27]. The presented basics here are based on this review. For a more detailed presentation of Monte Carlo simulations with the focus on statistical physics see [14].

The principle of a Monte Carlo simulation is to create a set of samples $\mathbf{x_i}$, which is distributed according to a certain distribution $p(\mathbf{x})$. The generation of such a sample is usually called a *draw* from the distribution. The set of drawn samples, of size K, allows to calculate an approximation of the expectation value of a certain quantity.

In general the expectation value of some function f is given by

$$\langle f(\mathbf{x}) \rangle = \int d\mathbf{x}\, f(\mathbf{x})\, p(\mathbf{x})$$

This can be estimated by

$$\boxed{\bar{f} = \frac{1}{K} \sum_{i=1}^{K} f(\mathbf{x}_i)} \tag{3.1}$$

In the limit $K \to \infty$ this unbiased estimator $\bar{f}$ approaches the expectation value $\langle f(\mathbf{x}) \rangle$. To obtain a meaningful estimate, it always needs to be accompanied by an estimate of its error.

The variance of $f(\mathbf{x})$ under the distribution $p(\mathbf{x})$ is defined as

$$\sigma^2 = \int d\mathbf{x}\, p(\mathbf{x}) \left[f(\mathbf{x}) \; - \langle f(\mathbf{x}) \rangle \right]^2$$

The estimator of the variance, employing the estimator of the mean, reads as follows

$$\boxed{var(\bar{f}) = \frac{\sigma^2}{K}} \tag{3.2}$$

This estimate of the variance is independent of the dimension of $\mathbf{x}$, which is not the case for e.g. quadrature methods. The $1/K$ dependence further gives the possibility to reduce the variance with increasing sample size. These two points are essential foundation of Monte Carlo algorithms.

One needs to be aware that the estimators are just valid in the case of independent draws of samples. Unfortunately the difficult part is to draw independent samples from a distribution $p(\mathbf{x})$. For standard distributions (Cauchy, student-t, Gauss) algorithms using a non-linear transformation of uniformly distributed random numbers are available. For non-standard distributions other algorithms like rejection sampling or importance sampling exist [27]. In high dimensions, though, all of this methods, which allow us to generate independent samples, become very inefficient.
For these problems Markov Chain Monte Carlo (MCMC) has first been developed in 1953 by Metropolis [27].

3.1 Markov Chain Monte Carlo

Instead of drawing independent samples, now successive draws are not independent any more, but they depend on the preceding sample. Therefore the samples in such a chain fulfil the following property

$$p(x_{t+1}|x_0, x_1, ..., x_t) = p(x_{t+1}|x_t)$$

This means a new state does just depend on the current one and not on the previous history. A series of samples evaluated in such a manner is called a *Markov chain*. It is specified by the initial probability $p(x_0)$ and the transition probability $T(x, x')$ from the state x at time t_i to the state x' at t_{i+1}.

An invariant distribution of a Markov chain is a distribution which will not change any more once it is reached.
The Markov chain is named irreducible if it is able to reach all of its possible states. Further it is called aperiodic if it is not captured in periodic cycles. If the chain is irreducible and aperiodic it is called ergodic.
In Markov chain Monte Carlo (MCMC) algorithms we use ergodic chains to sample from their invariant distribution.
For a Markov chain to be reversible it needs to fulfil the so called *detailed balance* condition. The detailed balance condition

$$T(x, x')p(x) = T(x', x)p(x')$$

ensures that the distribution is invariant.
In the next section we will see how we can construct a Markov chain with a desired invariant distribution.

3.1.1 Metropolis Hastings algorithm

The Metropolis Hastings (MH) algorithm accepts a new state of the chain with the acceptance probability

$$P_{accept} = min\left(1, \frac{p(x')\, p_p(x_t|x')}{p(x_t)\, p_p(x'|x_t)}\right)$$

where p_p denotes the proposal function and x' the proposed state. If this function is symmetric $p_p(x'|x) = p_p(x|x')$ then one gets the original Metropolis algorithm. If the proposed step gets accepted then $x_{t+1} = x'$, else the actual state is kept $x_{t+1} = x_t$.
The proposal function has crucial influence on the acceptance rate and therefore on the efficiency of the method. A narrow function will suggest new positions in parameter space, close to the current one. The new position will be accepted with a high probability, but the samples of the chain will take a long time to reach all of the phase space. Using a broad proposal function give us the vice versa situation, where the low acceptance rate will be the reason for inefficiency. To find the optimal setting one can use the autocorrelation function.

Autocorrelation

The correlation between successive states can be computed by the autocorrelation function of the Markov chain [27]. It is defined as

$$\rho(j) = \frac{\sum_t (x_t - \langle x \rangle)(x_{t+j} - \langle x \rangle)}{\sqrt{\sum_t (x_t - \langle x \rangle)^2}\sqrt{\sum_t (x_{t+j} - \langle x \rangle)^2}} \tag{3.3}$$

From the autocorrelation function we are able to calculate the integrated correlation time

$$\tau_{int} = \frac{1}{2} + \sum_{j=1}^{n} \rho(j)\,(1 - \frac{j}{n}) \tag{3.4}$$

The integrated correlation time specifies the time, that is needed between two measurements in a timeseries to be uncorrelated. Therefore τ_{int} can be used to obtain the true variance from the naive variance given in equation 3.2.

$$var(\bar{f}) = \frac{\sigma^2}{K}\, 2\tau_{int} \tag{3.5}$$

This further implies that the number of effective measurements scales like

$$n_{uncorr} = \frac{n}{2\,\tau_{int}} \tag{3.6}$$

MH for Potts models

For the Potts model a potential choice for the acceptance probability, assuming symmetric proposal, looks like

$$\begin{aligned} P_{accept} &= \min\left(1, \frac{\exp\left[-H(\mathbf{s}')\right]}{\exp\left[-H(\mathbf{s})\right]}\right) \\ &= \min\left(1, \exp\left[-\Delta H\right]\right) \end{aligned}$$

A possible implementation of MH uses local spin updates. After an initialisation of the system, a randomly chosen single spin is flipped and the energy difference is computed. The flip gets accepted with the above probability. After each step an observable of the system can be measured. For correct measures one needs to take into account a certain time of thermalisation in the beginning of the simulation. Because of the tiny changes in configuration due to a single spin flip correlations

will play an important role.
Especially in MHMC simulations of Potts/Ising models near the phase transition the diverging correlation time causes problems. This behaviour is known as *critical slowing down*. The slowing down stems from the diverging correlation length ξ around that critical point. Large areas show the same colouring and therefore a single spin flip is rarely accepted. When dealing with finite systems, the correlation length can then be approximated by the system size L. For the finite two dimensional Ising model with local updates it is known that the correlation time grows as

$$\tau \propto \xi^z \approx L^z \tag{3.7}$$

where the critical exponent $z \approx 2$. (See [13] for the basic application of MC on spin systems)

3.2 Cluster algorithms

Now we consider how to improve the Markov process regarding to critical slowing down. Once the Markov-chain has reached a representative spin configuration, moves by single spin flips will definitively take a long time to be uncorrelated. They will not sufficiently change the state of the spin–system. We expect that reasonable MC–moves are rather those which 'rotate' large portions of ferromagnetic domains at once. Therefore our strategy is that all Potts spins in one domain acquire a new value $s \in \{1, 2, \ldots, q\}$. To do so we can now employ the clusters we found in section 2.1 for the FK model.

3.2.1 Introduction of Clusters

We want to examine the clusters in more detail. What we have to do is to glue ferro-magnetically aligned spins together to form rigid *clusters* which have to be rotated simultaneously in the subsequent Markov step. Of course, rotating all spins of ferromagnetic domain together would not be a valid procedure if those domains would never split apart. We again need a *stochastic ergodic process* describing the formation of clusters in a given spin configuration $\mathbf{s}$. This process must meet the following conditions:

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

Then clusters are described by a set of bonds d_{ij} between spins s_i and s_j. We use the convention $d_{ij} = 1$ if a bond links s_i with s_j and $d_{ij} = 0$ if this is not the case. Let N_d denote the number of actual bonds and $N_{\overline{d}}$ the number of vacant bonds. Then their sum must be the number of next neighbours N^{nn},

$$N_d + N_{\overline{d}} = N^{nn}$$

If spin s_1 is linked with s_2 by d_{12} and spin s_2 is liked with s_3 by d_{23} then all three belong to the same (perhaps even bigger) cluster. All N_c clusters can be identified on the basis of the binary bonds d_{ij}.

3.2.2 The Swendsen–Wang algorithm

A way to utilize these clusters and hence overcome critical slowing down has been developed by Swendsen and Wang (SW) [25]. The SW algorithm consists of four steps:

1. For each interaction-pair $< i, j >$ with equal spins $s_i = s_j$ a bond is inserted with probability p
2. All spins which are pairwise connected by bonds are frozen and form clusters. An isolated spin (not connected to any bond) is also called a cluster. The clusters are identified and enumerated $1, 2, \ldots N_c$. They constitute the objects to be manipulated by MC-moves.
3. In step three new spin values are generated for whole clusters. There are two commonly used approaches: single-cluster (Wolff method) or multi-cluster (actual SW) move [14]. In the **single-cluster algorithm** one cluster is selected at random and for all spins in the selected cluster a new common spin value is determined at random; the old value is excluded. In the Ising case $q = 2$ the cluster spins are flipped with probability 1. I.e. the spin-change proposal is always accepted. In the **multi-cluster algorithm** the

clusters are scanned sequentially and for each cluster a new spin value $s \in \{1, 2, \ldots, q\}$ is chosen at random. In this case there is a finite probability $1/q$ that the spins in a cluster are not changed. In the Ising case the spins in a cluster are flipped with probability $1/2$.

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

The key success factor of the SW-algorithm is that the bond creation is driven by the Hamiltonian itself. This leads to a proposal spin-configuration which will *always* be accepted. As opposed to the naive scheme where large modifications are proposed by an *non-physical* proposal distribution which then are decided upon by the Metropolis–Hastings acceptance probability and which lead to tiny acceptance rates.
The critical exponent for the SW algorithm is therefore way lower than for the single spin updates - $z \approx 0.25$.

3.2.3 Cluster Identification

One of the critical and most time consuming step in cluster algorithms is the identification of a cluster. A quite commonly used algorithm is the Hoshen Kopelmann method [14]. A further fast algorithm has been presented by Newman et al. [19].

3.3 Evaluation of the partition function

The standard MC algorithms are designed for the evaluation of expectation values of observables. This evaluation does not require the knowledge of the partition function. Computing the partition function Z itself, involves usually a much more structured integrand. This leads to severe problems for standard MC algorithms. Also when computing similar quantities like the evidence (also marginal likelihood) in Bayesian inference, standard MC methods encounter massive problems (for details see [27]). Hence for the evaluation of these quantities special algorithms, like thermodynamic integration (see chapter 4) or nested sampling (see chapter 6), are available .

4 Thermodynamic integration

A way to compute the partition function Z is depicted by the so called thermodynamic integration (TI), where an auxiliary parameter β is introduced. Consider the normalization constant in the form

$$Z(\beta) = \int d\mathbf{x}\, L(\mathbf{d}|\mathbf{x})^{\beta}\, p(\mathbf{x}) \tag{4.1}$$

where $Z(0) = 1$, because of the normalized prior and $Z(1) = \int d\mathbf{x}\, L(\mathbf{d}|\mathbf{x})\, p(\mathbf{x})$. The derivation of $\ln(Z(\beta))$ with respect to β leads to

$$\begin{aligned}
\frac{d}{d\beta}\, \ln(Z(\beta)) &= \frac{1}{Z(\beta)} \int d\mathbf{x}\, p(\mathbf{x})\, \frac{d}{d\beta} \underbrace{\left(L(\mathbf{d}|\mathbf{x})^{\beta}\right)}_{\exp(\beta\, \ln(L))} \\
&= \int d\mathbf{x}\, \underbrace{\frac{p(\mathbf{x})}{Z(\beta)} L(\mathbf{d}|\mathbf{x})^{\beta}}_{p_{\beta}(\mathbf{x})}\, \ln(L(\mathbf{d}|\mathbf{x})) \\
&= \langle \ln(L(\mathbf{d}|\mathbf{x})) \rangle_{\beta}
\end{aligned}$$

Here $\langle \ln(L(\mathbf{d}|\mathbf{x})) \rangle_{\beta}$ denotes the expectation value of $\ln(L(\mathbf{d}|\mathbf{x}))$ under the distribution $p_{\beta}(\mathbf{x})$. We are therefore able to retrieve $ln(Z)$ via a integration over β. Evaluating this expectation value by a SW simulation one can compute the logarithm of the partition function using

$$\ln(Z(1)) - ln(Z(0)) = \int_0^1 d\beta\, \langle \ln(L(\mathbf{d}|\mathbf{x})\,) \rangle_{\beta}$$

An approximation of this integral can be obtained by the sum over a sequence of inverse temperatures β_i.

$$\boxed{\ln(Z) \approx \sum_{i=1}^{N} \Delta\beta\, \langle \ln(L(\mathbf{d}|\mathbf{x})\,) \rangle_{\beta_i}} \tag{4.2}$$

where $\Delta\beta = \beta_i - \beta_{i-1}$ with the start value $\beta_0 = 0$.

4.1 Thermodynamic integration for Potts models

According to equation 4.2, we need the expectation value $\langle \ln(L(\mathbf{d}|\mathbf{x})) \rangle_{\beta_i}$ for the calculation of an estimate for $ln(Z)$ with thermodynamic integration. Therefore we compare equation 4.1, from which we derived this estimate, with the partition function given in equation 2.4. We find

$$L(\mathbf{d}|\mathbf{x})^{\beta} = \left(\mathrm{e}^{H(\mathbf{s})}\right)^{\beta} \tag{4.3}$$

Using the Hamiltonian from equation 2.1, we can retrieve

$$L(\mathbf{d}|\mathbf{x})^{\beta} = \left(\mathrm{e}^{J\,S}\right)^{\beta} \tag{4.4}$$

$$\Rightarrow \quad \ln(L(\mathbf{d}|\mathbf{x})) = J\,S \tag{4.5}$$

S denotes the number of neighbouring spins in the same state. We therefore need to compute the expectation value for S at certain β_i values. With equation 4.2 the estimate for $ln(Z)$ is then given by

$$\ln(Z) = \sum_{i=1}^{M} S_{\beta_i}\,\Delta\beta \tag{4.6}$$

The evaluation of $\langle S_{\beta_i} \rangle$ at certain β_i values is performed as follows. First the range of $\beta_0 = 0$ to $\beta_{max} = \beta_{N_\beta}$ is divided into N_β equally spaced steps. At β_0, which corresponds to $T = \infty$, the partition function is fixed at the known value $\ln(Z) = N \ln(q)$ (see section 2). Then S is sampled via Swendsen Wang a 100 times at temperature β_1. From these samples we can retrieve $\langle S_{\beta_i} \rangle$. The evaluations are iterated until β_{max} is reached. For each SW time-series the configurations after the first couple of updates (here 5) are discarded due to the above mentioned thermalisation of the sequence. We introduce the variable E, which depicts the maximal number of bonds $2N$. The $J\beta$ dependence of $(E - \langle S_{\beta_i} \rangle)$ is computed that way for a 3×3 and $q = 5$ system for illustration. The result is displayed in figure 4.2. Indicated are plots for different number of β_i values. The plots now show the same progression for all number of steps, though smaller differences in the β values lead to less fluctuations.

Murray et al. employed less optimal parameters for evaluating the results in [18] via a similar approach called annealed importance sampling. They used a single SW update for each β_i and computed this way 100 sequences of $\langle S_{\beta_i} \rangle$ values. For the evaluation of the estimate in equation 4.6 they employed the average of this sequences. Figure 4.2 shows the result of this approach. This way of computing the expectation values is bound to fail, because the configurations after just one SW update show correlations. Further there is no possibility for thermalisation. This becomes evident when looking at figure 4.2. The plot for $N_\beta = 100$ steps shows a strongly correlated behaviour and differs in his progression clearly from the other plots with $N_\beta = 500, 1000$. For $N_\beta = 500$ and $N_\beta = 1000$ the correlations are oppressed more, because the small changes in temperature will not cause a big change in the configuration.

The results for the partition function for this system are displayed in figure 4.3. At 500 β steps an increasing number of SW updates have been performed. With the increasing number of SW updates $M = 100, M = 500$ and $M = 1000$, the result of the thermodynamic integration converges to the exact result (calculated in section 6.6.2).

Using multiple SW updates for evaluating the estimates at each β_i value does not cause a total failure of the thermodynamic integration approach, as described in [18], where only one SW updates for each β_i value has been used. For further analysis the correlation time of thermodynamic integration is investigated in more detail.

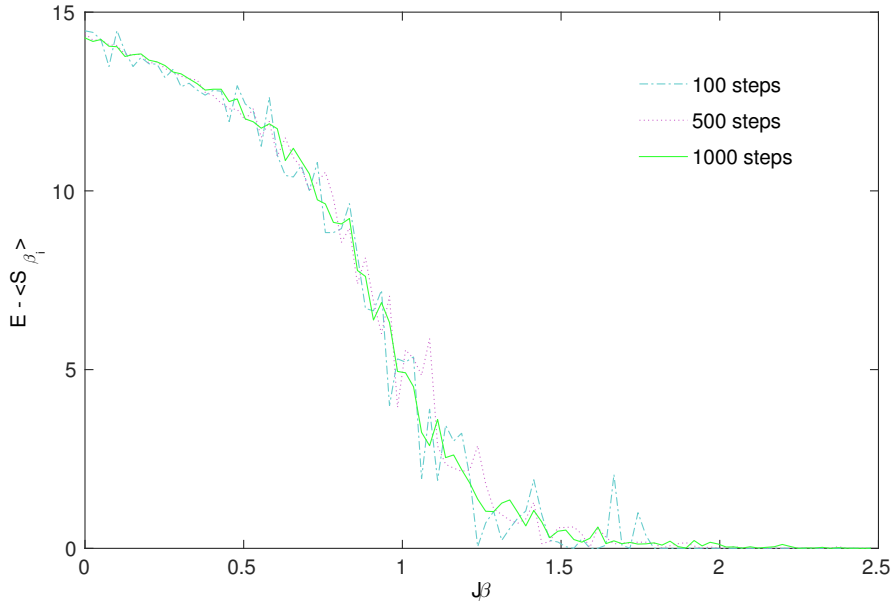


Figure 4.1: The difference $E - \langle S_{\beta_i} \rangle$ in dependence of $J\beta$ for a 3×3, $q = 5$ system and for different number of β steps. At each $J\beta$ value 100 SW updates are performed.

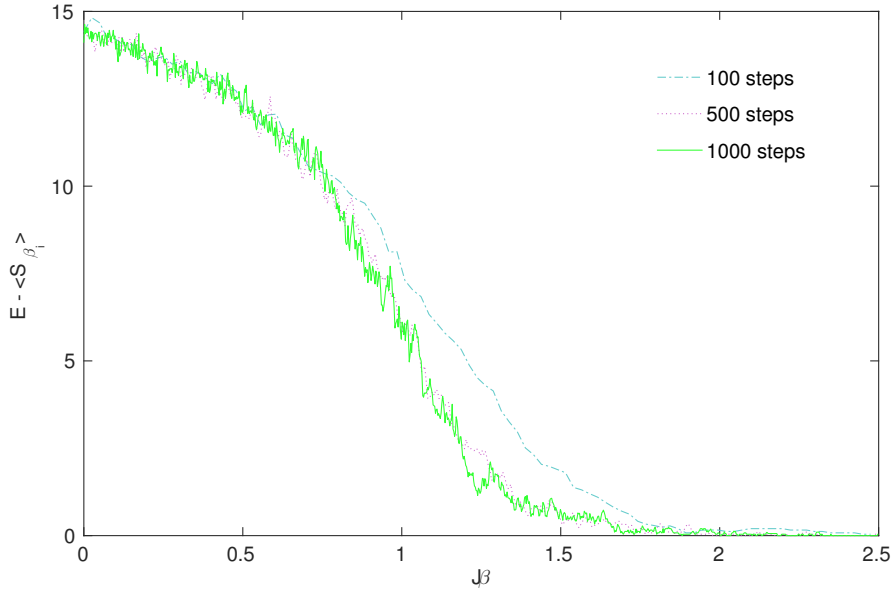
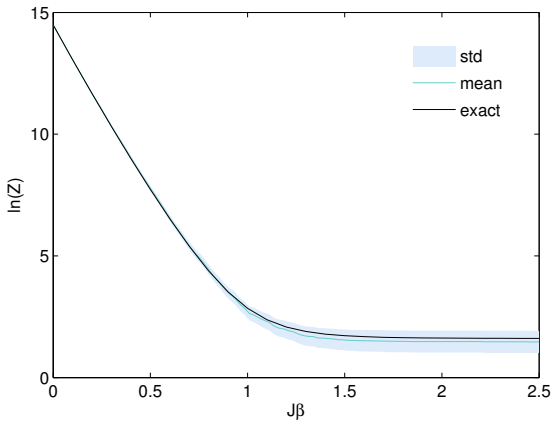
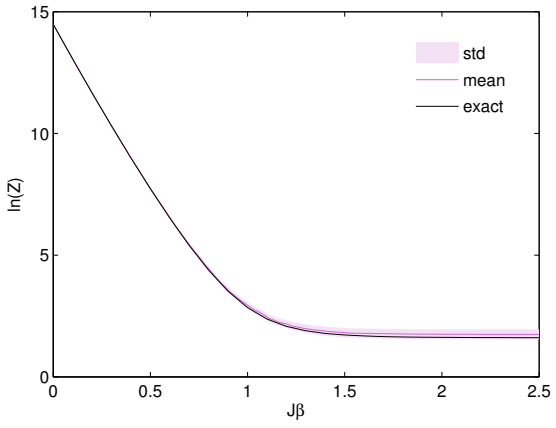


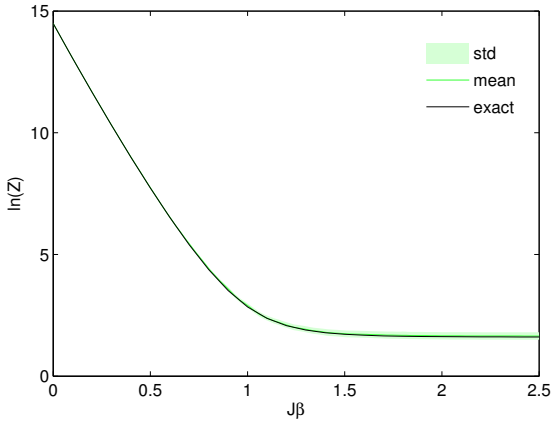
Figure 4.2: The difference $E - \langle S_{\beta_i} \rangle$ in dependence of $J\beta$ for a 3×3, $q = 5$ system and for different number of β steps averaged over 100 runs.



(a)



(b)



(c)

Figure 4.3: The logarithm of the partition function $\ln(Z)$ for a 3×3, $q = 5$ Potts system evaluated via TI for different number of SW updates M: (a) $M = 100$, (b) $M = 500$ and (c) $M = 1000$. The exact result is depicted by the black line.

4.1.1 Correlation time analysis

For later comparison of nested sampling with the performance of thermodynamic integration, we analyse the correlation time of the SW time series, required by TI at each β_i.

The Swendsen Wang algorithm used for thermodynamic integration, has to deal with increasing correlation times near phase transitions for Potts models with $q > 4$ (see [10]).

In figure 4.4 one can see a distinct peak in correlation time near the phase transition temperature. Here a 8×8 system with $q = 10$ has been analysed. At each β step 10000 SW updates were computed. Further $J\beta = ln(2)$[1] is indicated in the figure by a dotted black line. The correlation times for SW runs at $J\beta$ values below that line all show a value around 2 (see figure 4.5). Therefore the correlation causes no complication, when computing values of $ln(Z)$ up to this temperature.

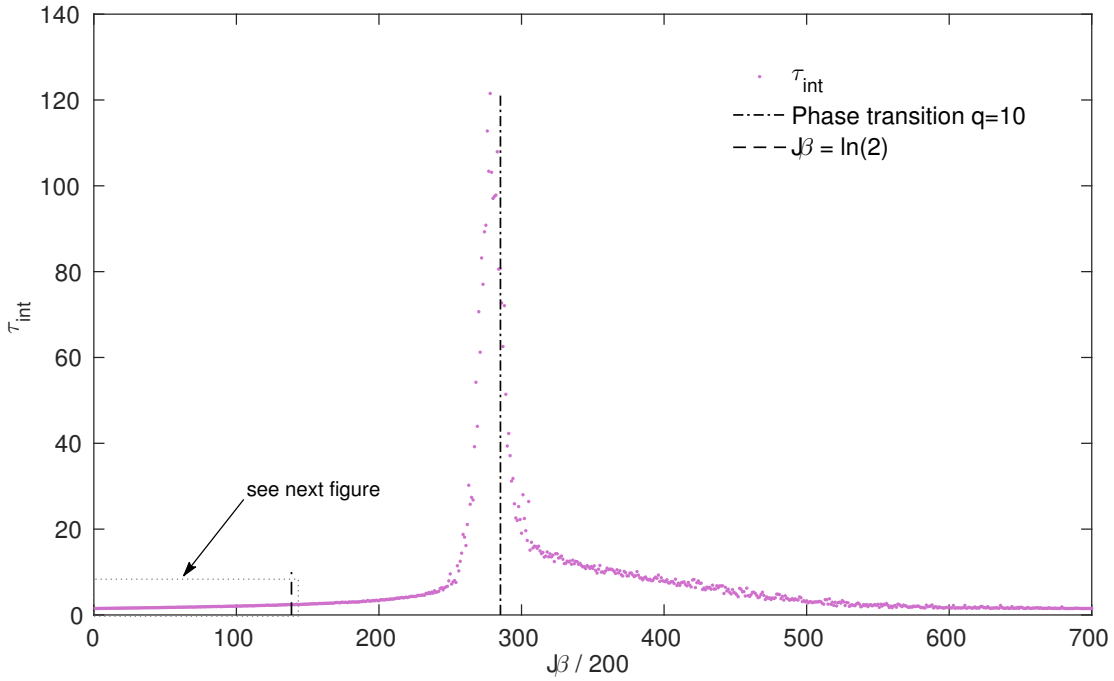


Figure 4.4: Correlation times for a 8×8, $q = 10$ system computed for temperatures around the phase transition. $J\beta = ln(2)$ and the exact phase transition are indicated with vertical lines.

[1] TI up to this temperature is exploited later to compute the prior normalization required for the nested sampling algorithm for Potts models.

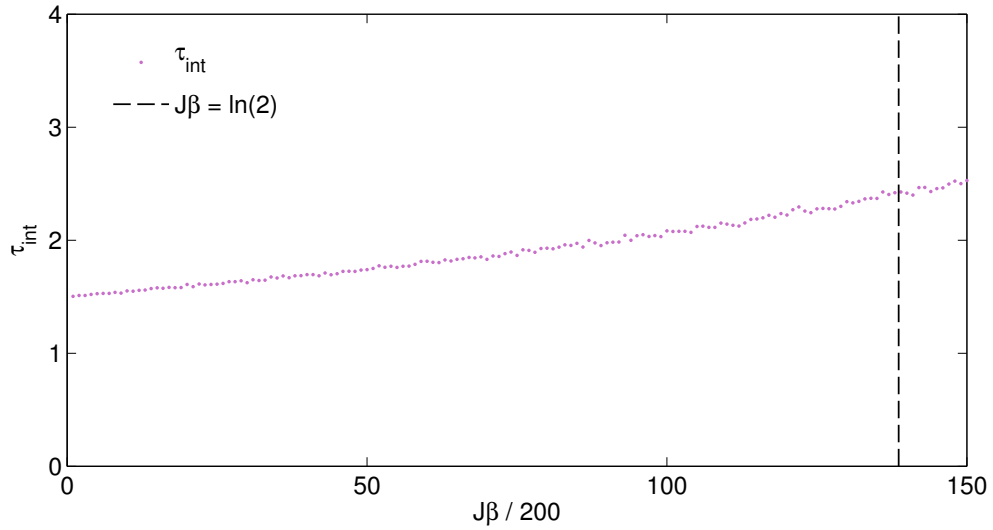


Figure 4.5: Zoom into the area of indicated in the previous figure. The correlations in this $J\beta$ range have a value around 2 for the 8×8 and $q = 10$ system.

However if one wants to assess the partition function for $J\beta$ above the phase transition, the correlation needs to be taken into account with respectively large number of SW updates or a proper calculation of the errors (see 3.1.1).
Baillie et al. [3] evaluated the correlation time at the critical temperature for the $q = 2$ and $q = 3$ Potts system. For the analysis of this critical slowing down, of the $q = 10$ Potts system, we determined the integrated correlation time τ_{int} for different grid sizes, as described in [3]. For a first test we perform the calculation for q=3 systems. We computed at each $J\beta$ value a time series of 5000 SW updates. For each of this time series we calculated the mean autocorrelation function $\rho(t)$ out of 10 independent runs. To get rid of noise in the data, each $\rho(t)$ has been cut off, before either an increase or an negative value in $\rho(t)$ occurred.
Afterwards a linear fit through the last 50 percent of the logarithm of the autocorrelation function $ln(\rho(t))$ was determined. This fit is further used to approximate $\rho(t)$ for values above the cut-off value. Finally the correlation time τ_{int} is computed by summing $\rho(t)$ up to the cut-off value and then adding the values from the linear fit, which can be expressed analytically by a geometric sum. For an estimate of the error, τ_{int} has also been separately calculated for each of the above mentioned 10 runs.

The results in [3] for the $q = 3$ systems, were compared to our results in order to get an verification (see table 4.1). Our estimates yield the same values within the

error bars. Our errors exhibit larger values, because Baillie et al. retrieved τ_{int} from runs with $5 \times 10^5 - 10^6$ sweeps, which is about a factor of 100 larger than in our measurements.

Table 4.1: Comparison of τ_{int} at the infinite critical point for $q = 3$ Potts systems. For further discussion see text

Grid-size	τ_{int} (Baillie et al.)	τ_{int}
8×8	6.056 $\pm$0.009	6.59 $\pm$0.49
16×16	8.99 $\pm$0.03	9.9 $\pm$1.9
32×32	13.30 $\pm$0.06	12.9 $\pm$1.7
64×64	19.58 $\pm$0.12	16.9 $\pm$3.7

After verifying the way of evaluation of τ_{int}, we focus again on the q=10 Potts model. The integrated correlation times for this system are listed in table 4.2. The number of SW updates for each grid-size has been set as follows: 10000 for the 4×4, 20000 for the 8×8, 100000 for the 16×16 and 250000 for the 32×32 system.

Figure 4.6 shows the power law function $\tau_{int} = a\, L^{z_{int}}$ fitted to the data in the table.

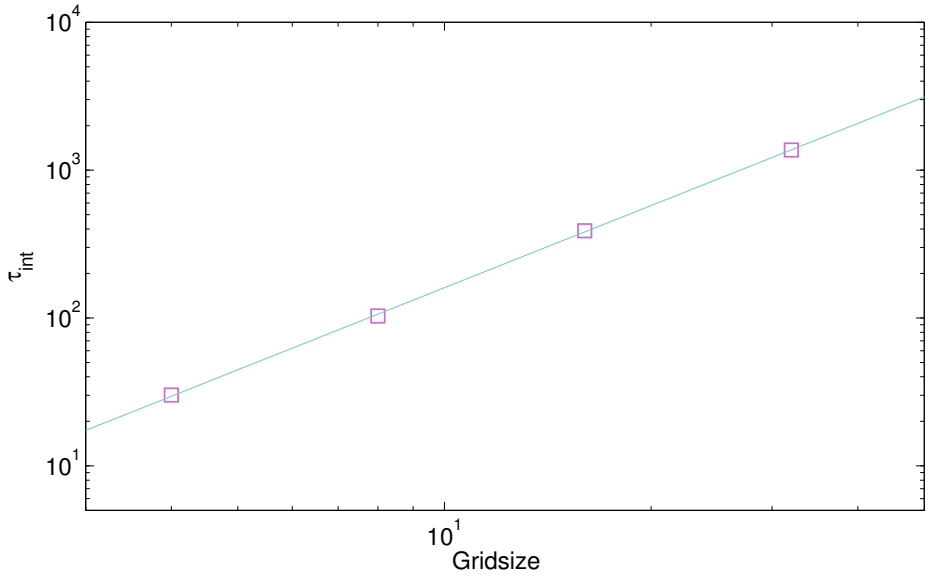


Figure 4.6: τ_{int} results for a $q = 10$ Potts model evaluated for different grid-sizes and fit to a power law. Note the double logarithmic scale.

The scaling exponent z_{int} from this fit yields

$$z_{int} \quad = 1.85 \pm 0.07 \tag{4.7}$$

For the pre-factor $a = 2.29 \pm 0.52$ is obtained. The fit has been computed with Matlab's curve fitting tool.

Table 4.2: τ_{int} at the infinite critical point for $q = 10$ Potts systems.

Grid-size	τ_{int}	
4x4	30.1 $\pm$	3.7
8x8	103.3 $\pm$	14.6
16x16	389.4 $\pm$	94.8
32x32	1369.8 $\pm$	353.3

With the fitted power law model at hand we can now compute the proper scaling of the length of the SW time series at the critical temperature as well as correct the respective variance according to equation 3.5.

5 Multicanonical sampling

Multicanonical Sampling (MUCA) is a method introduced by Berg and Neuhaus [4]. It is designed to overcome the problems of classical Monte Carlo methods in a highly varying phase space with multiple separated maxima. In such cases the classical methods could easily get stuck in a separated peak and therefore correct sampling is difficult. Still transitions between the peaks are possible but very improbable and therefore the relative weights of the maxima will be not determined correctly.
For example the magnetization of the Ising model below the critical temperature is most probable either positive or negative. This is displayed in figure 5.1 (taken from [21]). Here the canonical distribution $p_{can}(M)$ drawn in red shows two well separated maxima.

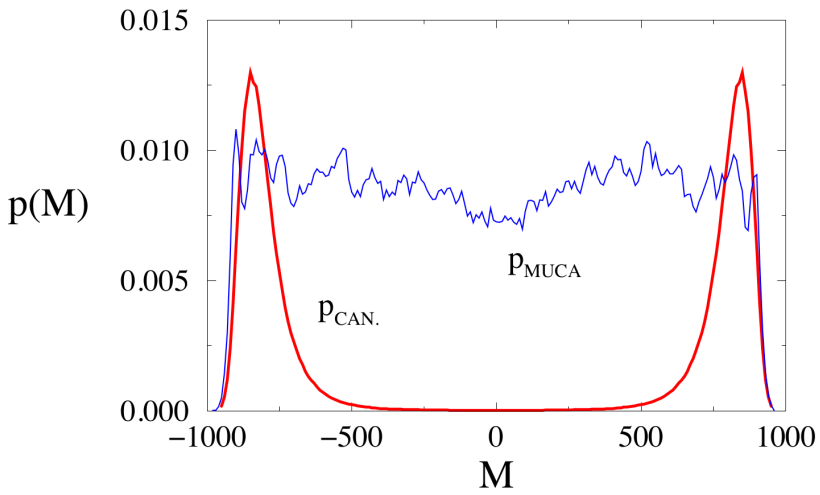


Figure 5.1: Comparison of the multicanonical and the canonical distribution

In principle the distribution can be a function of any order parameter of the system [21]. As we want to obtain the partition function later on, we focus on the investigation of the probability distribution as a function of the energy.

The principal idea of multicanonical sampling is to tune the weights of the Monte Carlo sampling in such a way that the whole phase space can be reached (see [21]). The Boltzmann weight

$$\omega_{can}(E) \quad \propto \quad \exp(-\beta E)$$

leads to the following canonical probability distribution for the energy

$$p_{can}(E) \quad = \quad \frac{1}{Z_{can}} n(E) \exp(-\beta E) \tag{5.1}$$

with $n(E)$ being the spectral density of the energy.
To determine the multicanonical probability distribution an additional weight-function $\omega_{muca}(E)$ is introduced (the resulting multicanonical distribution for the above example is indicated by the blue line in figure 5.1).

$$p_{muca}(E) \quad \propto \quad p_{can}(E)\, \omega_{muca}(E)$$

The canonical probability distribution can be retrieved again by re-weighting

$$p_{can}(E) \quad \propto \quad \frac{p_{muca}(E)}{\omega_{muca}(E)}$$

A MC simulation, with the modified weights, enables us to evaluate expectation values for observables under the multicanonical distribution. Canonical expectation values again are retrieved via re-weighting.
The factor $\omega_{muca}(E)$ is not known in the first hand, otherwise we would already know p_{can}, therefore we perform a canonical simulation in the beginning to retrieve a rough estimate for it Find more details on the estimation in [5], and for the estimation related to Potts models in [21].

5.1 Weight estimate

The weight function can for instance be estimated by means of an iterative process, as described in [21]. In the beginning the order parameter E is divided into bins E_i, which cover all possible values of E.

We define the weight-function as $\omega_{muca}(E_i) = \exp(W(E_i))$ and determine $W(E_i)$ by means of the following algorithm.

Algorithm 5.1.1: WEIGHT ITERATION($W(E_i)$)

initialize $W(E_i) = 0 \;\; \forall\, i,\; C = 1,\; C_{\text{threshold}}$
while $C > C_{\text{threshold}}$
do
- **while** not every bin visited
  - **do**
    - MC simulation with the modified probability
    - every time visiting bin i set:
    - $W(E_i) \leftarrow W(E_i) - C$
- $C \leftarrow C/2$

return $(W(E_i))$

During the iteration in the presented algorithm, the value of the function $W(E_i)$ of frequently visited bins is lowered to a higher extent than in bins, where visits happen rarely. Therefore the probability of visiting it again is reduced. After all bins have bin reached during the Monte Carlo runs, C is reduced. The process eventually comes to halt, when a certain threshold value of C is reached.
With the weight function at hand we sample instead of the canonical distribution p_{can} from the multicanonical distribution

$$p_{muca}(E) \;\; = \;\; \frac{1}{Z_{muca}} n(E)\, w_{muca}(E) \tag{5.2}$$

via Monte Carlo.

5.2 Evaluation of the partition function

It is furthermore possible to retrieve the partition function from the results of a multicanonical simulation. An outline of this calculation can be found in [4]. Substituting $n(E)$ from equation 5.2 in equation 5.1, the canonical probability

distribution can be estimated after a multicanonical simulation by

$$p_{can}(E) = \frac{Z_{muca}}{Z(\beta)} \frac{H_{muca}(E)}{w_{muca}(E)} \exp(-\beta E)$$

Here the histogram generated by the multicanonical simulation $H_{muca}(E)$, serves as estimate for $p_{muca}(E)$.
The partition function $Z(\beta)$ is the quantity we want to retrieve. For $\beta = 0$ one finds $Z(0) = q^N$, because at infinite temperature the system is in total disorder. Therefore the unknown normalization Z_{muca} of the multicanonical distribution can be obtained at this temperature from the normalization of the canonical distribution.

$$\begin{aligned} \sum_E p_{can}(E) &= \sum_E \frac{Z_{muca}}{Z(\beta)} \frac{H_{muca}(E)}{w_{muca}(E)} \exp(-\beta E) = 1 \\ \text{at } \beta = 0 \quad &\rightarrow \quad \sum_E \frac{Z_{muca}}{q^N} \frac{H_{muca}(E)}{w_{muca}(E)} = 1 \\ &\rightarrow \quad Z_{muca} = q^N \frac{1}{\sum_E \frac{H_{muca}(E)}{w_{muca}(E)}} \end{aligned}$$

With Z_{muca} at hand, it is possible to compute $Z(\beta)$ for $\beta > 0$ again by means of the normalization $\sum_E p(E) = 1$.

$$\boxed{Z(\beta) = Z_{muca} \sum_E \frac{H_{muca}(E)}{w_{muca}(E)} \exp(-\beta E)} \tag{5.3}$$

6 Nested sampling

6.1 Basics

Nested sampling (NESA) was proposed by John Skilling in 2004 [22]. The basics presented here are based on the chapter nested sampling in [26] and the paper from Murray et al. [18]. For a more detailed explanation and analysis of the method the reader is referred to the cited book.

Again the underlying problem is the computation of an integral of the form

$$Z = \int d^R\mathbf{x}\, L(\mathbf{x})\, p(\mathbf{x}) \tag{6.1}$$

where the integration is done in any arbitrary number of dimensions R. Here $L(\mathbf{x})$ denotes the likelihood and $p(\mathbf{x})$ the prior. The following considerations hold also when dealing with a high dimensional sum $Z = \sum_i \delta(\mathbf{x} - \mathbf{x}_i)\, p(\mathbf{x}_i)\, L(\mathbf{x}_i)$.

Starting from an integral as in equation 6.1, Skilling suggested to rewrite it as

$$Z = \int d\lambda\, X(\lambda) \tag{6.2}$$

$$X(\lambda) := \int d^R\mathbf{x}\, p(\mathbf{x})\, \Theta(L(\mathbf{x}) > \lambda) \tag{6.3}$$

The prior mass $X(\lambda)$ cumulates the prior, which is subject to a constraint on the likelihood, over the parameter space $\mathbf{x}$. The Heaviside function Θ ensures the integral is just non-zero in areas, where the likelihood $L(\mathbf{x})$ exceeds a certain threshold λ.
To verify expression 6.2 insert equation 6.3.

$$\int d\lambda\, X(\lambda) = \int d^R\mathbf{x}\, p(\mathbf{x}) \int d\lambda\, \Theta(L(\mathbf{x}) > \lambda) = Z$$

Figure 6.1, taken from [26], shows the principle of the nested sampling algorithm in a one dimensional parameter space with a discrete likelihood function.
In the first two columns plots of the likelihood $L(\mathbf{x})$ and the prior $p(\mathbf{x})$ are shown. The thresholds λ_i are indicated in the likelihood plots by horizontal lines. Areas of the histogram fulfilling the constraint $L(\mathbf{x}) > \lambda$ are displayed with a greyish shade. With increasing values of the threshold (λ_1 to λ_4) the prior masses $X(\lambda)$ decrease. The rightmost diagram shows the dependence of the likelihood $L(X)$ on the prior mass X.

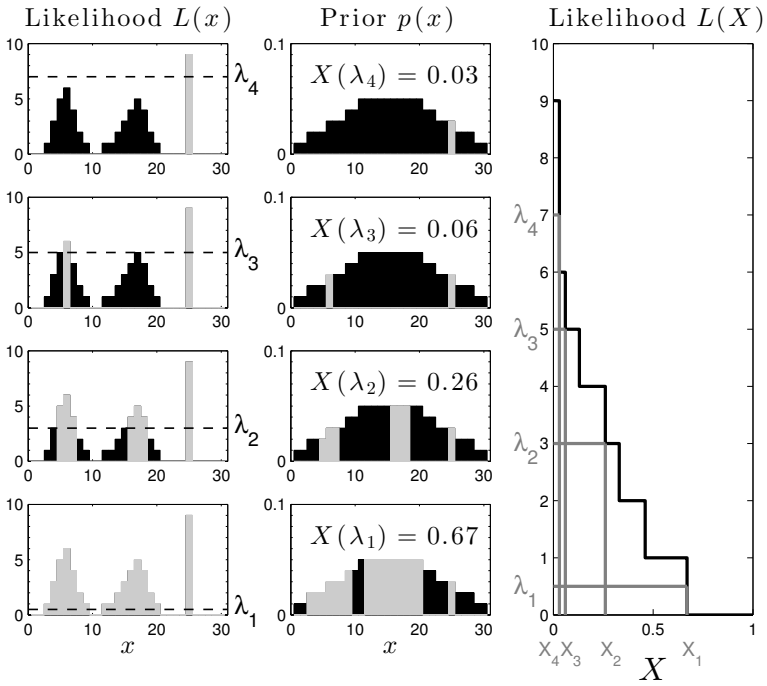


Figure 6.1: Principle of the nested sampling algorithm. The first two columns show plots of the likelihood $L(x)$ and the prior $p(x)$ with different threshold values λ_n. The last column shows the likelihood in dependence of the prior mass X

The value of the prior mass $X(\lambda)$ is constrained to values in the range from 0 to 1. The maximum value $X = 1$ is obviously obtained with the constraint $\lambda = 0$, because integration of the normalized prior $p(\mathbf{x})$ runs over the whole parameter space. Equation 6.3 ensures further, that with increasing λ the integrand decreases monotonically.
Strict monotonicity can be achieved by adding an infinitesimal value $\kappa(\lambda)$ to the prior mass demanding finally $\kappa(\lambda) \to 0$. We allow the likelihood L to be equiva-

lently a function of $\mathbf{x}$ and X, depending on the present context. Integrating the likelihood over the prior mass leads to the form

$$Z = \int_0^1 dX\, L(X) \tag{6.4}$$

So eventually the preceding rearrangements simplify our multidimensional integral in equation 6.1 to an one dimensional one, where the integrand is monotonically decreasing and the integral is bound by 0 and 1.

6.2 Nested sampling algorithm

The integral in equation 6.4 can be approximated through a Riemann sum

$$Z_K = \sum_{n=1}^{\infty} L(X_n)\Delta X_n \tag{6.5}$$

$$\Delta X_n = X_n - X_{n+1} \tag{6.6}$$

The monotonicity of $L(X)$ ensures that the sum yields a lower bound for the integral value. Equivalent the upper bound reads as

$$Z_K = \sum_{n=1}^{\infty} L(X_n)\,(X_{n-1} - X_n) \tag{6.7}$$

Skilling came up with the following algorithm to compute the $L(X_n)$ values.

Algorithm 6.2.1: NESTED SAMPLING ALGORITHM($\hat{\lambda}_n$)

initialize $\hat{\lambda}_0 = 0$, $n = 0$,
while $\lambda_{n+1} - \lambda_n > \epsilon_\lambda$
do
- increment the iteration count $n \leftarrow n + 1$
- draw sample $\{x_i\}$ of size K from $p(\mathbf{x}|\hat{\lambda}_{n-1})$
- calculate the likelihood for each sample $\lambda_i = L(x_i)$
- determine sample minimum $\hat{\lambda}_n = min_i(\lambda_i)$

set $n_{max} = n$
return $(\hat{\lambda}_n, n_{max})$

Here the prior

$$p(\mathbf{x}|\hat{\lambda}_{n-1}) = \frac{p(\mathbf{x})}{X(\lambda)}\,\Theta(L(\mathbf{x}) > \lambda) \tag{6.8}$$

depicts the prior restricted to areas, where the likelihood $L(\mathbf{x})$ exceeds the λ threshold. K is the size of the sample and will from now on be referred to as number of walkers.
Given the likelihood minima $\hat{\lambda}_n$ from the algorithm 6.2.1, the Riemann sum in Eq. 6.5 can be estimated by

$$\hat{Z} = \sum_{n=1}^{n_{max}} \hat{\lambda}_n \Delta \hat{X}_n \tag{6.9}$$

The figures 6.2 and 6.3, used to illustrate the principle of the nested sampling algorithm, are taken from [15] and adapted to the notation used in this work. The first figure illustrates the sorting of elements of a 2 dimensional parameter space. The lower graphs display the increase of the likelihood values over the enclosed prior mass.

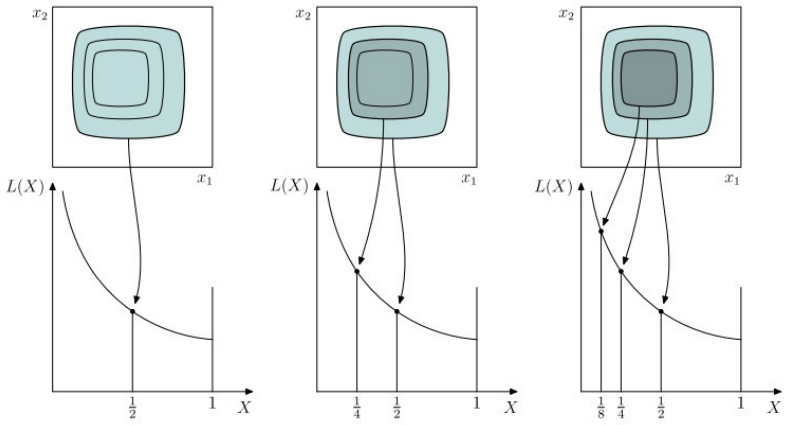


Figure 6.2: Relation of $L(\mathbf{x})$ and $L(X)$ in a 2 dimensional parameter space.

Using multiple walkers to explore the phase space, we first draw the walkers from the unconstrained prior. Then the likelihood of each walker is computed and the values are sorted in increasing order. Instead of sampling all of the walkers from the constrained prior in every update, one can also only update the walker with the lowest likelihood. All other walkers are already a valid sample of the actual constrained prior. Figure 6.3 illustrates such a update. The outermost walker

indicated by the black dot is replaced by its updated version indicated by the purple dot.

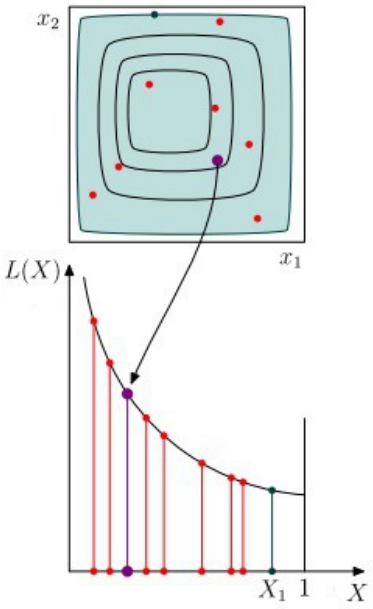


Figure 6.3: Update of the walker with the lowest likelihood. The upper plot depicts the two dimensional parameter space and the lower diagram shows the sorted likelihood values.

Drawing samples from the constrained prior

Sampling from a prior underlying a constraint can be a challenging task. Especially if one is dealing with multi-modal likelihood functions, the increasing threshold leads to disjoint regions during nested sampling. A common approach is to clone a walker, which lies within an allowed region, and then move it by e.g. MCMC in allowed areas of the phase space. For this approach it is necessary for multiple peaks to have enough walkers to populate all of the peaks from the beginning to achieve a proper sampling. Because walker will die out in smaller peaks and be reborn in the higher ones, nested sampling is able to sample the correct height of these structures. Classical MCMC fails in this case.

Up to this point just the likelihood values are known. Although the prior mass values are in principle defined through the relation

$$\hat{\lambda}_n = L(\hat{X}_n)$$

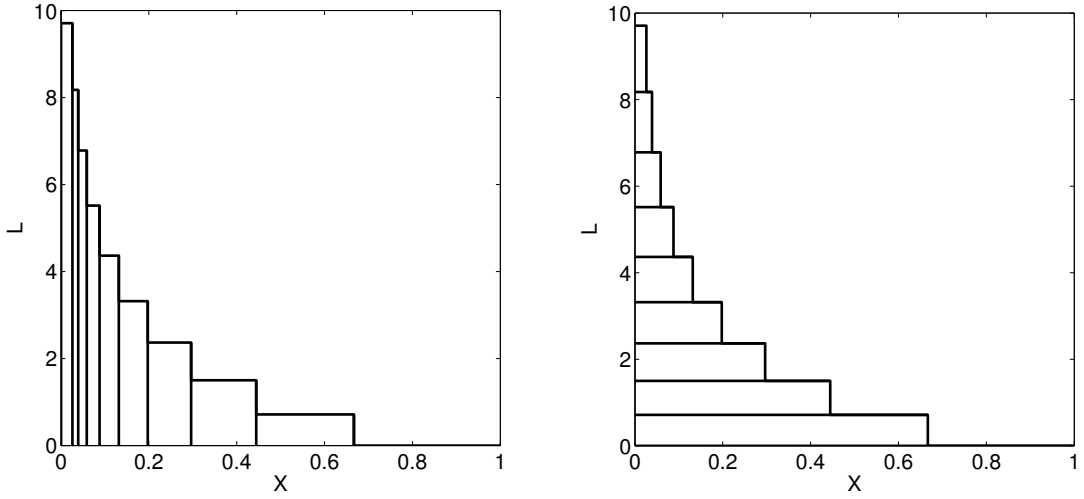


Figure 6.4: Possibilities to evaluate the estimate of Z (see equation 6.11).

where we unfortunately do not know the function $L(X)$. But we know that the values of the prior mass $\hat{X}_n$ are random variables, which are independent from the likelihood values. This independence is one of the key points of nested sampling. The bulk of the statistical analysis of NESA focuses on these variables. We will further analyse the prior mass in the section 6.3.
But before digging deeper into the statistics of $\hat{X}_n$, an alternative way of writing the estimator for Z is presented. We remind that the value of $X_0 = 1$ and the value $\lambda_0 = 0$.

$$Z = \sum_{n=1}^{n_{max}} \hat{\lambda}_n \underbrace{(\hat{X}_n - \hat{X}_{n+1})}_{\Delta\hat{X}_n} = \sum_{n=1}^{n_{max}} \hat{X}_n \underbrace{(\hat{\lambda}_n - \hat{\lambda}_{n-1})}_{\Delta\hat{\lambda}_n} \tag{6.10}$$

$$= \sum_{n=1}^{n_{max}} \hat{\lambda}_n \, \Delta\hat{X}_n = \sum_{n=1}^{n_{max}} \hat{X}_n \, \Delta\hat{\lambda}_n \tag{6.11}$$

For problems with discrete likelihood values, like we obtain for the Potts model, the second representation is advantageous, because $\Delta\hat{\lambda}_n$ will be equal to zero for a lot of the summands. Therefore one just needs to process a summand when a change in the likelihood value occurs. For the Potts model this can happen at most $2N$ times (see the definition of the likelihood in section 6.6.1). This is also useful for storing the resulting likelihood sequence, because in this case just $2N$ values are relevant to keep.

6.2.1 Degeneracy

In the case of discrete systems the occurrence of degenerated likelihood values is possible. For the nested sampling algorithm to work, we need an unique ordering of these values. Therefore an additional independent random variable m, used as label for the likelihood values, is introduced (see [18]). Therefore an enhanced prior probability can be chosen of the form

$$p(\mathbf{x}, m) = p(\mathbf{x})\, p(m) = \frac{1}{Z} L(\mathbf{x}) p(\mathbf{x})\, \frac{1}{Z_m} L(m) p(m)$$

Drawing a sample from this probability, if a likelihood constraint $\hat{\lambda}_n$ is applied yields the restriction

$$L(\mathbf{x}')\, L(m') \; > \hat{\lambda}_n \tag{6.12}$$

on the new likelihood values. Eventually this only affects the nested sampling algorithm if $L(\mathbf{x}_{i'}) = L(\mathbf{x}_i)$. This new configuration is then accepted if $m' > m$, otherwise not. The treatment of this label m during a nested sampling update for the Potts model is described in section 6.6.1.

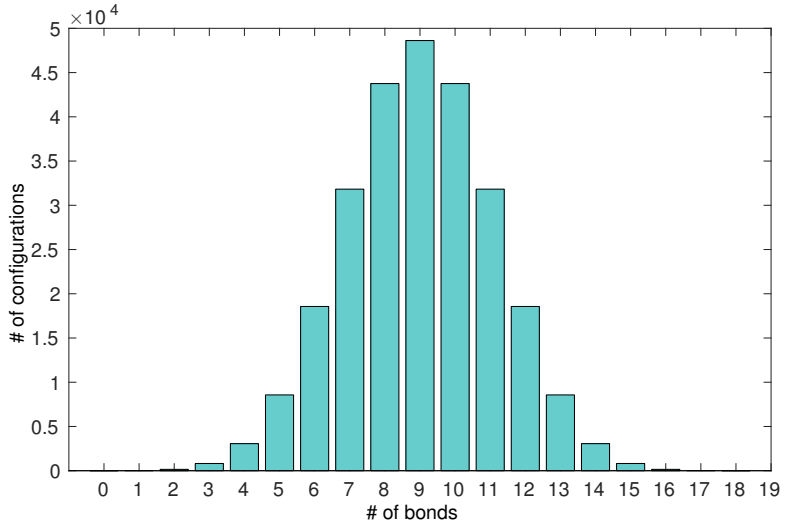


Figure 6.5: Degeneracy of configurations with a certain number of bonds for a 3×3, $q = 2$ system.

As we will see the likelihood values of nested sampling for the Potts model will depend only on the number of active bonds (section 6.6.1). Figure 6.5 displays the

degeneracy of configurations in dependency on the number of active bonds for a 3x3 Potts system. This should emphasize that a proper treatment of degenerated likelihoods is critical for the treatment of the Potts model.

6.3 Statistical analysis of the prior masses

The evaluation of the likelihood-minima sequence is the most time consuming part of the nested sampling algorithm, due to the time intense sampling from the constrained prior. Fortunately we are able to retrieve the distribution of the prior mass. Hence we can obtain the Skilling estimate including an estimate for its error from a *single* nested sampling run.

In the first iteration, without any constraint, the sample of the prior masses is uniformly distributed in the domain $[0, 1]$. If now only one walker is updated, the maximum prior mass, corresponding to the minimal likelihood, of the sample, constrains the values of the prior masses for the next iteration. Thus $[0, \hat{X}_1]$ is the domain for the second iteration. Therefore the probability for iteration n is given by

$$p(X|X_n) = \frac{\Theta(X < X_n)}{X_n} \tag{6.13}$$

$\hat{X}_{n+1}$ is the maximum of a sample of size K, where the samples are uniformly drawn from the interval $(0, X_n)$. Hence the compression at each step $\frac{X_{n+1}}{X_n}$ follows the same distribution as X_1.

For n iterations, we can therefore draw n independent samples $\hat{X}_1$ and compute $\hat{X}_n$ as the product of these samples.

$$\hat{X}_n \;=\; \prod_{i=1}^{n} \hat{X}_1 \tag{6.14}$$

For further analysis of $\hat{X}_n$, we obtain the mean and the variance of $\hat{X}_1$ in the next step.

6.3.1 Order statistic

In the basic nested sampling algorithm the walker with the largest prior mass X_n is discarded. We can also think of the general case of discarding k walkers each step. This will become relevant later for the parallelised version of the nested sampling algorithm. Hence we need to find the expectation value for the kth largest element of the prior masses after each step. It can be retrieved from order statistic. A more detailed explanation of order statistic can be found in [26, p. 119].

The probability density function (PDF) for the kth largest elements in a list of sorted random numbers (in our case the prior mass values in a certain step) drawn from a certain probability density $\rho(x)$ is given via $p(X_k \in (X, X + dX)|K, \rho, I)$.

It should fulfil the following propositions

- K-k elements have a value lower or equal to X
- k-1 elements have a value higher or equal to X
- 1 element has a value in the interval $X + dX$

The definition of the distribution function $F(X)$, yields the probability for one element to be smaller than X. The probability to have a higher value is therefore $1 - F(X)$. The probability for an element to have a value in the interval $X + dX$ equals $\rho(x)dx$.

This yields

$$p(X_k \in (X, X + dX)|K, \rho, I) = \frac{F(X)^{(K-k)}[1 - F(X)]^{(k-1)}}{B(K - k + 1, k)} \rho(X)dX$$

Here $B(K - k + 1, k)$ denotes the Beta function (see appendix). The above probability is similar to the beta distribution (eq. 6.15), which we will exploit soon. The beta distribution and the first and second moment of it, are defined as

$$p_\beta(f|\alpha, \gamma) = \frac{1}{B(\alpha, \gamma)} f^{\alpha-1}(1 - f)^{\gamma-1} \quad \text{with } f \in [0, 1] \tag{6.15}$$

$$\langle x \rangle = \frac{\alpha}{\alpha + \gamma} \tag{6.16}$$

$$\langle x^2 \rangle = \frac{(\alpha + 1)\,\alpha}{(\alpha + \gamma + 1)\,(\alpha + \gamma)} \tag{6.17}$$

To evaluate the mean of the order statistic, the monotonicity of the distribution function and therefore the availability of the inverse function $F^{-1}(X)$ is used. Again the derivation can be found in [26, p. 119]. As result we obtain

$$\langle \hat{X} \rangle_k \;=\; \int_0^1 F^{-1}(f)\, p_\beta(f|K-k+1,k)\, df$$

The samples of prior masses X_n are drawn from a uniform PDF, where the distribution function is given through $F(x) = x$ with $\Theta(0 \leq x \leq 1)$ with the inverse $F^{-1}(f) = f$ in the interval $0 \leq f \leq 1$.
In this case the mean can be read directly from equation 6.16.

$$\langle X \rangle_k \;=\; \int_0^1 f\, p_\beta(f|K-k+1,k)\, df = \frac{K-k+1}{K+1}$$

Depending on the number k of discarded walkers, the mean compression rate at each iteration step is

$$\boxed{\langle X \rangle_k = \frac{K-k+1}{K+1}} \tag{6.18}$$

The second moment then reads

$$\boxed{\left\langle X^2 \right\rangle_k = \frac{(K-k+2)(K-k+1)}{(K+2)(K+1)}} \tag{6.19}$$

Now we can go back to equation 6.14. The mean of $\hat{X}_1$ is given trough equation 6.18. Because the samples are independent, the expectation value of $\hat{X}_n$ can be expressed as product of the individual expectation values.

$$\boxed{\langle \hat{X}_n \rangle \;=\; \prod_{i=1}^{n} \langle \hat{X}_1 \rangle \;=\; X_n} \tag{6.20}$$

Next we want to obtain the covariance $\left\langle \Delta\hat{X}_{n'} \Delta\hat{X}_n \right\rangle$. The derivation for the case of $k = 0$ can be found in [26, p. 617]. Here the expression for an arbitrary k value is derived. According to the symmetry of the covariance matrix, we can set

$n' = n + m$ with $m \geq 0$. Generally

$$\left\langle \Delta\hat{X}_{m+n}\Delta\hat{X}_n \right\rangle = \left\langle \hat{X}_{m+n}\hat{X}_n \right\rangle - \left\langle \hat{X}_{m+n} \right\rangle \left\langle \hat{X}_n \right\rangle$$

With equation 6.20 we already know the last two terms. The first term, evaluated with equation 6.14 and 6.20, gives

$$\begin{aligned} \left\langle \hat{X}_{n+m}\hat{X}_n \right\rangle &= \left\langle \prod_{i=1}^{n+m} \hat{X}_{1_i} \prod_{i'=1}^{n} \hat{X}_{1_{i'}} \right\rangle \\ &= \prod_{i=n+1}^{n+m} \left\langle \hat{X}_{1_i} \right\rangle \prod_{i'=1}^{n} \left\langle \hat{X_{1_{i'}}}^2 \right\rangle \\ &= \left(\frac{K-k+1}{K+1} \right)^m \left(\frac{(K-k+2)(K-k+1)}{(K+2)(K+1)} \right)^n \end{aligned} \tag{6.21}$$

Hence we obtain for the covariance

$$\begin{aligned} \left\langle \Delta\hat{X}_{m+n}\Delta\hat{X}_n \right\rangle &= \left(\frac{K-k+1}{K+1} \right)^m \left(\frac{(K-k+2)(K-k+1)}{(K+2)(K+1)} \right)^n \\ &\quad - \left(\frac{K-k+1}{K+1} \right)^{n+m} \left(\frac{K-k+1}{K+1} \right)^n \\ &= X_{m+n}X_n \left[\underbrace{\left(\frac{(K-k+2)(K+1)}{(K+2)(K-k+1)} \right)^n}_{\Gamma} - 1 \right] \end{aligned} \tag{6.22}$$

with

$$\begin{aligned} \Gamma &= \left(1 + \frac{k}{(K+2)(K-k+1)} \right)^n \\ &= \exp\left(n \ln\left(1 + \frac{k}{(K+2)(K-k+1)} \right) \right) \end{aligned}$$

From equation 6.18 we obtain

$$\ln(X_n) = n \ln\left(\frac{K-k+1}{K+1} \right) = n \ln\left(1 - \frac{k}{K+1} \right)$$

Considering these results, equation 6.22 can be written as

$$\left\langle \Delta\hat{X}_{m+n}\Delta\hat{X}_n \right\rangle = X_{m+n}X_n \left[\underbrace{\exp\left(-ln(X_n)\,\kappa\right)}_{X_n^{-\kappa}} - 1 \right]$$

$$\text{where} \quad \kappa = \frac{\ln\left(1+\frac{k}{(K+2)(K-k+1)}\right)}{\ln\left(\frac{K+1}{K+1-k}\right)}$$

Here κ is of order $1/K$. Finally the covariance yields the following result

$$\boxed{\left\langle \Delta\hat{X}_{m+n}\Delta\hat{X}_n \right\rangle = X_{m+n}X_n \left[\, X_n^{-\kappa} - 1 \,\right]} \tag{6.23}$$

6.3.2 Mean and variance for the estimator of Z

Next we approach the desired quantities $\langle Z \rangle$ and $\langle (\Delta Z)^2 \rangle$. In equation 6.9 an estimator for the partition function was given. The γth moment of this has the following form (see [26, p. 582])

$$\langle Z^\gamma \rangle = \left\langle \left(\sum_{n=1}^{n_{max}} \hat{X}_n \Delta\hat{\lambda}_n \right)^\gamma \right\rangle \tag{6.24}$$

Therefore the mean can be computed as

$$\langle Z \rangle = \sum_{n=1}^{n_{max}} \left\langle \hat{X}_n \right\rangle \Delta\hat{\lambda}_n \quad \text{(with eq. 6.20)}$$

$$\boxed{\langle Z \rangle = \sum_{n=1}^{n_{max}} X_n \Delta\hat{\lambda}_n} \tag{6.25}$$

For the variance we will need also the square of the mean

$$\langle Z \rangle^2 = \sum_{n=1}^{n_{max}} \left\langle \hat{X}_n \right\rangle^2 \Delta\hat{\lambda}_n^2 + \underbrace{\sum_{n=1}^{n_{max}} \sum_{n'=1}^{n_{max}}}_{n \neq n'} \left\langle \hat{X}_{n'} \right\rangle \left\langle \hat{X}_n \right\rangle \Delta\hat{\lambda}_{n'} \Delta\hat{\lambda}_n$$

The second moment is

$$\begin{aligned}
\langle Z^2 \rangle &= \left\langle \left(\sum_{n=1}^{n_{max}} \hat{X}_n \Delta\hat{\lambda}_n \right)^2 \right\rangle \\
&= \left\langle \sum_{n=1}^{n_{max}} \hat{X}_n{}^2 \Delta\hat{\lambda}_n^2 \right\rangle + \left\langle \underbrace{\sum_{n=1}^{n_{max}} \sum_{n'=1}^{n_{max}}}_{n \neq n'} \hat{X}_{n'} \hat{X}_n \, \Delta\hat{\lambda}_{n'} \Delta\hat{\lambda}_n \right\rangle \\
&= \sum_{n=1}^{n_{max}} \left\langle \hat{X}_n{}^2 \right\rangle \Delta\hat{\lambda}_n^2 + \underbrace{\sum_{n=1}^{n_{max}} \sum_{n'=1}^{n_{max}}}_{n \neq n'} \left\langle \hat{X}_{n'} \hat{X}_n \right\rangle \Delta\hat{\lambda}_{n'} \Delta\hat{\lambda}_n
\end{aligned}$$

Inserting the above results we eventually obtain for the variance

$$\begin{aligned}
\left\langle (\Delta Z)^2 \right\rangle &= \langle Z^2 \rangle - \langle Z \rangle^2 \\
&= \sum_{n=1}^{n_{max}} \left\langle (\Delta\hat{X}_n)^2 \right\rangle \Delta\hat{\lambda}_n^2 + \underbrace{\sum_{n=1}^{n_{max}} \sum_{n'=1}^{n_{max}}}_{n \neq n'} \left\langle \Delta\hat{X}_{n'} \Delta\hat{X}_n \right\rangle \Delta\hat{\lambda}_{n'} \Delta\hat{\lambda}_n \\
&= \sum_{n,n'=1}^{n_{max}} \left\langle \Delta\hat{X}_{n'} \Delta\hat{X}_n \right\rangle \Delta\hat{\lambda}_{n'} \Delta\hat{\lambda}_n \qquad (6.26)
\end{aligned}$$

$\left\langle \Delta\hat{X}_{n'} \Delta\hat{X}_n \right\rangle$ is the covariance matrix, derived in section 6.3.1.
In the end we obtain the desired variance of Z:

$$\left\langle (\Delta Z)^2 \right\rangle = \sum_{n,n'=1}^{n_{max}} X_{n'} X_n \left[X_n^{-\kappa} - 1 \right] \Delta\hat{\lambda}_{n'} \Delta\hat{\lambda}_n \quad \text{with } n' \geq n$$

$$\boxed{\left\langle (\Delta Z)^2 \right\rangle = \sum_{n}^{n_{max}} \Delta\hat{\lambda}_n X_n \left(X_n^{-\kappa} - 1 \right) \sum_{n' \geq n}^{n_{max}} \Delta\hat{\lambda}_{n'} \left(2 - \delta_{nn'} \right) X_{n'}} \qquad (6.27)$$

This gives us the possibility to calculate an the mean and the variance for Z out of a single likelihood sequence.

Unfortunately this expression is just useful for very small Potts systems. For larger systems the likelihood values λ_n will massively increase and we are forced to use the logarithmic values, if we want to prevent numerical problems. Therefore another

way of evaluating the variance is presented.

6.3.3 Samples of the prior mass

The way of directly calculating the variance of Z out of a single sequence of likelihood values becomes numerically problematic for large likelihood values. Also then the distribution of Z will not be Gaussian any more, but rather the distribution of $ln(Z)$ [22].
Instead of using the derived expression in equation 6.27, one can also use a set of prior mass sequences, which are drawn form the known distribution. Then multiple estimates for Z can be computed out of a single likelihood sequence. These estimates enable us to retrieve the respective variance.

The extremely large values of L, which are common in problems of our interest, will restrict us to work with the logarithmic values. For correct evaluation of the estimate of $ln(Z)$ see section 9.1 in the appendix. The determination of the logarithmic likelihood value via NESA does not require any principal changes in the algorithm. But to sample the values of $ln(X)$, we need to derive the respective distribution.
First lets us transform our random variable x to a new random variable l

$$x := \hat{X}_n \;\rightarrow\; l := -\ln(\hat{X}_n) \tag{6.28}$$

The transformation keeps the infinitesimal probability mass constant (see [26, p. 121]).

$$p_x(x)dx = p_l(l)dl \;\;\Rightarrow\;\; p_l(l) = p_x(x)\left|\frac{dx}{dl}\right| \tag{6.29}$$

The probability for x being the largest of N uniform random numbers in the interval [0,1] is

$$p_x(x) = K\, x^{K-1}$$

This corresponds to equation 6.15 for k=1. Inserting this and the inverse of equa-

tion 6.28 into equation 6.29 we retrieve as distribution for our new variable l

$$p_l(l) = K \exp(-K\, l) \tag{6.30}$$

$p_l(l)$ exhibits the form of a Γ distribution, hence we can determine the mean and the variance by comparing it to the following definition (see [26])

$$\begin{aligned} p_\Gamma(x|\alpha,\beta) &= \frac{\beta^\alpha}{\Gamma(\alpha)}\, x^{\alpha-1}\, e^{-\beta\, x} \\ \langle x \rangle &= \frac{\alpha}{\beta} \\ var(x) &= \frac{\alpha}{\beta^2} \end{aligned}$$

Here x lies in the range $[0, \infty)$ and $\Gamma(\alpha)$ denotes the Gamma function (see appendix).
So we are able to identify from equation 6.30

$$\begin{aligned} \langle l \rangle &= \frac{1}{K} \\ \langle (\Delta l)^2 \rangle &= \frac{1}{K^2} \end{aligned}$$

Again we are interested in the general case of discarding k walkers at once. Discarding k walkers means that we compress the prior mass k times. The same compression could be achieved via independently taking away k times just one walker. Therefore the mean and the variance for the omission of k walkers yield

$$\langle\, l\, \rangle = (K-k+1)^{-1} + (K-k+2)^{-1} ... + K^{-1} \tag{6.31}$$

$$\langle (\Delta l)^2 \rangle = (K-k+1)^{-2} + (K-k+2)^{-2} ... + K^{-2} \tag{6.32}$$

An estimate for the variance of $ln(Z)$ can be determined as follows (see [26]). First we employ the samples $\hat{l}_n$, drawn from the known distribution, to get an estimate $ln(\hat{Z})$. The respective mean $l_n = n/K$ is used to get an estimate $ln(\bar{Z})$. An estimate for the variance of $ln(Z)$ can then be computed via

$$\left\langle \left(ln(\hat{Z}_K) - ln(\bar{Z}_K) \right)^2 \right\rangle = \left\langle \left[ln\left(\frac{\sum_{n=1}^{n_{max}} e^{-\hat{l}_n} \Delta\hat{\lambda}_n}{\sum_{n=1}^{n_{max}} e^{-\frac{n}{K}} \Delta\hat{\lambda}_n} \right) \right]^2 \right\rangle \tag{6.33}$$

6.3.4 Upper bound of variance

In [26] an analytical expression for the upper bound of the standard deviation of $ln(Z)$ is derived. Below, an outline of this derivation is presented for $k = 1$.
Starting from equation 6.33, we know that $\Delta\hat{\lambda}_n$ is positive, so we can assign normalised weights

$$\omega_n := \frac{e^{-\frac{n}{K}}\Delta\hat{\lambda}_n}{\sum_{n=1}^{n_{max}} e^{-\frac{n}{K}}\Delta\hat{\lambda}_n}, \qquad \omega_n > 0, \qquad \sum_{n=1}^{n_{max}} \omega_n = 1$$

This yields

$$\left\langle \left(ln(\hat{Z}_K) - ln(\bar{Z}_K) \right)^2 \right\rangle = \left\langle \left[ln\left(\sum_{n=1}^{n_{max}} \omega_n e^{-\Delta\,\hat{l}_n} \right) \right]^2 \right\rangle$$

where $\Delta\hat{l}_n = \hat{l}_n - \left\langle \hat{l}_n \right\rangle$. For obtaining an upper bound for the variance all summands can be replaced by the largest one, say $e^{-\Delta\,\hat{l}_{n'}}$, which leads to

$$\begin{aligned} \left\langle \left(ln(\hat{Z}_K) - ln(\bar{Z}_K) \right)^2 \right\rangle &\leq \left\langle \left[ln\left(e^{-\Delta\,\hat{l}_{n'}} \sum_{n=1}^{n_{max}} \omega_n \right) \right]^2 \right\rangle \\ &= \left\langle \left[ln\left(e^{-\Delta\,\hat{l}_{n'}} \right) \right]^2 \right\rangle \\ &= \left\langle \left(\Delta\,\hat{l}_{n'} \right)^2 \right\rangle = \frac{n'}{K^2} \end{aligned}$$

The upper bound will therefore be maximal for $n' = n_{max}$.
Finally we arrive at the result for the upper bound for the standard deviation of $ln(Z)$

$$\boxed{\sqrt{\left\langle \left(ln(\Delta\,\hat{Z}_K) \right)^2 \right\rangle} \leq \frac{\sqrt{n_{max}}}{K}} \tag{6.34}$$

6.4 Quadrature and truncation error

Besides the statistical uncertainty stemming from the statistic of the prior mass, we need to take the error of the quadrature procedure and the error of the truncation of the likelihood sequence into account.

The quadrature error of the Riemann sum is easily accessible through the lower (eq. 6.5) and upper (eq. 6.7) bound. From the mean prior compression rate (eq. 6.18) we derived for X_n

$$X_n = \left(\frac{K-k+1}{K+1}\right)^n \tag{6.35}$$

and therefore we retrieve for the lower bound

$$\begin{aligned} X_n - X_{n+1} &= \left(\frac{K-k+1}{K+1}\right)^n \left(1 - \frac{K-k+1}{K+1}\right) \\ &= \left(\frac{K-k+1}{K+1}\right)^n \left(\frac{k}{K+1}\right) \end{aligned}$$

and respectively for the upper bound

$$\begin{aligned} X_{n-1} - X_n &= \left(\frac{K-k+1}{K+1}\right)^n \left(\frac{K+1}{K-k+1} - 1\right) \\ &= \left(\frac{K-k+1}{K+1}\right)^n \left(\frac{k}{K-k+1}\right) \end{aligned}$$

The boundaries for the exact value are therefore given by

$$\boxed{ln(Z) \;\leq\; ln(Z_{exact}) \leq\; ln\left(\frac{K+1}{K-k+1}\right) \;+\; ln(Z)} \tag{6.36}$$

With the assumption that we compute the sum of Z after n_{max} nested sampling steps, the truncation error can be estimated by the residue

$$R_{n_{max}} = \sum_{n=n_{max}+1}^{\infty} \hat{\lambda}_n \left(X_n - X_{n+1}\right)$$

Once the minimal likelihood values $\hat{\lambda}_n$ converge, the residue can be approximated by

$$\begin{aligned} R_{n_{max}} &\approx \hat{\lambda}_{n_{max}} \sum_{n=n_{max}+1}^{\infty} \left(X_n - X_{n+1}\right) \\ &= \hat{\lambda}_{n_{max}} X_{n_{max}} \end{aligned}$$

Accordingly we can write for the error due to truncation after n_{max} steps

$$\boxed{R_{n_{max}} \approx \hat{\lambda}_{n_{max}} \left(\frac{K-k+1}{K+1}\right)^{n_{max}}} \tag{6.37}$$

6.5 Parallel implementation

Observing the computational time of nested sampling, one identifies the update of each walker within the likelihood constraint as the most time consuming part. For example, as presented in section 6.6.1, the update for the Potts model is performed by means of the Swendsen Wang algorithm (see section 3.2.2). Here especially the search for the clusters in a given configuration costs most of the time.
As we found in section 6.3.3 the mean and the variance of the logarithmic prior compression yield

$$\begin{aligned} \langle\, l \,\rangle &= (K-k+1)^{-1} + (K-k+2)^{-1} ... + K^{-1} \\ \langle (\Delta l)^2 \rangle &= (K-k+1)^{-2} + (K-k+2)^{-2} ... + K^{-2} \end{aligned}$$

According to this values, Skilling argues in [24] that the replacement of just one walker yields the highest accuracy. To compensate this loss in accuracy, equation 6.32 suggests, to increase the number of live walkers K. For a higher number of walkers K the mean shrinkage of the prior mass (eq. 6.31) becomes smaller. Hence the number of iterations to reach all of the relevant prior mass regions increases. Although with a higher number of walkers, regions of the prior, which are separated through the likelihood constraint are more likely completely covered. This comes especially into account when dealing with multi-modal likelihood functions.

The idea of parallelising the method has recently found its way into some publications ([11], [6],[16], [28]) Also Skilling already thought of the possibility of a parallel implementation in his original paper [22, p. 846].

Henderson et al. proposed therefore to parallelise the nested sampling algorithm and take the advantage of a large number of walkers while reducing the time needed for a nested sampling run [11]. Instead of one walker, they discard k walkers at each iteration. The update of each walker position in parameter space is subsequently

calculated in parallel. Henderson et al. derived the proper scaling of the number of live walkers with respect to the number of omitted walkers, so that the value of the variance is kept constant. The derivation as described in [11] is presented hereafter.

We demand the variance to stay constant

$$\langle (\Delta l_1)^2 \rangle = \langle (\Delta l_k)^2 \rangle \tag{6.38}$$

The subscripts here denote the number of omitted walkers each step.
Inserting equation 6.32 into equation 6.38 yields

$$\begin{aligned} \frac{1}{K_1^2} &= \sum_{n=0}^{k-1} \left(\frac{1}{K_k - n} \right)^2 \\ &= \frac{1}{K_k^2} + \sum_{n=1}^{k-1} \left(\frac{1}{K_k - n} \right)^2 \\ \Rightarrow K_1 &= \frac{K_k}{\sqrt{1 + \sum_{n=1}^{k-1} \left(\frac{K_k}{K_k - n} \right)^2}} \end{aligned}$$

We assume $K >> k$, that means the number of live samples K needs to be much greater than the number of discarded walkers k. This leads to the relation

$$\boxed{K_k \approx \sqrt{k}\, K_1} \tag{6.39}$$

Substituting this relation into the equation for the mean gives

$$\langle\, l_1 \,\rangle = \frac{1}{K_1} \approx \frac{\sqrt{k}}{K_k} \tag{6.40}$$

Respectively

$$\langle\, l_k \,\rangle = \frac{1}{K_k} + \sum_{n=1}^{k-1} \frac{1}{K_k - n} \approx \frac{k}{K_k} \tag{6.41}$$

Therefore the mean for an abitrary k can be expressed as

$$\langle\, l_k \,\rangle = \sqrt{k}\, \langle\, l_1 \,\rangle \tag{6.42}$$

Eventually if we increase k, given the requirement of constant variance, we arrive at the result that the shrinkage of the prior mass at each iteration increases compared to the case of one walker. Because of this increase in shrinkage, we expect the walkers to reach convergence with fewer steps.

6.5.1 Parallel nested sampling algorithm

For the parallel implementation the earlier presented nested sampling algorithm (6.2.1) needs some modifications.

Algorithm 6.5.1: PARALLEL NESA ALGORITHM($\hat{\lambda_n}, n_{max}$)

initialize $\hat{\lambda_0} = 0,\ n = 0,$

draw sample $\{x_i\}$ of size K from $p(\mathbf{x}|\hat{\lambda}_0)$
determine sorted list of likelihood values $\lambda_i = L(x_i)$
determine k^{th} minimal likelihood $\hat{\lambda_1}$
set $n = 1$
while $\lambda_{n+1} - \lambda_n\ >\ \epsilon_\lambda$

do
- increment the iteration count $n \leftarrow n + 1$
- **parallel**
  - start thread $j = 1 : k$
  - draw x_n^j from $p(\mathbf{x}|\hat{\lambda}_{n-1})$
- insert the new likelihood values $\lambda_n^j = L(x_n^j)$ into sorted list
- determine k^{th} minimal likelihood $\hat{\lambda}_n$

set $n_{max} = n$
return $(\hat{\lambda}_n, n_{max})$

In the standard realisation of NESA only the walker with the minimal likelihood value needs to be found. If one discards multiple walkers k, respectively the k minimal likelihood values are wanted. Henderson et al. suggest using a so called *partial quicksort* routine [11]. We used a slightly different approach. Instead of sorting the likelihood values each iteration, we kept them listed in increasing order. For each iteration, the walkers with the k least values are updated and sorted into

the list with their new values. The sorting is done using the so called *binary search* algorithm. This performs better than looking for the correct insertion place brute force.

6.5.2 OpenMP

We used OpenMP (Open Multi Processing) to implement the parallel nested sampling algorithm.
Our problem represents a case of a so called *embarrassingly parallel* problem, because the single threads do not need to communicate throughout their tasks, hence OpenMP can be implemented in a straight forward manner.

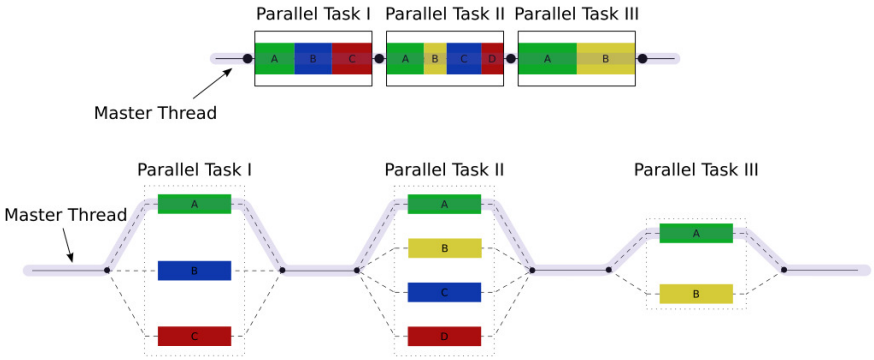


Figure 6.6: Spawning of threads with OpenMP [1]

Figure 6.6 shows how in principle a serial code is parallelised with OpenMP. In our realisation we spawn threads for every update process of the walkers. Each thread gets a walker and computes the evolution through phase space. Each thread awaits the update of all walkers. Then the master thread sorts the new likelihood values into the present list, where the likelihood values of all live walker are stored.

6.6 Nested sampling for Potts models

6.6.1 Adaption of Nested sampling

In [18] it is shown how the formalism of nested sampling can be applied to Potts models in the representation of Fortuin and Kasteleyn. Starting from the FK model, which we obtained in equation 2.10, we can rewrite the bond distribution

like

$$\begin{aligned} P(\mathbf{d}) &= \frac{1}{Z_P} p^{D(\mathbf{d})} (1-p)^{E-D(\mathbf{d})} q^{C(\mathbf{d})} \quad \text{with } p = (1 - e^{-J\beta}) \\ &= \frac{1}{Z_P} e^{\kappa D(\mathbf{d})} e^{-J\beta E} q^{C(d)} \quad \text{with } \kappa = \ln(e^{J\beta} - 1) \end{aligned} \tag{6.43}$$

The quantity we are interested in, the partition function Z_P, then reads as

$$\begin{aligned} Z_P &= \sum_{\mathbf{d}} Z_\pi \, e^{\kappa D(\mathbf{d})} \, e^{-J\beta E} \, \frac{q^{C(d)}}{Z_\pi} \\ &= Z_\pi \, e^{-J\beta E} \underbrace{\sum_{\mathbf{d}} e^{\kappa D(\mathbf{d})} \, \frac{q^{C(d)}}{Z_\pi}}_{Z_{\mathrm{NESA}}} \end{aligned} \tag{6.44}$$

We choose $q^{C(d)} \propto \pi(\mathbf{d})$ as our prior probability for the NESA algorithm. To get a proper normalized prior the summands in equation 6.44 are extended by a factor $\frac{Z_\pi}{Z_\pi}$, where Z_π denotes the prior normalization

$$Z_\pi = \sum_{\mathbf{d}} q^{C(d)}$$

The likelihood function is then defined as

$$L(\mathbf{d}) = e^{\kappa D(\mathbf{d})} \quad \text{with } D(\mathbf{d}) = \sum_{i,j} d_{ij} \tag{6.45}$$

We can use NESA to determine

$$Z_{\mathrm{NESA}}(\beta) = \sum_{\mathbf{d}} L(\mathbf{d}) \, \pi(\mathbf{d}) \tag{6.46}$$

Afterwards the wanted partition function of the Potts model $Z_P(\beta)$ can be obtained via

$$\boxed{Z_P(\beta) = Z_{\mathrm{NESA}}(\beta) \, Z_\pi \, e^{-J\beta E}} \tag{6.47}$$

Noteworthy that the prior normalization is not a function of β: $Z_\pi \neq Z_\pi(\beta)$. To obtain Z_π we can try to exploit $Z_P(\beta)$, which is known for $\beta = 0$ and $\beta = \infty$.

For $\beta = 0$ we find

$$Z_P(0) = q^N = Z_{\mathrm{NESA}}(0)\, Z_\pi \;\rightarrow\; Z_\pi = \frac{q^N}{Z_{\mathrm{NESA}}(0)}$$

If $\beta = 0$ it follows that $\kappa = -\infty$ and hence

$$\begin{aligned} L(\mathbf{d}) &= 1 \quad \text{for} \quad D(\mathbf{d}) = 0 \\ L(\mathbf{d}) &= 0 \quad \text{for} \quad D(\mathbf{d}) > 0 \end{aligned}$$

According to our prior $\pi(\mathbf{d})$, configurations with $D = 0$ will be very unlikely and therefore sampling of $Z_{\mathrm{NESA}}(0)$ is extremely inefficient.

In the limit $\beta \rightarrow \infty$ we find

$$Z_P(\beta \rightarrow \infty) = q = Z_{\mathrm{NESA}}(\beta \rightarrow \infty)\, Z_\pi\, e^{-J\beta E} \rightarrow\; Z_\pi = \frac{q}{Z_{\mathrm{NESA}}(\beta \rightarrow \infty)\, e^{-J\beta E}}$$

For $\beta \gg 1$ we find $\kappa \approx J\beta$ and hence

$$\begin{aligned} L(\mathbf{d})\, e^{-J\beta E} &= 1 \quad \text{for} \quad D(\mathbf{d}) = E \\ L(\mathbf{d})\, e^{-J\beta E} &= 0 \quad \text{for} \quad D(\mathbf{d}) < E \end{aligned}$$

Again sampling configurations with $D = E$ from our prior $\pi(\mathbf{d})$ is not useful due to the lousy statistics for this case. Hence, because we do not know any other values for $Z_P(\beta)$, we are not able to retrieve Z_π from equation 6.46. Therefore we need to determine Z_π separately. $J\beta = ln(2)$ provides an optimal value of evaluation, because then $\kappa = 0$, $L(\mathbf{d}) = 1$ and hence $Z_{\mathrm{NESA}}(\beta = ln(2)) = 1$. Thus for this $J\beta$ we do not need to compute Z_{NESA}. Inserting this into equation 6.47, we find

$$Z_P(\beta = ln(2)) = Z_\pi\, e^{-ln(2)\, E} = Z_\pi\, \frac{1}{2^E} \tag{6.48}$$

Hence, via a separat evaluation of $Z_P(\beta = ln(2))$ for example by thermodynamic integration (as by Murray et al. [18]) we can determine Z_π.

Next we have a closer look on the likelihood definition and how it affects the NESA algorithm. The value of the likelihood depends on the number of active bonds D. The value of $J\beta$ determines if the value of the likelihood L will either increase or

decrease, with a increase in the number of bonds D. In the NESA algorithm we have to be able to sort the likelihood values in increasing order. To ensure that behaviour one has to make a distinction between the following cases:

- $J\beta < \ln(2) \rightarrow$ the number of active bonds D has to decrease
- $J\beta > \ln(2) \rightarrow$ the number of active bonds D has to increase

For us only the second case is of further relevance, because by means of the previous TI run, we already know the values of Z for $J\beta < ln(2)$.
The steps in a single nested sampling update for the Potts model are:

1. Start out from a given bond occupation $\mathbf{d}$
2. Compute the number D of bonds in $\mathbf{d}$
3. Then determine the clusters and choose at random a spin configuration $\mathbf{s}$ a la Swendsen Wang.
4. Given $\mathbf{s}$, determine the number of possible bonds, which represents the number of nearest spins with the same orientation.
5. a.) According to probability p_0 (see figure 6.7) decide, if the number of bonds D' in the new bond occupation $\mathbf{d}'$ is the same as the number in $\mathbf{d}$, namely $D' = D$, or not.

   b.) Otherwise (with probability $1 - p_0$) the number of bonds is determined according to the probability

   $$P(D'|E(\mathbf{s})) = 2^{-E} \binom{E}{D'} \ . \tag{6.49}$$

   This is the binomial distribution with the probability 1/2 for a single realisation (bond set or not).
6. Eventually there are $\binom{E}{D'}$ ways to place D' bonds on the E possible sites. This bond configuration is from the set $D(\mathbf{s})$, where one is chosen uniformly.

In other words, the probability is

$$P(\mathbf{d}'|\mathbf{s}, D') = \frac{1}{\binom{E}{D'}} \Theta(\mathbf{d}' \in D(\mathbf{s})) \tag{6.50}$$

In section 6.6.1 a closer look on the statements of the fifth step in the above list is taken.

Selection of the new number of bonds

The selection of the new number of bonds is the important difference of the nested sampling approach to the general SW algorithm. Here the likelihood constraint on the prior comes into play.
As described in equation (6.50), the new number gets chosen out of a truncated binomial distribution. Lets have a look at this in more detail for an increasing bond number. For putting D' bonds on E possible positions the binomial distribution

$$\binom{E}{D'} \tag{6.51}$$

is obtained. With the constraint for increasing likelihood values, the number of the actual bonds D needs to increase. Therefore we avoid smaller values by truncating the binomial distribution for $D' < D$. To get a proper probability distribution from which samples, for the new bond number can be drawn, the truncated distribution needs to be normalized. Figure 6.7 shows a schematic illustration of a truncated binomial distribution.
The decision tree shown in figure 6.8 illustrates how to obtain the probability for either the branch (1) $D' > D$ or the branch (2) $D' = D$. In the graph r_1 and r_2 denote random numbers, uniformly distributed between 0 and 1. p_0 is indicated in figure 6.7.
Let us analyse the decision tree to retrieve a probability for each branch. If the first condition $r_1 < p_0$ is fulfilled the second one ensures an increasing likelihood value via increasing the random label m. This means that the number of bonds stays only the same if $m' > m$. Otherwise we start from the top, a new random number r_1 is drawn and the first condition has to be checked again. Continuing this procedure we can derive the following geometrical series, which enables us

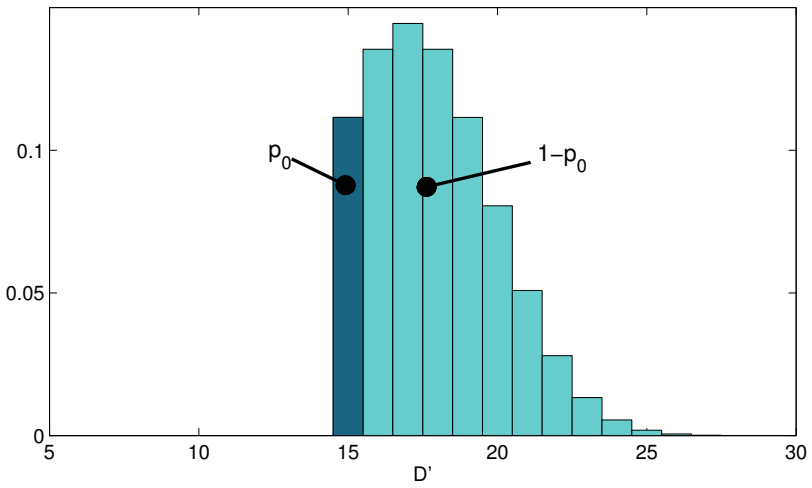


Figure 6.7: Truncated binomial distribution of the new number of bonds D'

to assign probabilities for both branches. For the case $D' > D$ the probability therefore reads

$$\begin{aligned} p_{1.branch} &= (1-p_0) + (1-p_0)\, p_0 m + (1-p_0)\, (p_0 m)^2 + ... \\ &= (1-p_0)\, \frac{1}{1-p_0 m} \end{aligned} \tag{6.52}$$

and for the second case, where D' retains the value of D,

$$\begin{aligned} p_{2.branch} &= p_0(1-m) + p_0 m\, p_0(1-m) + (p_0 m)^2\, p_0(1-m) + ... \\ &= p_0(1-m)\, (1 + p_0 m + (p_0 m)^2 + ...) \\ &= p_0(1-m)\, \frac{1}{1-p_0 m} \end{aligned} \tag{6.53}$$

The sum $p_{1.branch} + p_{2.branch} = 1$ assures the right normalization.

If the bond number stays constant one needs to compute a new random label m'.

$$m' = m\,(1-m)\,r, \tag{6.54}$$

where r denotes a uniform random number. This way the constraint $m' > m$ is satisfied. If a change in the bond-number is accepted, one calculates the discarded and normalized binomial distribution again, which contains only $D' > D$ and randomly draws the new number. Because of the changing bond-number the label

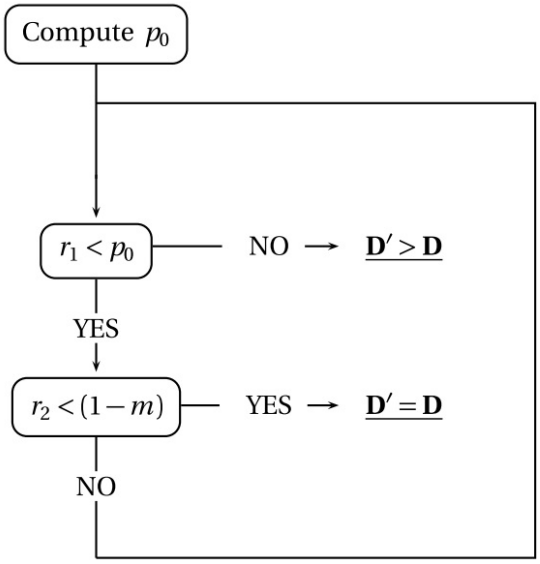


Figure 6.8: Decision tree for evaluating the probability of the two displayed cases.

m' is not subject to any constraint and is chosen randomly $m' = r$.

If the likelihood constraint requires the bond number to decrease, the procedure is analogous to the above one, exept of a slight difference, that needs to be taken into account. If the possible number of bonds E is lower than the actual number D one needs to ensure that the constraint not just fulfils $D' \leq D$ but also $D' \leq E$.

Performance comparison of bond selection

The presented approach of computing the new bond number is to prefer over the straight forward way of calculating the whole truncated binomial distribution and drawing the new bond number from it (as in [18]). The truncated binomial distribution will decline very fast and hence a lot of possible new bond numbers will be very improbable.
Comparing our approach with the straight forward method yields a considerably reduction of CPU time needed for one NESA run (see figure 6.9). For larger systems the time of our version is in the 10 percent range of the second one.

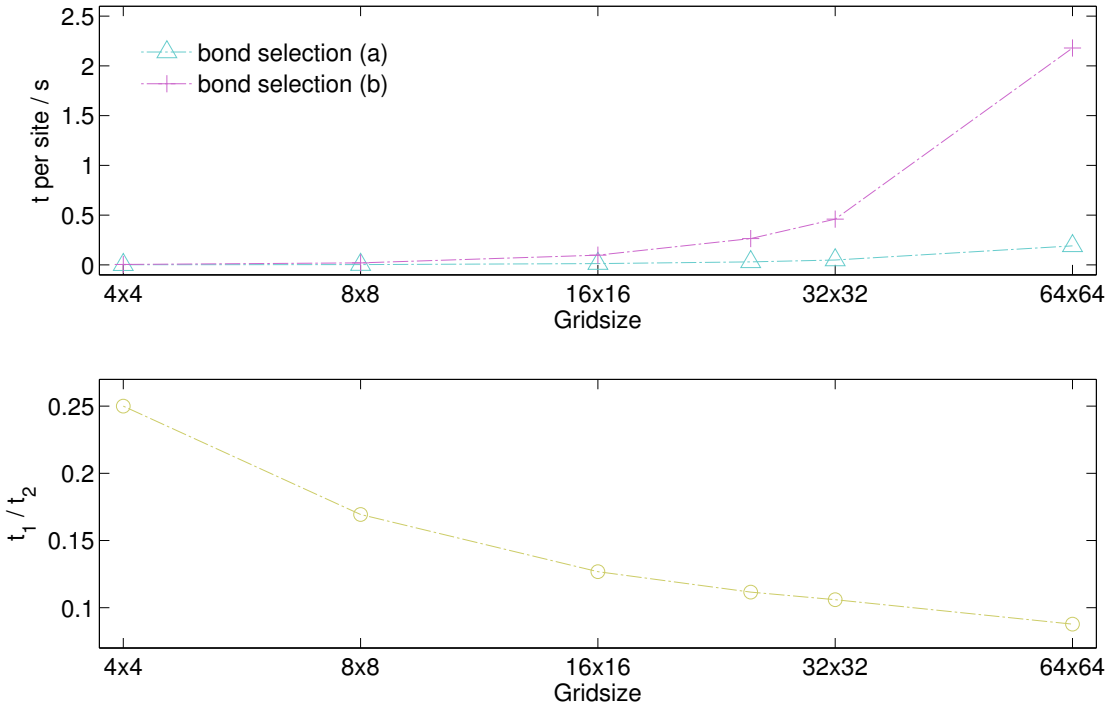


Figure 6.9: Comparison of the CPU times for a nested sampling computation, applied on a $q = 10$ Potts model, for the two ways (see text) of choosing the new bond number.

Thermalisation

In the beginning, the randomly chosen spin colouring with the resulting random bond occupancy is unlikely to be a representative state of the system. The system needs a certain amount of updates to reach relevant areas in state space. Therefore at the beginning of each simulation multiple SW steps are executed. This initial process is referred to as *thermalisation* of the system.

The prior normalisation

The prior normalisation $\mathbf{Z}_\pi$ of equation 6.44 is not known. It needs to be calculated separately. We use thermodynamic integration (see section 4) to compute Z_π at $J\beta = ln(2)$. In [18] the same approach has been used. This separate calculation is a drawback in the use of nested sampling for the Potts model. Although, the need of an additional computation arises, the low $J\beta$ value ensures that TI does not have to deal with diverging correlation times (see section 4.1.1). Hence its calculation is computationally still fairly cheap.

Analysis of the Z_π evaluation

The CPU time needed by the thermodynamic integration method will of course depend on the number of $J\beta$ values, at which $ln(Z)$ is calculated, and furthermore on the number of the SW updates at each β_i. For thermalisation 10 percent of the SW updates at each β_i are discarded. Before evaluating $ln(Z)$ for different system sizes, we want to know how these parameters will affect the outcome. All evaluations in this section were performed for a $q = 10$ system.

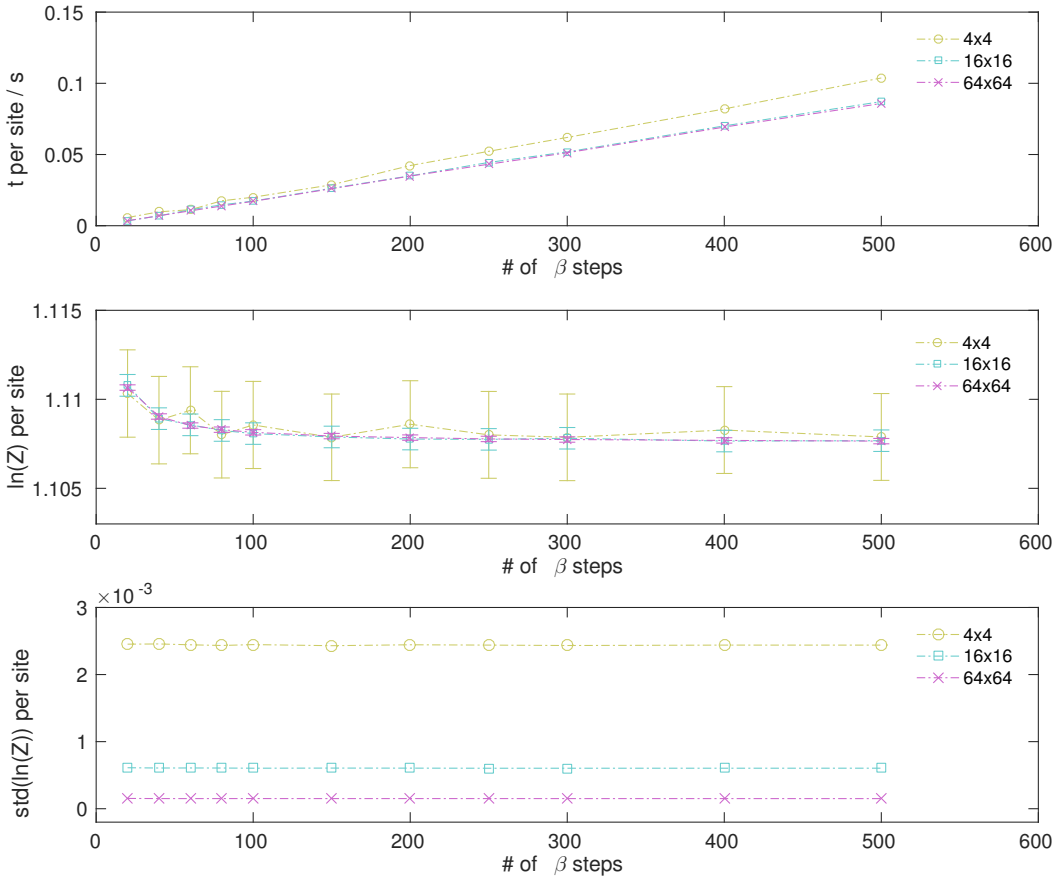


Figure 6.10: Dependence of the time and partition function per site on the number of $J\beta$ steps with a maximum value of $J\beta_{max} = ln(2)$. The computations were performed for a $q = 10$ Potts system for various grid-sizes.

In figure 6.10 the dependence of the partition function on the number of $J\beta$ steps is shown. Here 2000 SW updates (hence with a mean correlation time of 2 we obtain 500 effective updates) have been evaluated at each $J\beta$. The time per site shows a linear dependence, which is intuitively correct, because the same amount of calculations have to be executed for each step. The value of $ln(Z)$ per site

in the second subplot converges at about 250 $J\beta$ steps, the respective standard deviations stays fairly constant over the whole range of $J\beta$ steps. Hence we will use 250 $J\beta$ steps for the following calculations.

In figure 6.11 the number of SW updates (for the effective number the same consideration as above applies) for each measurement is successively increased. Again the time per site rises linearly with increasing number of updates, because at each update the same amount of computational work is done. $ln(Z)$ varies very little over the plotted range and the standard deviation shows the MC typical $\frac{1}{\sqrt{x}}$-decay with increasing number of SW updates. This is emphasized in figure 6.12. A $f(x) = a/\sqrt{x}$ fit displays an excellent agreement.

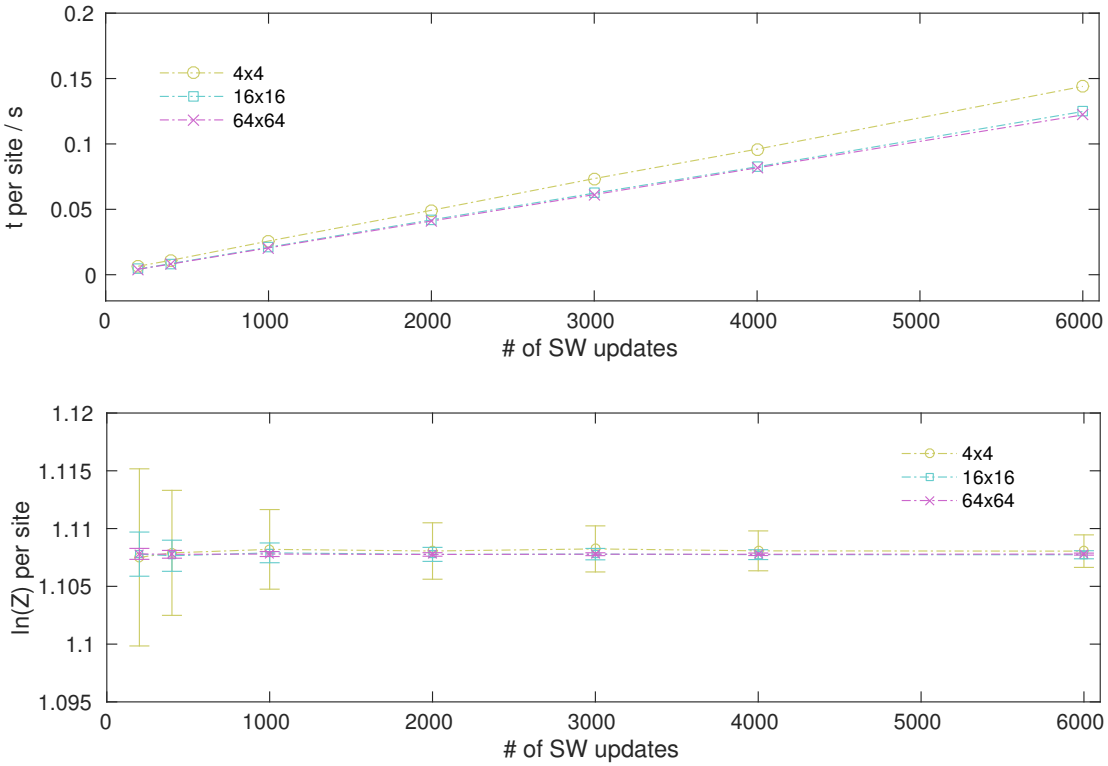


Figure 6.11: Dependence of the time and partition function per site on the number of SW updates at $J\beta = ln(2)$. The computations were performed for a $q = 10$ Potts system for various grid-sizes.

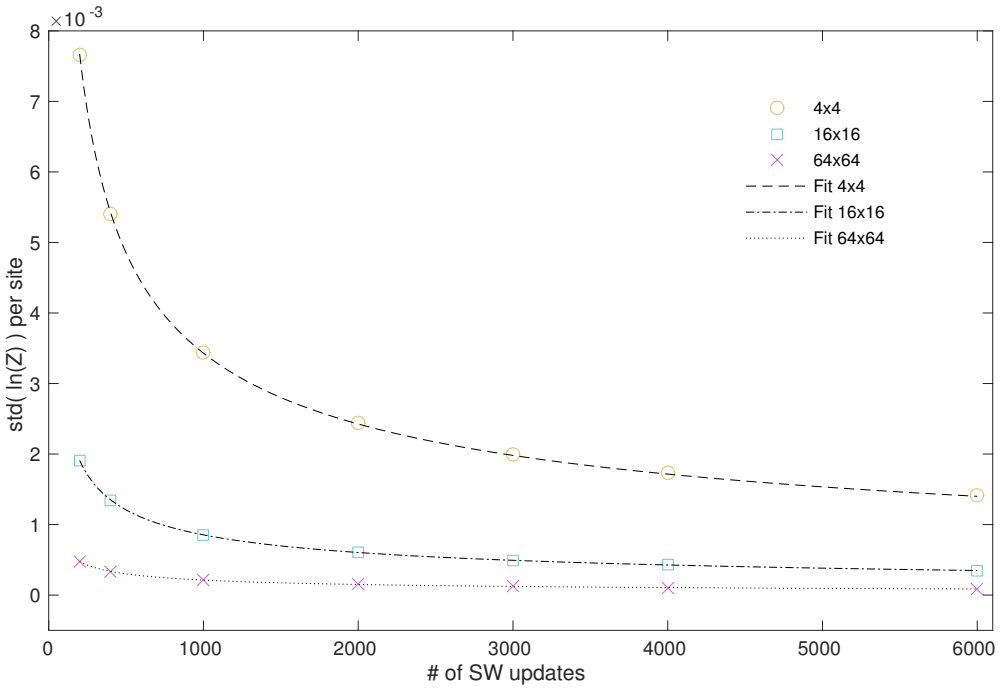


Figure 6.12: The standard deviation of the partition function for the $q = 10$ Potts model and various grid sizes shows the expected $1/\sqrt{x}$ decay over the number of SW updates.

Due to the previous analysis, we determine the parameters for the evaluation of the prior normalisation. 250 β steps and 4000 (1000 effective) SW updates allow a good trade-off between CPU time and accuracy. Figure 6.13 shows the required CPU time per site for the Potts system with $q = 10$ colours in dependence on the system size for these parameters. It stays quite constant for sizes in the range from 16×16 to 512×512. For small systems, the parts of the code, which do not scale with the system size (like e.g. initialisation, in- and output), claim a higher percentage of the total time. Therefore the time per site results in little larger values for these sizes.

The results for $ln(Z_\pi)$ per site are plotted in the lower subfigure. The standard deviation decreases due to *self averaging* of the system. Larger systems can be pictured as combination of e.g. multiple 4×4 systems. Therefore the standard deviation exhibits a decrease for larger systems (see figure 6.14), the same way as we would obtain for using the respective number of these smaller systems.

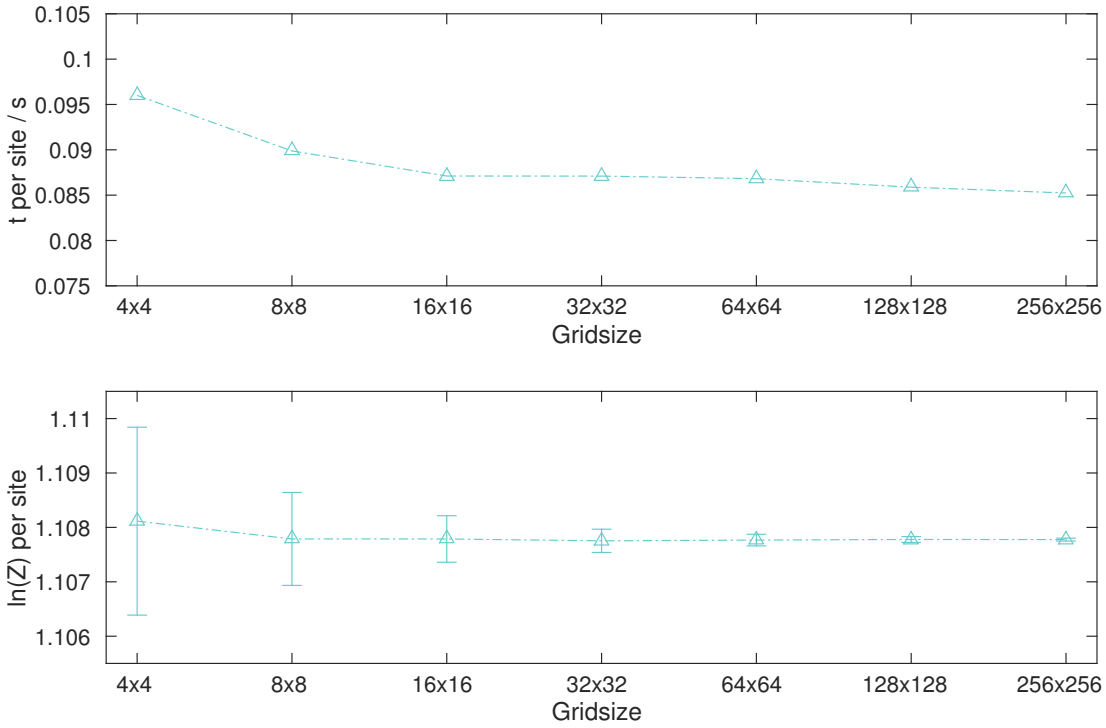


Figure 6.13: CPU time per site and $ln(Z_\pi)$ per site for a $q = 10$ Potts model and $J\beta_{max} = ln(2)$ in dependence on the grid-size, computed with 4000 SW updates and 250 β steps.

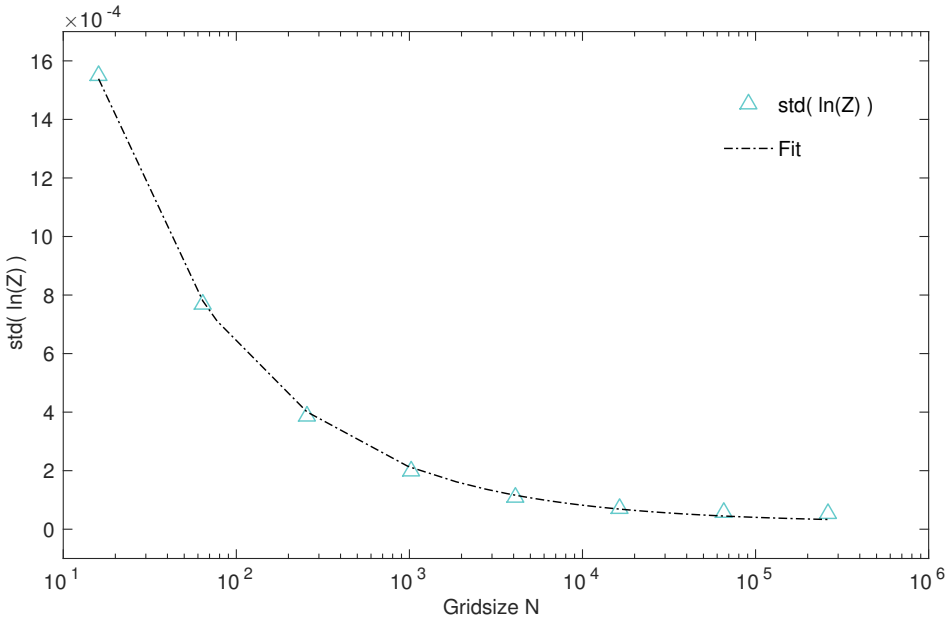


Figure 6.14: The standard deviation of the logarithmic partition function for the $q = 10$ Potts model shows the expected $1/\sqrt{x}$ decay over the increasing grid-size due to self averaging.

6.6.2 Brute force calculation of small Potts systems

Generally a system with N spins and q colours exhibits q^N possible configurations if no restrictions are applied. For a small number of spins and colours, we are able to calculate the exact partition function. Though, as the lattice size or the number of colours increases just a little, the partition function becomes soon intractable by such a brute force approach.
For the system of size 3×3 the partition function for $q = 2$ and $q = 5$ have been computed exactly (see figure 6.15). These exact results can be used for the verification of the simulation estimates of nested sampling, multicanonical sampling or thermodynamic integration.

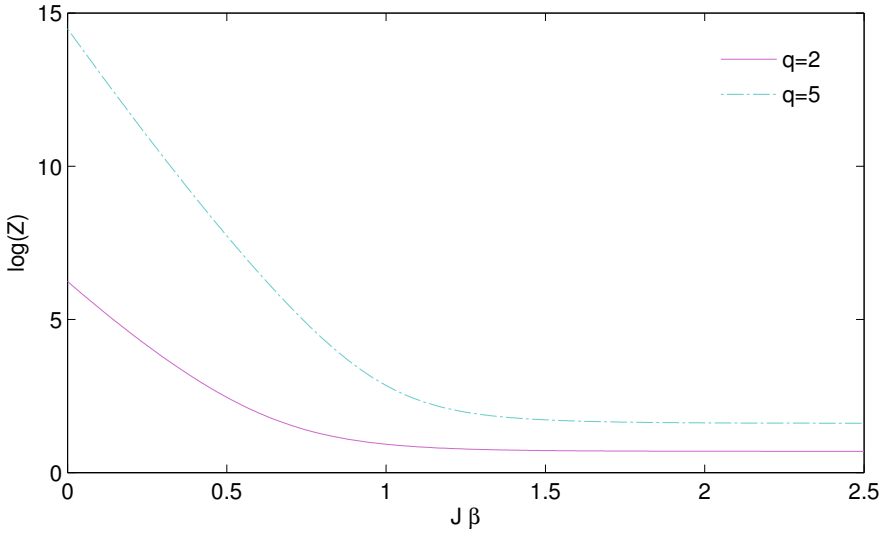


Figure 6.15: Exact solution of the partition function for $3x3$ systems with the number of colours $q = 2$ and $q = 5$.

The partition function can be computed with the Hamiltonian expressed either in the spin $\mathbf{s}$ or in the bond $\mathbf{d}$ variables.

$$\begin{aligned} Z(\mathbf{s}) &= \sum_{\mathbf{s}} \exp(-\beta H) \\ Z(\mathbf{d}) &= \sum_{\mathbf{d}} \exp(-\beta H) \end{aligned}$$

In the spin representation our Hamiltonian is of the form (compare section 2)

$$H_{Potts} = -J \sum_{\langle i,j \rangle} (2\,\delta_{s_i,s_j} - 1) \tag{6.55}$$

The summation then runs over 2^9 possible configurations for 9 spins with 2 possible states. Equivalent for 5 states, it sums over 5^9 possible configurations.

The second way is to calculate Z_{exact} with the Hamiltonian expressed in bond variables. The partition function is then calculated as in equation 2.10. Now the number of possible configurations is 2^{18} for a system with 9 spins and $q = 2$, because instead of 9 spins with up or down orientation, there are now 18 bonds which can be active or not.

The calculation of Z_{exact} according to equation 2.10 needs the number of clusters $C(\mathbf{d})$. They are not required in the spin representation. Hence, through comparison of the two results for Z_{exact}, the outcomes of the subprogram for the cluster identification, which is used for the SW update in nested sampling and thermodynamic integration, can be verified. The dependence of the resulting partition functions on the inverse temperature β is shown in figure 6.16. Since the two representations of the Hamiltonian are equivalent, the two calculations yield the same results.

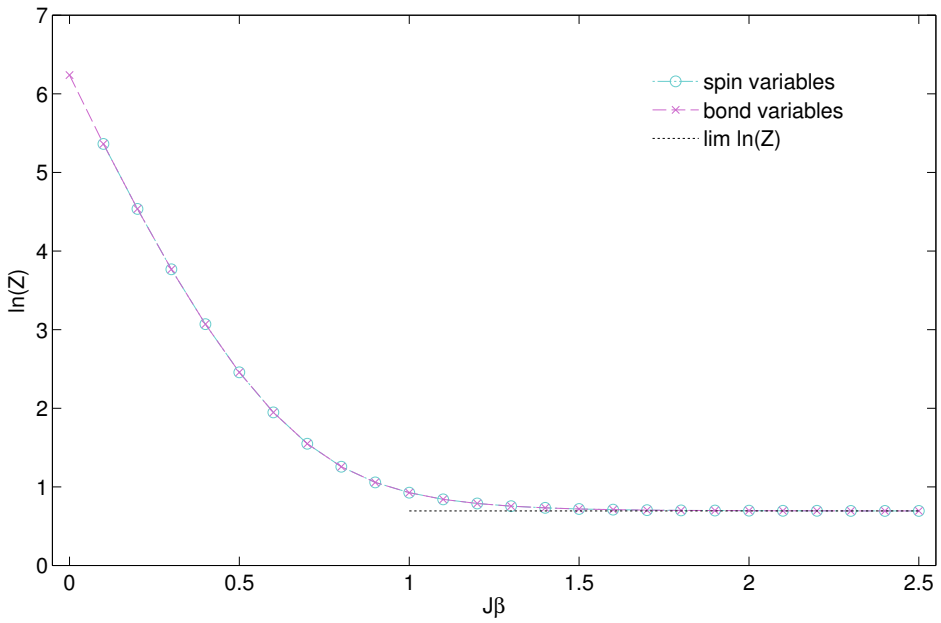


Figure 6.16: Exact partition function computed in spin and bond variables for the 3×3 and $q = 2$ Potts model.

6.6.3 Correlations

Successive configurations drawn from the prior as described in chapter 6.6 will show correlations to a certain extent. Hence we determine the correlation times of NESA for the 3×3 system for $q = 2$ and $q = 5$. The correlation has been analysed for a single walker for two cases:

- First no constraint is applied on the number of bonds (therefore on the likelihood) and so the walker can move through all bond configurations.
- Second the movement underlies a constraint that just configurations with increasing number of bonds are allowed (This corresponds to the case $\beta J > \log(2)$ as described in section 6.6.1).

The time series for the analysis are obtained as follows. The walker starts a 100 times from a fixed bond configuration and it moves for 100 steps. Then the configuration is released for the same amount of steps (100×100) and fixed again. This procedure is performed for 1000 fixed configurations. For the measurements, starting from a fixed configuration, the resulting autocorrelation functions are averaged. The integrated correlation time is then evaluated the same way as described in section 4.1.1.
The histograms of the resulting 1000 mean correlation times τ_i are shown in figure 6.17. Table 6.6.3 lists the integrated correlation times with their corresponding standard deviation for the above mentioned cases. The numbering (a) - (d) indicates the related picture in figure 6.17.

Gridsize 3x3	$q = 2$	$q = 5$
no constraint	(a) **1.72 ± 0.05**	(c) **1.85 ± 0.21**
constraint	(b) **1.54 ± 0.26**	(d) **1.71 ± 0.03**

Table 6.1: Integrated correlation times $\bar{\tau}_{int}$. See text for further explanation.

Figure 6.18 shows the same distribution of correlation times for a 16×16 system with $q = 10$ colours, evaluated in the same way as described above. The likelihood has been constrained during this simulation. The histogram indicates no increase in correlation time for larger systems.

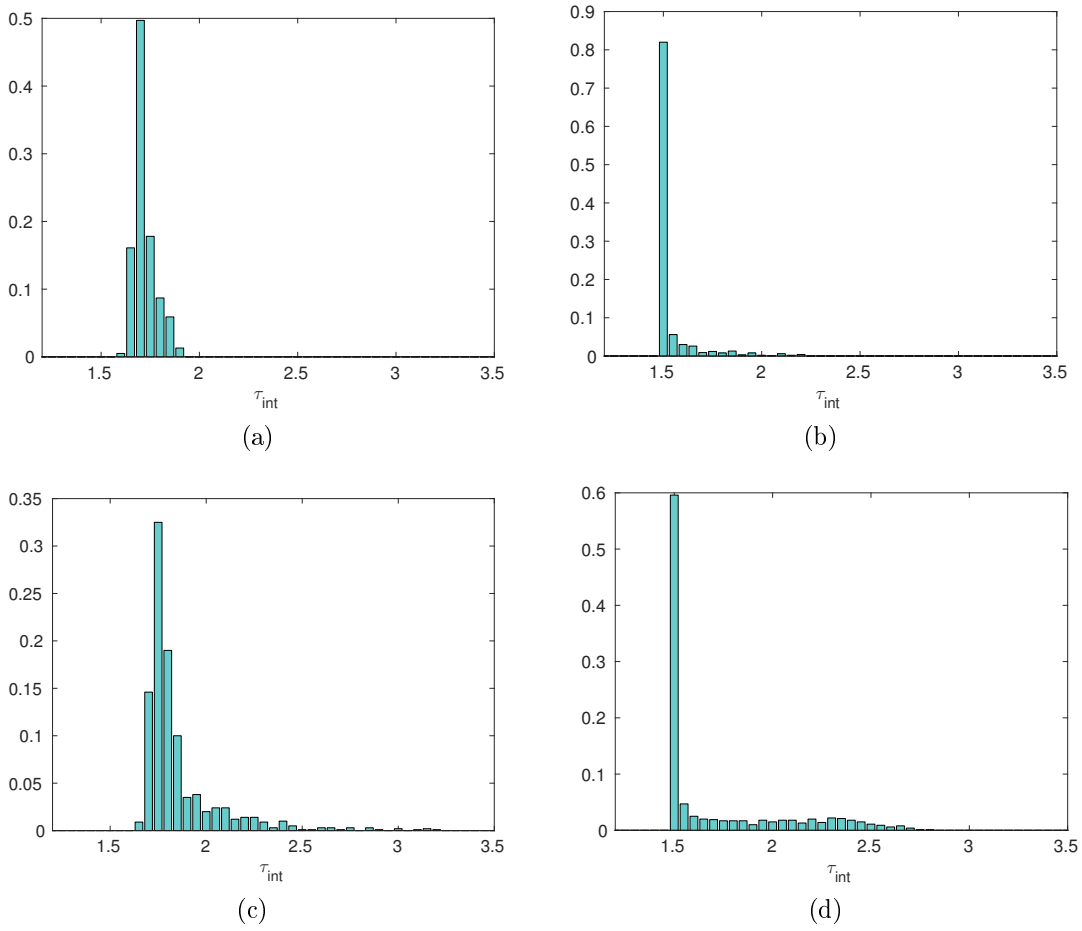


Figure 6.17: Distributions of the integrated correlation time for a 3×3 system with: (a) $q = 2$, not constrained, (b) $q = 2$ constrained, (c) $q = 5$ not constrained and (d) $q = 5$ constrained

This observation is supported by figure 6.19, where the mean $ln(Z)$ of 10 independent runs is plotted over the number of SW updates between measurements. The greyish error bar indicates the standard deviation of the 10 runs, whereas the turquoise one indicates the mean error calculated from 100 prior samples. The values do not exhibit any significant convergence to the true value with increasing number of SW updates, nor any reduction in the error values.

The low values for the correlation times $\bar{\tau}_{corr}$ give reason to assume that the successive states of a walker during the evolution of the simulation are not significantly correlated. Therefore it is not necessary to discard measurements of the likelihood in order to get unbiased estimates.

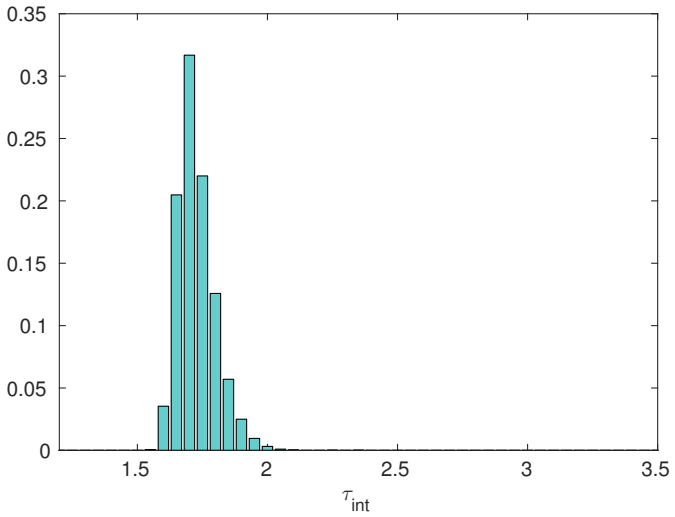


Figure 6.18: Correlation times of a 16×16 Potts model with $q = 10$ and a constraint on the likelihood.

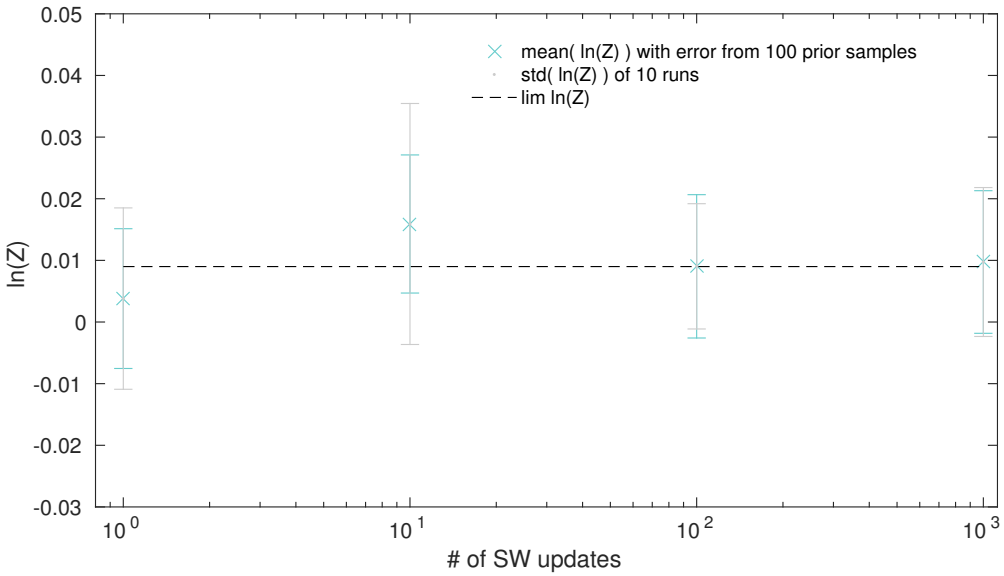


Figure 6.19: $ln(Z)$ for the 16×16 system with standard deviation from 10 runs and the error evaluated from 100 prior samples.

7 Results

First the nested sampling algorithm has been tested for a simple illustrative problem - a one dimensional Gaussian likelihood function. This gives us the possibility to study the behaviour of the algorithm, while having all quantities analytically available.
Further the evaluation of the $q = 10$ Potts model partition function by means of NESA is analysed in more detail. Afterwards results for thermodynamic quantities, derivable from the partition function, are compared to the respective results from a multicanonical simulation.
The performance comparison of retrieving the partition function $ln(Z)$ via NESA and via TI reveals a distinct advantage for NESA. Finally the investigation of the parallel implementation of the NESA algorithm yields a potential way to reduce runtime.

7.1 One dimensional Gaussian likelihood function

Starting again from our primary problem, evaluating the integral

$$Z = \int dx\, L(x)\, p(x) \tag{7.1}$$

we are now employing

$$\begin{aligned} L(x) &= exp(-\frac{x^2}{2}) \\ p(x) &= \text{uniform in } [-1, 1] \end{aligned} \tag{7.2}$$

a Gaussian likelihood and a uniform prior as the components of our integrand. Utilising such a simple likelihood function gives us the possibility to test our implementation of the nested sampling algorithm. For this toy problem all steps can be compared to analytical results, because we know the functional form $L(X)$.

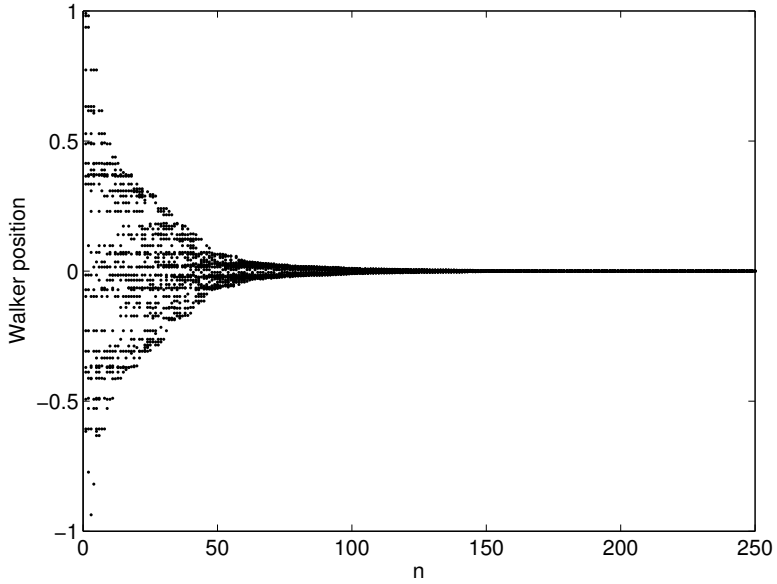
The analytical result for Z yields

$$Z = \frac{1}{Z_{pi}} \int_{-1}^{1} exp(-\frac{x^2}{2}) = \sqrt{\frac{\pi}{2}}\ \mathrm{erf}\left(\frac{1}{\sqrt{2}}\right) \approx 0.8556$$

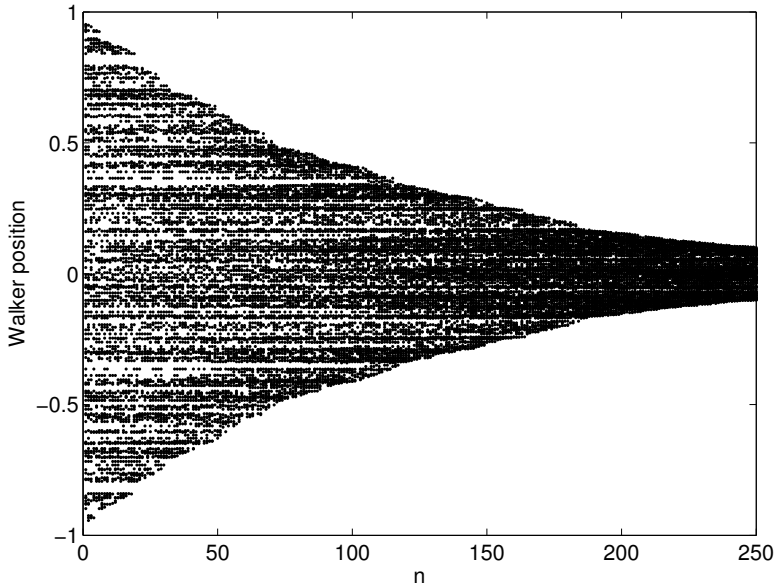
with the prior normalisation $Z_{pi} = 2$.

The application of the nested sampling algorithm begins with drawing the start sample of walkers from the uniform prior in the range of [-1,1]. The respective likelihood values can be computed directly from 7.2. We are now also able to retrieve the new parameter boundary x_{max} corresponding to λ_{min}, from the inverse of the likelihood function. The new x, for the walker with the minimal likelihood, is again uniformly drawn within $x \in [-x_{max}, x_{max}]$. The prior mass during each step is simply given by $X = 2\, x_{max}$.
The algorithm stops, when the variation of λ_{min} values falls beneath a specified threshold ϵ. Commonly the likelihood values are not increasing any more after a certain $log(X_i)$ is reached. Unfortunately in general one can not be sure that no further increase, hence further critical mass of the integral, will occur. The convergence criterion needs to be properly chosen for problems at hand.

The time evolution of the positions of the walkers is depicted in figure 7.1. Starting from positions in [-1,1], the increasing likelihood threshold forces the walkers more and more into the peak of our likelihood function. Fewer walkers achieve faster shrinkage therefore one needs fewer steps to reach the peak. Although using more walkers gives a better sampling of the phase space, which gets important when one is dealing with a multiple peaked likelihood function.



(a)



(b)

Figure 7.1: Positions of walkers for each step n during a single nested sampling run for the Gaussian likelihood function and the uniform prior. In the upper figure a sample of 20 walkers was used, in the lower a sample of 100.

Knowing the true prior mass values $X = 2\, x_{max}$, we can compare them to the X calculated from the Beta distribution of the shrinkage. Figure 7.2 shows the likelihood values, retrieved from a single nested sampling run, in dependence of their true X value and also in dependence of three samples of X sequences. The likelihood values have been calculated with a sample of 10 walkers. The first 25 update steps of the algorithm are shown. The curves computed with the sampled prior masses exhibit a similar shape as the one computed at the true X values, but with the difference of a random offset on the abscissa.

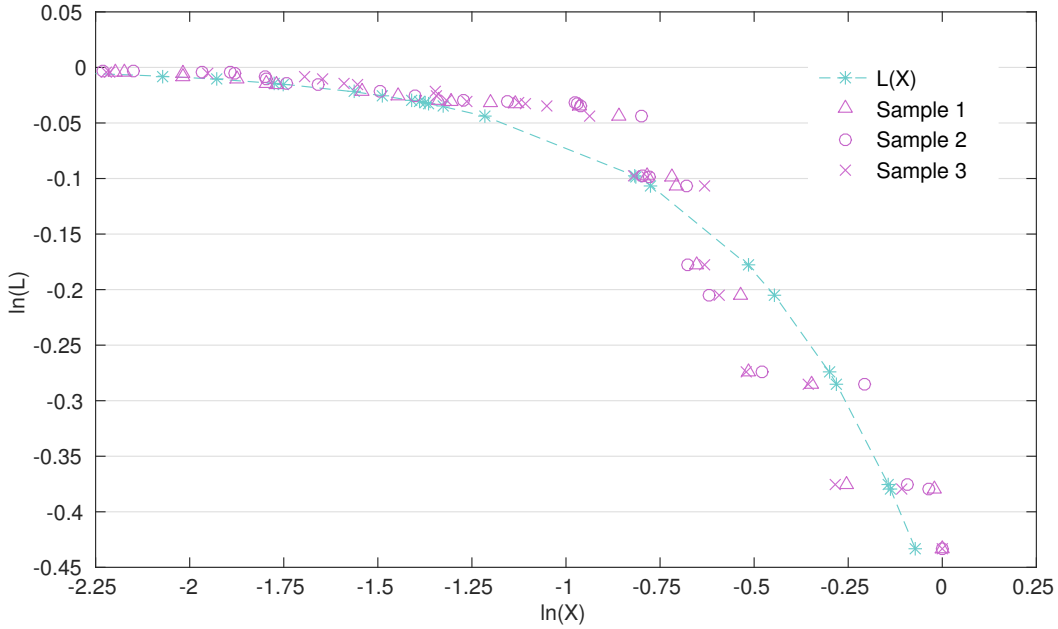


Figure 7.2: Likelihood values from a single NESA run with 10 walkers placed at their true X value and placed at three samples of prior sequences.

In figure 7.3 $ln(Z)$ is evaluated for different sets of parameters. In each subfigure the results of 20 independent nested sampling runs are displayed. The left column gives the results for the lower bound of the estimate of $ln(Z)$. Respectively the upper bound can be found in the right column.
Down the column k, the number of walkers, which are discarded in each update, is successively increased. For $k = 2$ and $k = 4$ the parallel implementation of the algorithm has been employed. The number of walkers K is scaled according to the increase of the parallel updates per step k (equation 6.39).
The analytic result for $ln(Z)$ (see equation 7.3) is depicted by the dotted black line.

The magenta triangles indicate the values of $ln(Z)$, where each of the respective likelihood sequences has been computed at the true X values. The turquoise crosses indicate the mean value of 100 $ln(Z)$ values stemming from 100 samples of prior mass sequences. The horizontal dashed lines in the respective colour indicate the mean values of the 20 independent runs. The values evaluated by the prior samples fluctuate around the true ones in a random fashion, hence the mean values of both variants show good agreement with each other.
The plotted error bars are evaluated as in equation 6.33, by means of the multiple samples of the prior masses. The errors conform to the observed fluctuations of $ln(Z)$, evaluated via the prior mass samples, around $ln(Z)$, evaluated at the true prior mass values.

The results for $k = 1$ agree within the error bars with the analytic result. For $k = 2$ and $k = 4$ the results differ in increasing fashion from the analytic result. Furthermore the error bars display a decrease with increasing number of walkers K. Remembering equation 6.36,

$$ln(Z) \leq ln(Z_{\text{analytic}}) \leq ln\left(\frac{K+1}{K-k+1}\right) + ln(Z)$$

and employing it for the depicted parameters, one finds that, the increasing gap between the lower and the upper bound can be explained by means of the quadrature error, which is not considered in the evaluation of the indicated errorbars.

Further we are interested how the number of walkers K affects the outcome of the computations. Figure 7.4 displays $ln(Z)$ versus the number of employed walkers K. Here the symbols refer to the same quantities as in the previous figure. With increasing K the results converge to the analytic value. The error bars show the expected $1/\sqrt{K}$ decline. This is in addition qualitatively emphasized in figure 7.5. The higher accuracy of course leads to longer computation times. The CPU times displayed in figure 7.6, roughly match a cubic increase over K. Also the quadrature error in equation 6.36 decreases with larger K.

At the end of this section we are able to conclude that our program for calculating the likelihood sequence, in the standard and the parallel implementation, as well as the program for evaluating the likelihood sequence via the sampled prior masses leads to proper results for the illustrative Gaussian likelihood function at hand.

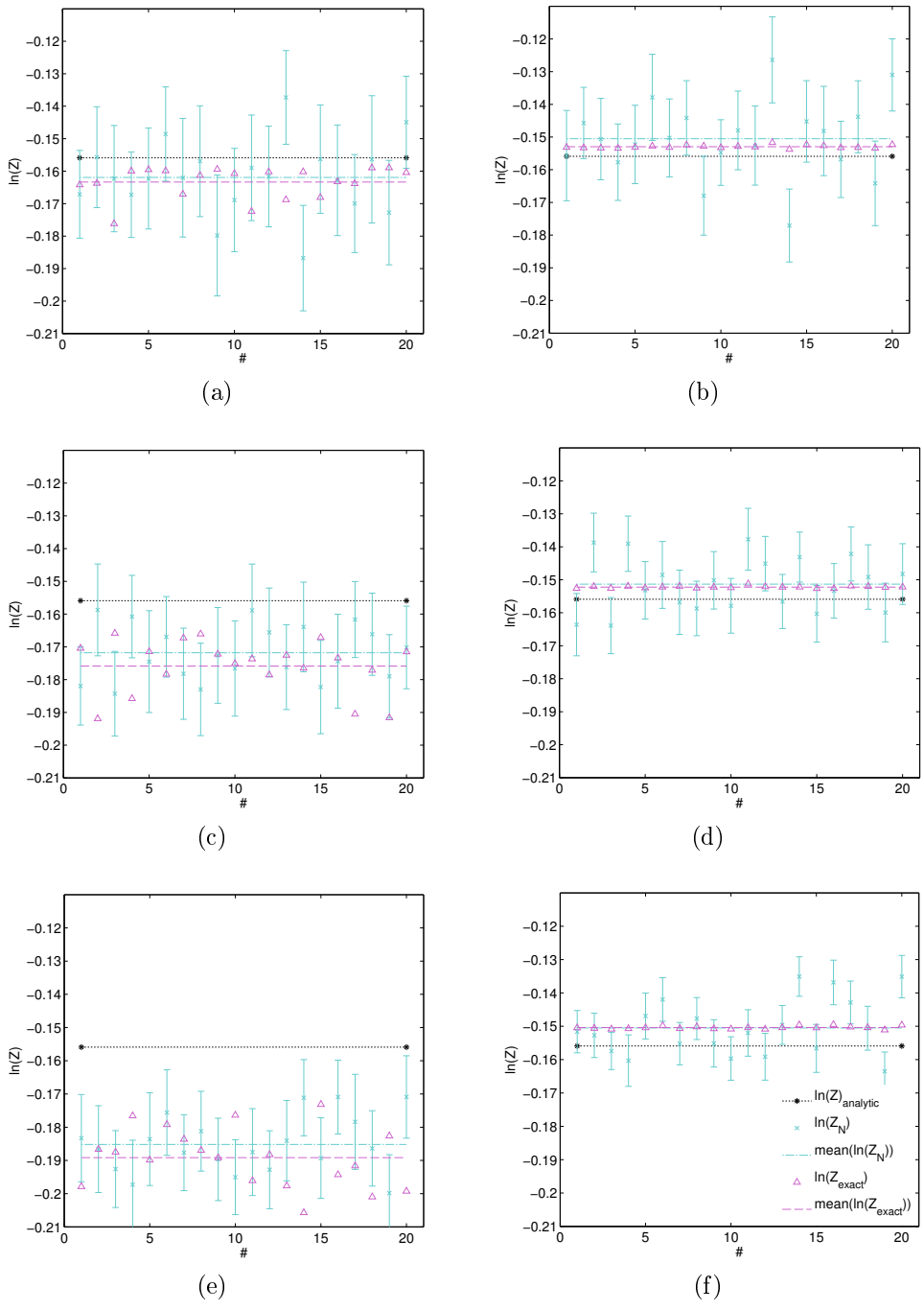


Figure 7.3: $ln(Z)$ results of 20 independent nested sampling runs: (a),(b) K=100, k=1; (c),(d) K=200, k=4; (e),(f) K=346, k=12. The left/right column shows the results for the lower/upper bound. The magenta triangles indicate $ln(Z)$ computed at the true X values. The turquoise crosses indicate the mean of 100 $ln(Z)$ values stemming from 100 prior mass samples. The horizontal dashed lines in the respective colour indicate the mean values of the 20 independent runs. The analytic result for $ln(Z)$ is depicted by the dotted black line.

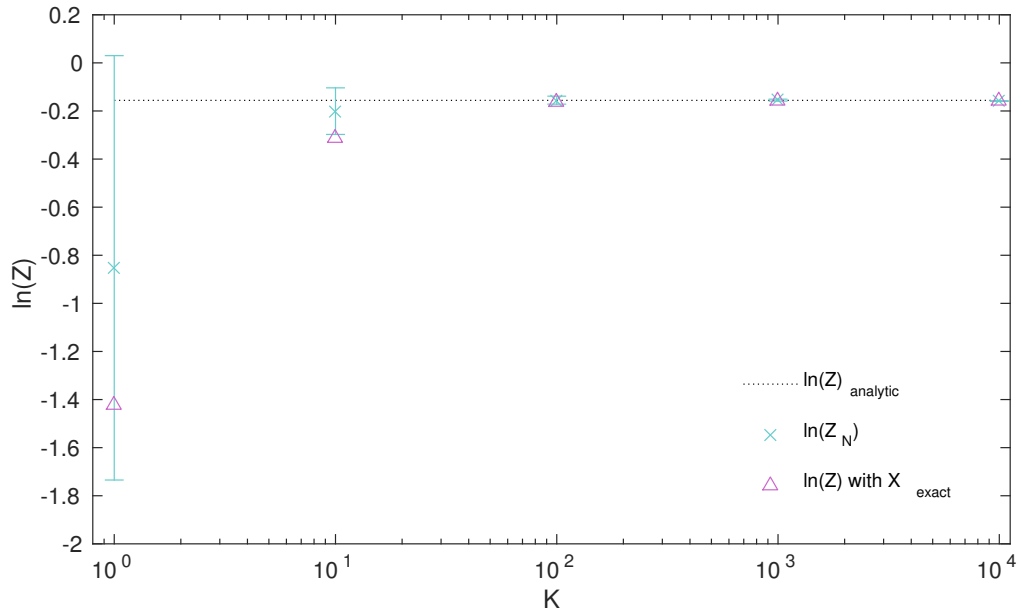


Figure 7.4: $ln(Z)$ of the Gaussian likelihood function in dependence of the number of walkers K for $k = 1$. The analytic result for $ln(Z)$ (see equation 7.3) is depicted by the dotted black line. The magenta triangles indicate $ln(Z)$ computed at the true X values. The turquoise crosses indicate the mean value of 100 $ln(Z)$ values stemming from 100 samples of prior mass sequences.

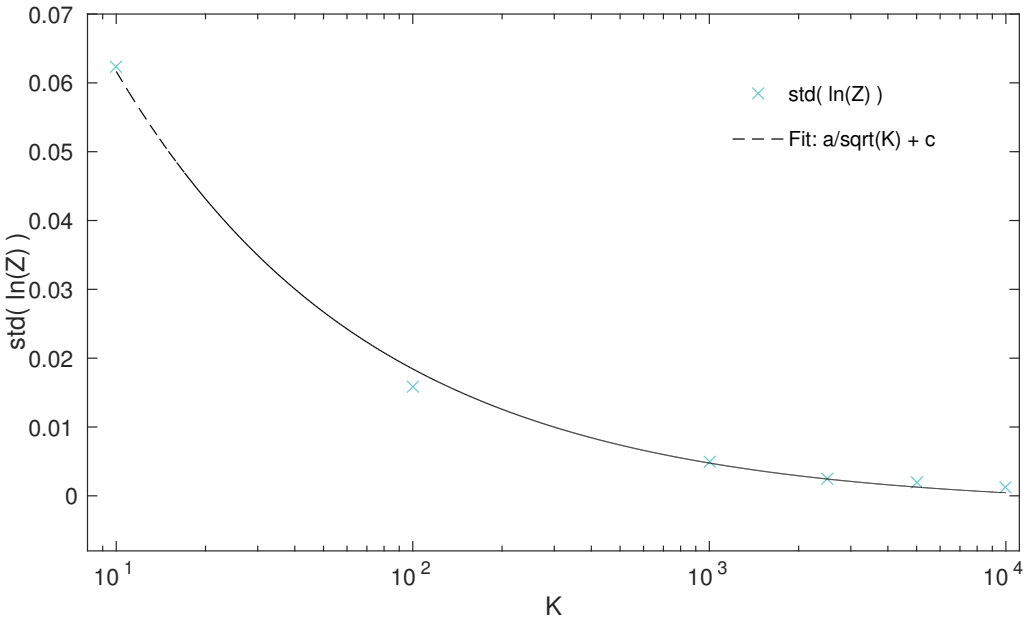


Figure 7.5: Standard deviation of $ln(Z)$, indicated by the errorbars in the previous figure, in dependence of the number of walkers K.

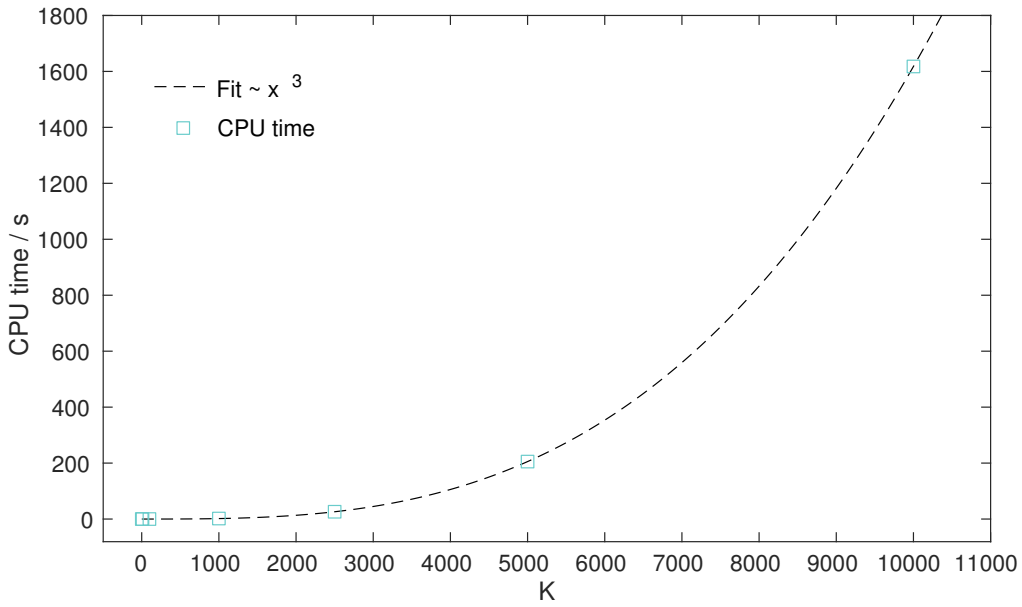


Figure 7.6: The increase of CPU time over increasing number of walkers K fitted by a cubic function.

7.2 Potts model with 10 colours

In this section we analyse the computation of the partition function $ln(Z)$ for Potts models in more detail. In the beginning we focus on a 4×4 system. A single nested sampling run, using K=100 walkers, gives us the likelihood sequence used for the further calculations. 100 samples of the prior mass are produced to compute the mean and the variance of the partition function (see section 6).

First we want to understand how the number of nested sampling steps influences the outcome of the evaluation.

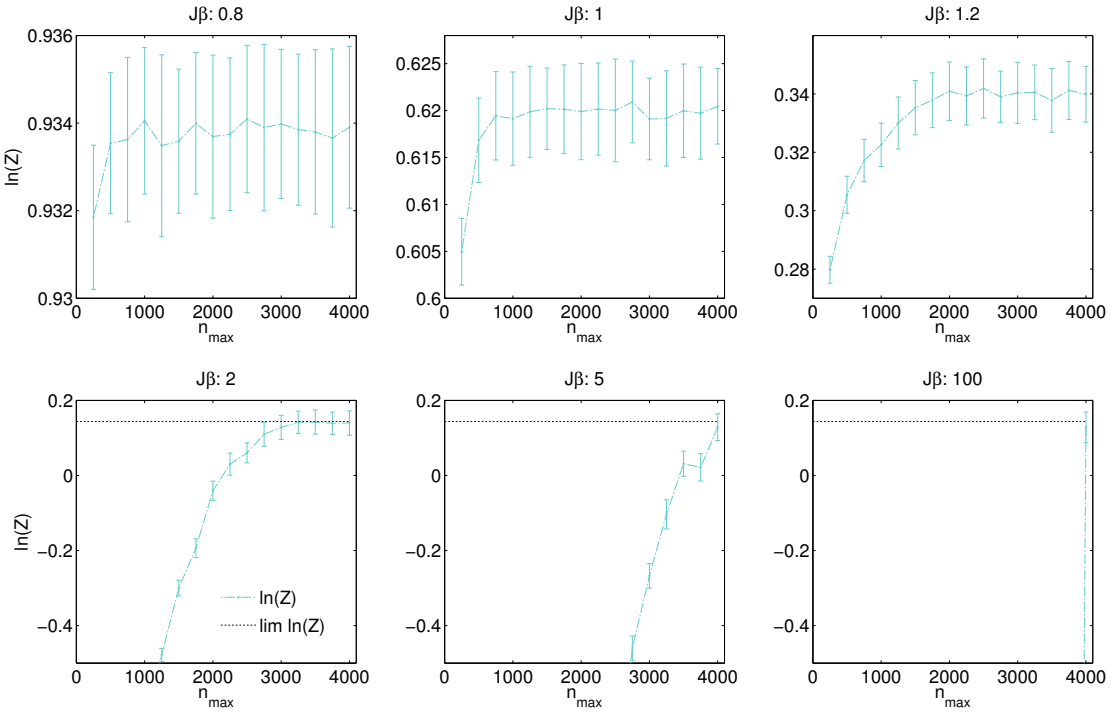


Figure 7.7: The values for $ln(Z)$ for a 4×4, q=10 Potts model are displayed in dependence on the number of NESA steps n_{max} used for their evaluation. For increasing values of $J\beta$ the n_{max} dependence of $\ln(Z)$ becomes important, which means that for larger $J\beta$-values the values of $ln(Z)$ need more steps to converge to a constant value. The lower row of plots shows that for values of $J\beta$ way above the phase transition, all likelihood values need to be considered, to yield the known limiting value of $ln(Z)$ (indicated by the black dotted line). Note that the scale on the y-axis changes in the upper row.

Remember that the likelihood value of the Potts model, as defined in 6.45, reaches a maximum if all bonds are active. The required number of steps is strongly

influenced by the value of $J\beta$, for which we want to determine $ln(Z)$. Figure 7.7 shows the results for $ln(Z)$ for different $J\beta$ values in dependence of n_{max}. n_{max} here denotes the number of NESA steps, corresponding to the number of likelihood values, used to compute $ln(Z)$.
At low values of $J\beta$, already half (for $J\beta = 1.2$), or even a quarter (for $J\beta \leq 1.0$) of the maximal number of steps, yield converged results. For larger $J\beta$ values, where we can compare the results with the limiting $ln(Z)$ value (indicated by the black dotted line), only for large n_{max} values the correct results are obtained. Hence we find that for increasing $J\beta$-values the number of required NESA steps, to obtain correct results, increases. For very high $J\beta$ values (e.g. $J\beta = 100$), an accurate estimate for $ln(Z)$ is only available, if the whole possible sequence, including the likelihood values up to $D = 2N$, is considered.

The $J\beta$ dependence of the partition function stems from the different shapes of $L(X)$. Figure 7.8 displays in (a) $\ln(L)$ in dependence of $\ln(X)$. The stairplot should emphasize that for the Potts model L can just take on discrete values for various values of $J\beta$. Here the values have been shifted by $2NJ\beta$, so they all start out at $ln(L) = 0$ and crossing is avoided. One should not be deluded by the slight slope of the logarithmic values.
In (b) and (c) L is plotted over X, where X is displayed on a logarithmic scale, respectively for $J\beta = 0.8$ and $J\beta = 2$. These two plots display the extreme narrowing and increase of the likelihood peak with increasing $J\beta$, which is already distinctive for this small system. The points, where $\ln(Z)$ has been evaluated for the respective plots in figure 7.7, are marked with an black x. One can observe in both figures that for $J\beta = 2$ in (c) the last terms contribute more to the sum for $\ln(Z)$ than the last terms for $J\beta = 0.8$ in (b).

This becomes evident when plotting the summands of Z over the number n_{max} of evaluated steps. For $J\beta = 0.8$ it is now obvious why the second $ln(Z)$ evaluation in the respective plot in figure 7.7 yields approximately the same result as the last evaluation. For $J\beta = 2$ the main contribution to the sum stems already from the last third of the sequence and for $J\beta = 5$ the last summand clearly dominates. Again this is in good agreement with the results in figure 7.7.

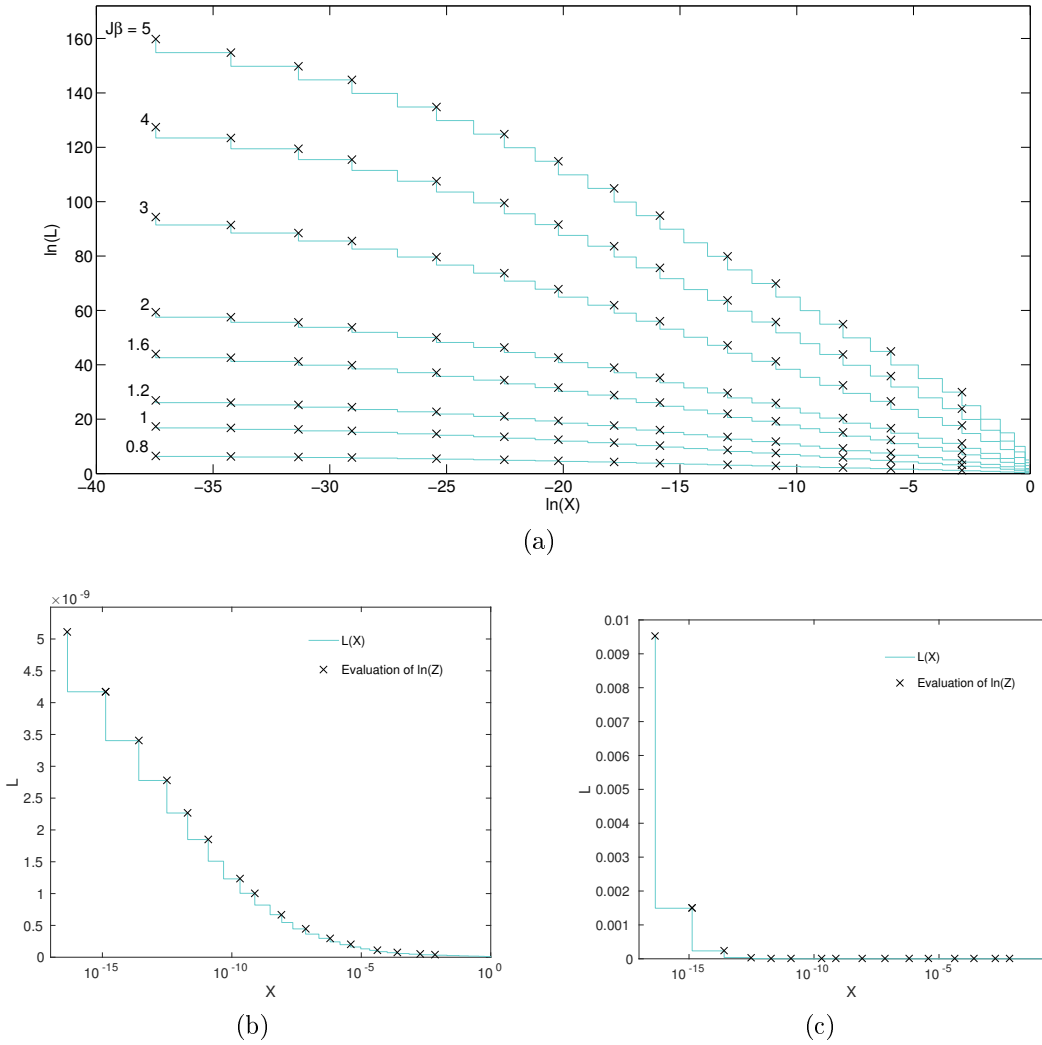


Figure 7.8: (a) Plot of the logarithmic likelihood over the logarithmic prior mass of the 4×4, $q = 10$ Potts system for different temperatures (b) Semilogarithmic plot of L(X) for $J\beta = 0.8$ (c) Semilogarithmic plot of L(X) for $J\beta = 2$.

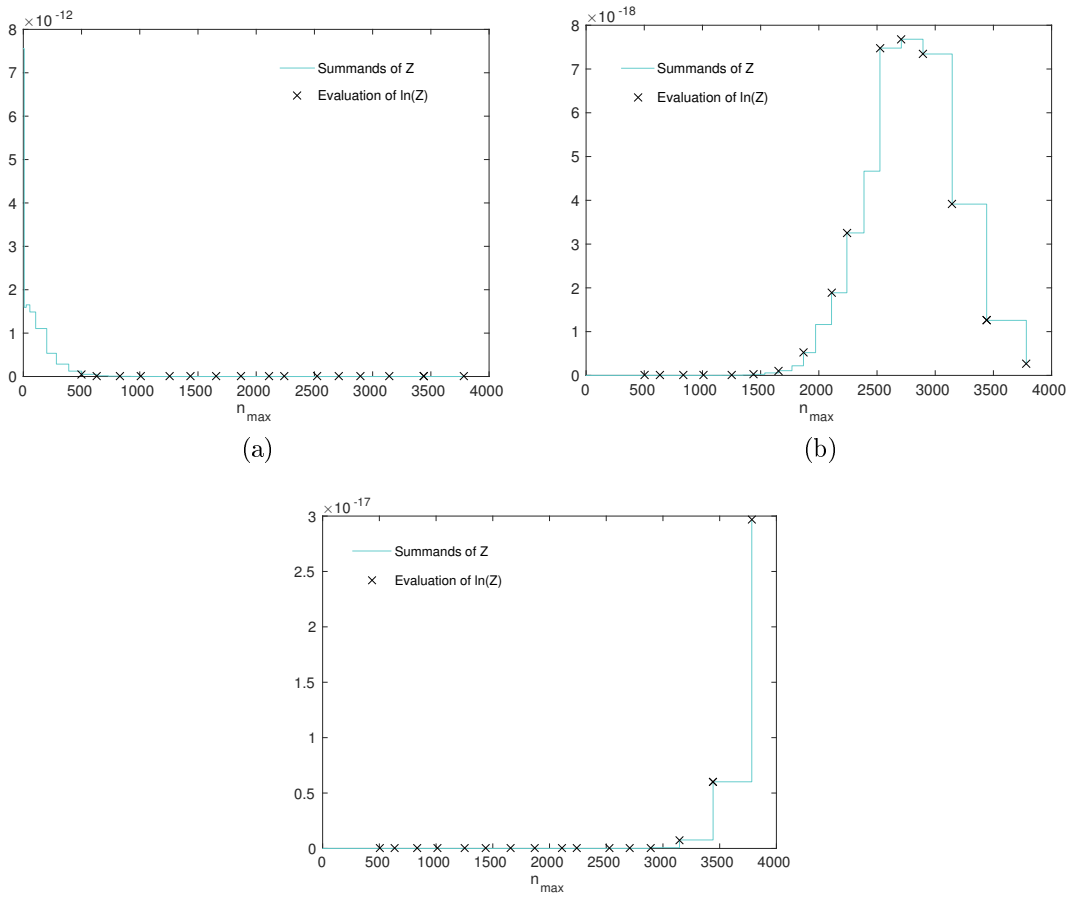


Figure 7.9: Summands of equation 4.6 for a 4×4 and $q = 10$ system at different temperatures: (a) $J\beta = 0.8$, (b) $J\beta = 2$, (c) $J\beta = 5$.

The above analysis applied to a 128×128 system yields in principal the same but more distinct behaviour. The results for $\ln(Z)$ in dependence on the number of NESA steps, as shown in figure 7.11, reveals that at high $J\beta$ values the importance of the last summand further increases for larger systems. Here the separate investigation of the exponential summands, as for the 4×4 system, is computationally not available any more, due to the occurrence of extremely large values.

Eventually we can conclude, that for the ability to evaluate $\ln(Z)$ properly up to large $J\beta$ values, it is required to calculate the whole likelihood sequence up to the maximum value, corresponding to $D = 2\,N$ set bonds.

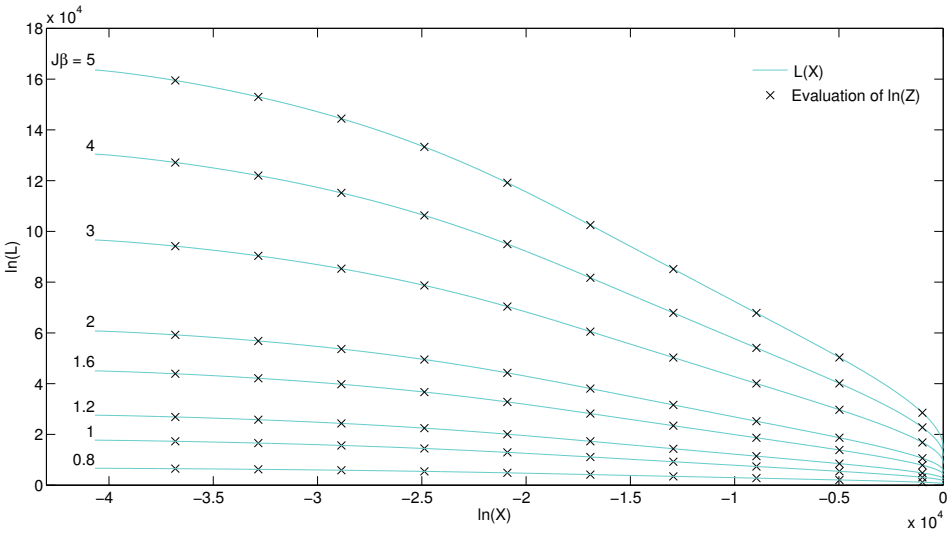


Figure 7.10: The values of the likelihood function over the prior mass fairly increase for the 128×128, $q = 10$ system, when comparing it to the $4x4$ system. Note the log scale on both axes.

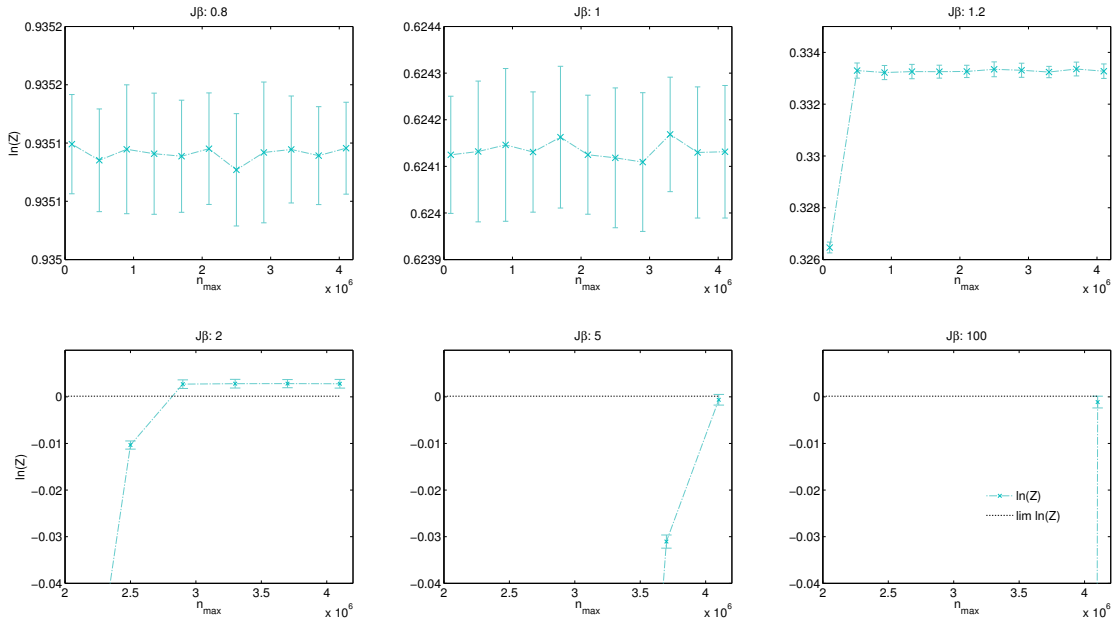


Figure 7.11: The values for $ln(Z)$ for a 128×128, q=10 Potts model are displayed in dependence on the number of NESA steps n_{max}. For large $J\beta$ values the whole likelihood sequence needs to be evaluated. The lower row of plots shows that for values of $J\beta$ way above the phase transition, all likelihood values need to be considered, to yield the known limiting value of $ln(Z)$ (indicated by the black dotted line) within the error bars. Note that the x-axis in the lower row only displays half the range of the upper.

Number of prior mass samples

Although the calculation of a prior mass sequence is computationally cheap in comparison to a nested sampling run, the influence of the number of prior mass samples is of interest.
Here we deal with a 32×32 system with 10 colours and evaluate $ln(Z)$ for a value of $J\beta = 10$. For the NESA run 100 walkers were employed. The dependence of $ln(Z)$ and the respective standard deviation on s, the number of prior sequences used for evaluation, is illustrated in figure 7.12. For each s the average of 100 repeated calculations of $ln(Z)$ is shown. Above $s = 50$, $ln(Z)$ as well as the standard deviation, stay fairly constant. Therefore for the following evaluations $s = 50$ sequences are sufficient for each computation of the partition function. At first glance it is surprising that the errors do not decrease with a larger number of prior samples. However, considering the standard deviation given in equation 6.33, one finds no dependence on s.

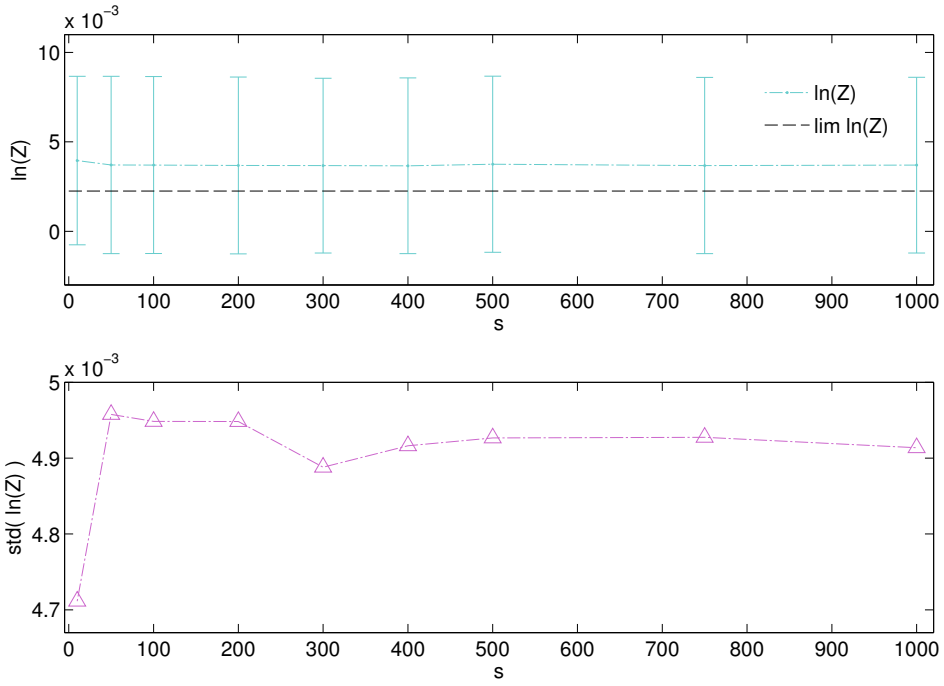


Figure 7.12: The number of prior sequences s has little influence on the evaluation of $ln(Z)$ (displayed here for a 32×32, $q = 10$ system at $J\beta = 10$).

Number of walkers

An increasing number of walkers will lead to convergence to the true value, with increasing accuracy. Figure 7.13 displays the results for $K = 10 - 10000$ and supports the expected $\frac{1}{\sqrt{K}}$ decay (as we found for the upper bound of the variance in equation 6.34).

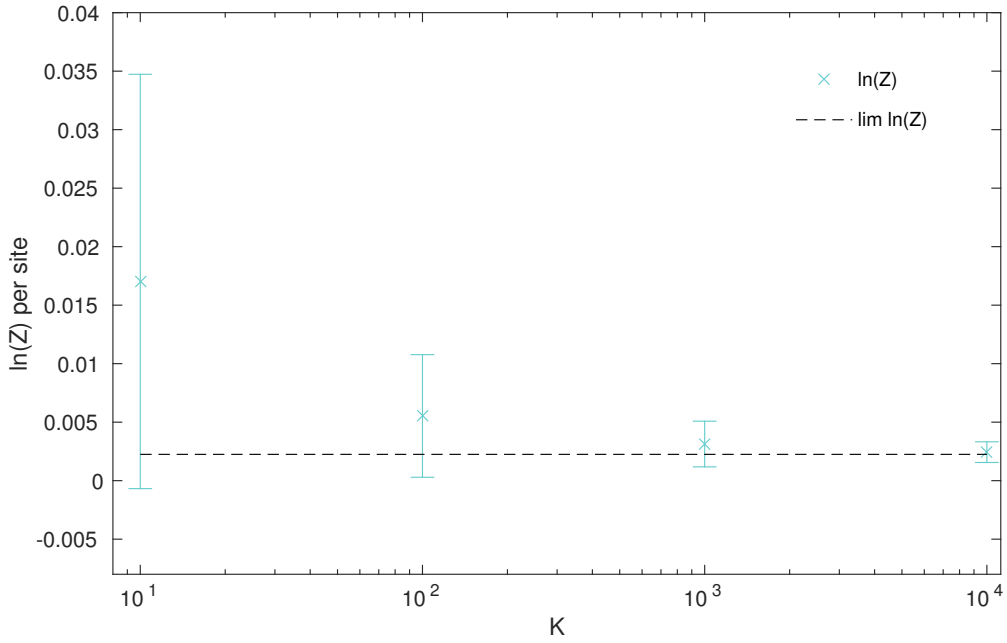


Figure 7.13: $ln(Z)$ of a 32×32 and q=10 system versus the number of walkers K. The limiting value of $ln(Z)$ for $J\beta \to \infty$ is indicated by the black dashed line.

Partition function dependence on $J\beta$

In statistical physics we are usually interested in the dependence of the partition function on the inverse temperature. In the canonical ensemble we this provides the whole thermodynamic information of the system.
Figure 7.14 and 7.15 display $ln(Z)$ per site versus $J\beta$ for grid sizes 16×16 and 128×128. Further depicted are the phase transition temperature and the limiting value of $ln(Z)$. For $J\beta$ values above the critical $J\beta$, $ln(Z)$ quickly approaches the limiting value $ln(q)/N$.

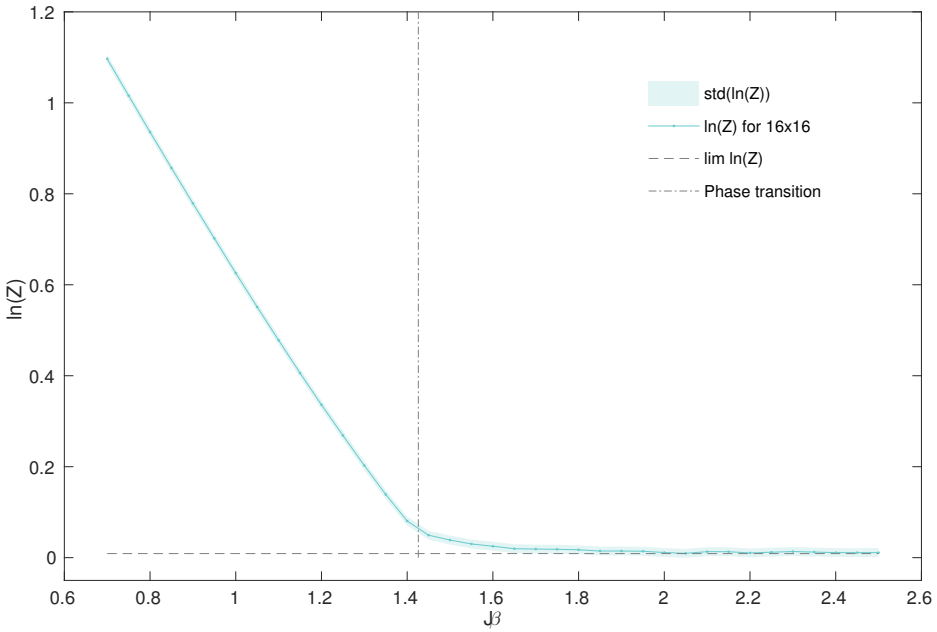


Figure 7.14: $ln(Z)$ of a 16×16, q=10 system versus $J\beta$ evaluated via NESA with $K = 100$. For $J\beta$ values larger than 2, $ln(Z)$ approaches the limiting value of $ln(Z)$ for $J\beta \to \infty$ (indicated by the black dashed line).

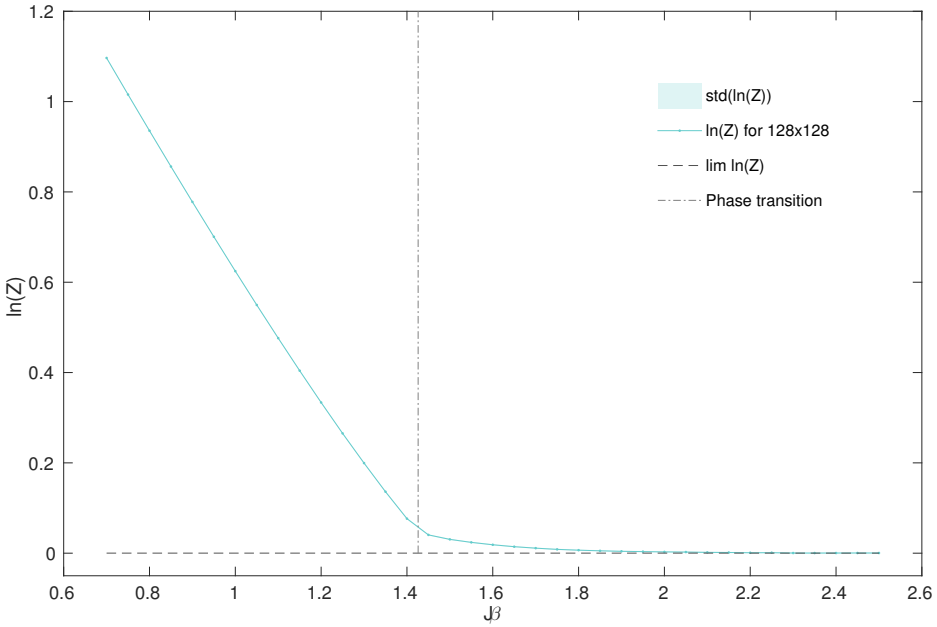


Figure 7.15: $ln(Z)$ of a 128×128, q=10 system versus $J\beta$ evaluated via NESA with $K = 100$. For $J\beta$ values larger than 2, $ln(Z)$ approaches the limiting value of $ln(Z)$ for $J\beta \to \infty$ (indicated by the black dashed line).

7.3 Thermodynamic quantities

Given the partition function of a system in the canonical ensemble, thermodynamic quantities like the free energy F and the inner energy U as well as the entropy S and heat capacity c_V can be deduced. The definitions of these quantities read as:

$$F = -\frac{1}{\beta} \ln(Z) \tag{7.3}$$

$$U = -\frac{\partial}{\partial\beta} \ln(Z) \tag{7.4}$$

$$S = \frac{U - F}{T} \tag{7.5}$$

$$c_V = \frac{\partial}{\partial T} U = k_B \beta^2 \frac{\partial^2}{\partial\beta^2} \ln(Z) \tag{7.6}$$

Starting from the definition of the estimate of the partition function, we deduce expressions for the first and second derivation of $ln(Z)$. Given this expressions we can evaluate the above physical quantities analytically employing the obtained likelihood sequence and samples of prior masses. Hence we can avoid the determination of the numerical derivatives and their associated errors.
Remembering equation 6.45 we get for $\ln(L)$

$$\ln(L) = D \ln(\exp(J\,\beta) - 1) \tag{7.7}$$

and the derivation of $\ln(L)$ with respect to β yields

$$\frac{\partial}{\partial\beta} \ln(L) = D \frac{J \exp(J\,\beta)}{\exp(J\,\beta) - 1} \tag{7.8}$$

Given $\ln(L_n)$ and $\ln(\Delta X_n)$ we can calculate the logarithm of the partition as follows

$$\ln(Z) = \ln\left(\sum_n \exp[\ln(L_n) + \ln(\Delta X_n)]\right) \tag{7.9}$$

Here it is beneficial to use the first definition of $\ln(Z)$ in 6.10, where L_n occurs only once in each summand and therefore simplifies following derivatives with respect to β.

In the following an expression for the first and second derivative of the partition

function in terms of the number of bonds D_n, the likelihood L_n and the prior masses X_n is given.

$$\begin{aligned} \frac{\partial}{\partial\beta}\,\ln(Z) &= \frac{\sum_n \exp\left[\,\ln(\Delta X_n)\right]\,\frac{\partial}{\partial\beta}\exp\left[\,\ln(L_n)\,\right]}{\sum_n \exp\left[\,\ln(L_n)+\ln(\Delta X_n)\,\right]} \\ \text{with}\quad \frac{\partial}{\partial\beta}\exp(\,\ln(L_n)\,) &= \exp\left(\underbrace{D_n\,\ln\left[\exp(J\beta)-1\right]}_{\ln(L_n)}\right) D_n\,\frac{J\,\exp\left(J\,\beta\right)}{\exp\left(J\,\beta\right)-1} \end{aligned} \tag{7.10}$$

So eventually the first derivative reads as

$$\frac{\partial}{\partial\beta}\,\ln(Z) = \frac{J\,\exp\left(J\,\beta\right)}{\exp\left(J\,\beta\right)-1}\,\frac{\sum_n \overbrace{\exp\left[\,\ln(D_n)+\ln(L_n)+\,\ln(\Delta X_n)\,\right]}^{\Gamma_n}}{\sum_n \underbrace{\exp\left[\,\ln(L_n)+\ln(\Delta X_n)\,\right]}_{\Omega_n}} \tag{7.11}$$

Here Γ_n and Ω_n are introduced for enhancing further readability. With this definition the second derivation has the following form

$$\begin{aligned} \frac{\partial^2}{\partial\beta^2}\,\ln(Z) &= \left(\frac{J\,\exp\left(J\,\beta\right)}{\exp\left(J\,\beta\right)-1}-\left(\frac{J\,\exp\left(J\,\beta\right)}{\exp\left(J\,\beta\right)-1}\right)^2\right)\,\frac{\sum_n\Gamma_n}{\sum_n\Omega_n} \\ &+ \frac{J\,\exp\left(J\,\beta\right)}{\exp\left(J\,\beta\right)-1}\,\frac{\sum_n\Gamma_n\,\frac{\partial}{\partial\beta}\,\ln(L_n)}{\sum_n\Omega_n} \\ &- \frac{J\,\exp\left(J\,\beta\right)}{\exp\left(J\,\beta\right)-1}\,\frac{\sum_n\Gamma_i\,\sum_n\Omega_n\,\frac{\partial}{\partial\beta}\,\ln(L_n)}{(\sum_n\Omega_n)^2} \end{aligned}$$

Inserting equation 7.10 leads to

$$\begin{aligned} \frac{\partial^2}{\partial\beta^2}\,\ln(Z) &= \left(\frac{J\,\exp\left(J\,\beta\right)}{\exp\left(J\,\beta\right)-1}-\left(\frac{J\,\exp\left(J\,\beta\right)}{\exp\left(J\,\beta\right)-1}\right)^2\right)\,\frac{\sum_n\Gamma_n}{\sum_n\Omega_n} \\ &+ \left(\frac{J\,\exp J\,\beta)}{(\exp\left(J\,\beta\right)-1}\right)^2\,\frac{\sum_i\Gamma_n\,D_n}{\sum_n\Omega_n} \\ &- \left(\frac{J\,\exp J\,\beta)}{(\exp\left(J\,\beta\right)-1}\right)^2\,\left(\frac{\sum_n\Gamma_n}{\sum_n\Omega_n}\right)^2 \end{aligned} \tag{7.12}$$

In both derivatives sums of exponentials occur. For an accurate calculation we refer once again to section 9.1.

7.3.1 Comparison of multicanonical sampling and nested sampling

The result of the energy per site, calculated according to 7.4 and 7.11, is compared to the respective result in [5]. The evaluation is performed for a 20×20 system with $q = 10$ colours. The results of the paper are reproduced by the Fortran code, provided by the author. The code has been downloaded from `www.hep.fsu.edu/~berg`. Details on the link and the usage of the code are explained in [5].

One needs to be aware of the definition of the energy, when comparing the results of the MUCA simulation with our nested sampling simulation. The definition of the energy per spin in [5] exhibits a factor 2 larger energy with a constant shift of $2/q$, in comparison to the definition used in this work.

The number of walkers for the NESA algorithm as well as the parameters for the TI were tuned to yield approximately the same total computation time as the multicanonical simulation (ca. 100 sec.). Therefore for nested sampling we employed 500 walkers and for TI we set the number of β steps to 250 and the number of SW updates to 1250 (here the assumed mean correlation time of $\tau_{int} = 2$ lead to a total number of SW updates of 5000).

The results of the energy for different values of $J\beta$ computed via nested sampling and via a multicanonical simulation are shown in figure 7.16. They exhibit a quite good agreement, though, the error bars of the multicanonical simulation are about a factor 10 smaller than the ones obtained by nested sampling. Although this crude comparison favours the multicanonical simulation, one should consider that the code of Berg will already be fairly optimized in terms of computational efficiency. Therefore further improvement of the code for NESA could yield improvement in runtime. Secondly the system investigated, is still rather small hence a comparison of the scaling with grid size of both methods would be of further interest. Moreover the multicanonical simulation in the employed implementation samples new states by means of the standard Metropolis algorithm. A more competitive approach, applying SW updates, would be the so called multibondic sampling algorithm introduced by Janke [12].

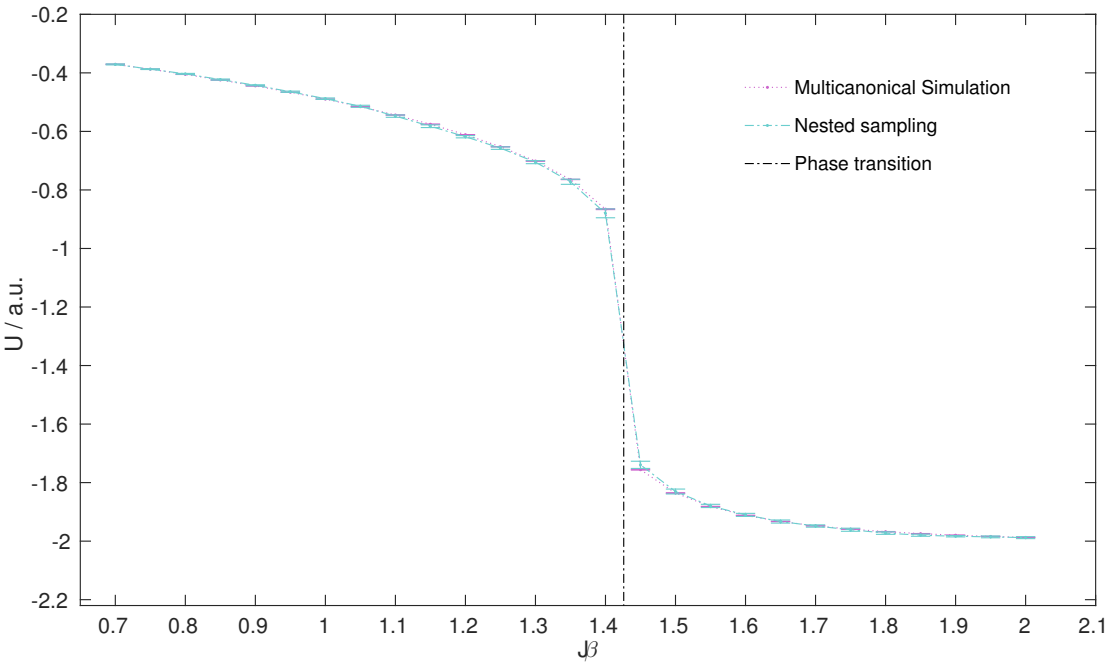


Figure 7.16: Energy per site versus $J\beta$ evaluated by means of NESA and MUCA for the 20×20 and $q = 10$ Potts system.

7.4 Performance comparison of thermodynamic integration and nested sampling

In table 7.1 our results are compared to the results of table 1 in Murray et al. [18]. The employed annealed importance sampling algorithm (AIS) is bound to fail, because using only one SW update per β value, will, especially for larger systems, give strongly correlated results (see 4.1). This leads to the strongly deviating results in [18]. Although, working for the presented problem in table 7.1, TI encounters massive slowing down if the system size increases as we found in section 4.1.1.

Table 7.1: $\ln(Z)$ for a 16×16 model computed by TI for parameters near phase transition. The second row shows the results from [18].

Method	q=2, $J\beta = 1$	q=10, $J\beta = 1.477$
Swendsen Wang TI	7.4 ±1.5	11.4 ±1.6
Swendsen Wang AIS	7.4 ±0.1	1.2

First we analyse the required time of the NESA algorithm to compute the partition

function of various grid sizes N. Having $\ln(Z_\pi)$ from a TI run and the sequence of likelihood values from a *single* NESA run at hand, one is able to calculate $\ln(Z)$ for *all* $J\beta$ values greater than $\ln(2)$. The importance of this point should be emphasized, because the most time consuming part are the SW updates during the NESA run, whereas the subsequent evaluation of $ln(Z)$ is computationally cheap. Lower values would also be possible to retrieve from the corresponding likelihood sequence, but because of the required previous thermodynamic integration, these values are already known.

In figure 7.17 one finds the required CPU time for NESA in dependence of the system size (from 4×4 up to 256×256). The total time as well as the time spent for the preceding TI run and the following nested sampling run are indicated separately. The TI was computed with 250 β steps and 1000 SW updates per step. The number of updates has been corrected by a correlation time of $\tau_{int} = 2$ (therefore we run 4000 SW updates) for all β values. For this parameter setting the error of $ln(Z_\pi)$ finally accounts for about $4 - 5\%$ of the error we obtain for $ln(Z_N)$ from NESA.
While the total TI runtime scales linearly (hence the time per site stays constant), we identify a power law in the increase of the total NESA runtime. From a linear fit ($y = a\,x + b$) of $ln(t_{\mathrm{NESA}})$ versus $ln(N)$ we obtain

$$a = \mathbf{1.96} \pm \mathbf{0.03} \quad \text{and} \quad b = -\mathbf{9.63} \pm \mathbf{0.25} \tag{7.13}$$

For small systems the time needed for the TI computation prevails. For larger systems, though, the power law increase of the NESA algorithm dominates. As reference point for the total time, the computation of the 256×256 grid with NESA took about 57h.
Figure 7.18 shows the results for $ln(Z)$ per site at $J\beta = 10$ evaluated via NESA. The error bars are in good agreement with the indicated limiting value of $ln(Z)$ for low temperature.

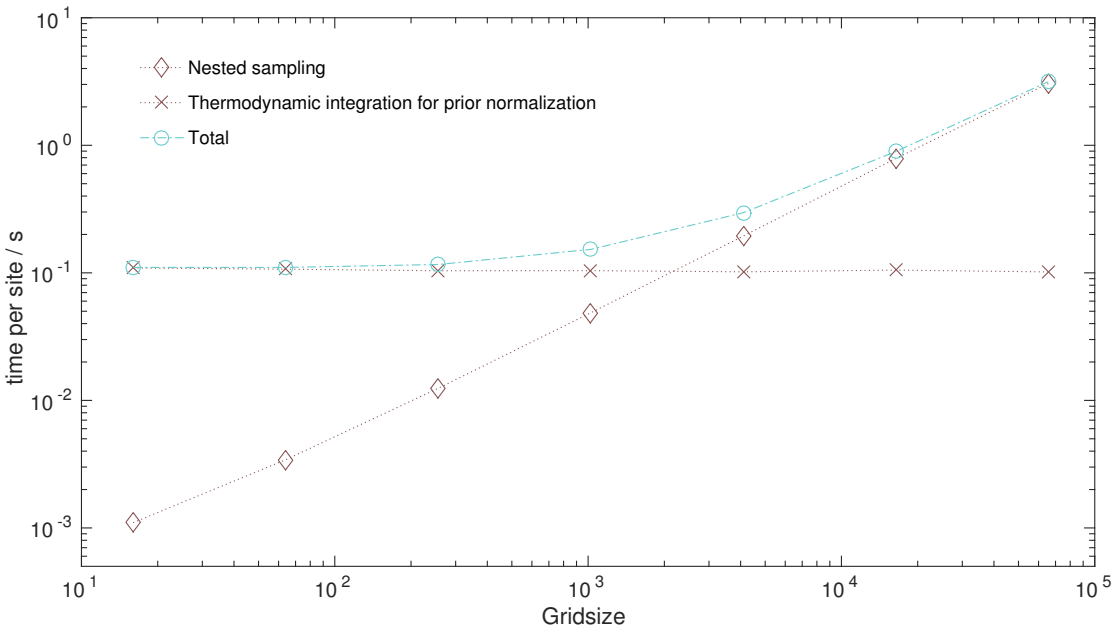


Figure 7.17: Breakdown of the total CPU time versus N for the nested sampling evaluation of the $q = 10$ Potts model.

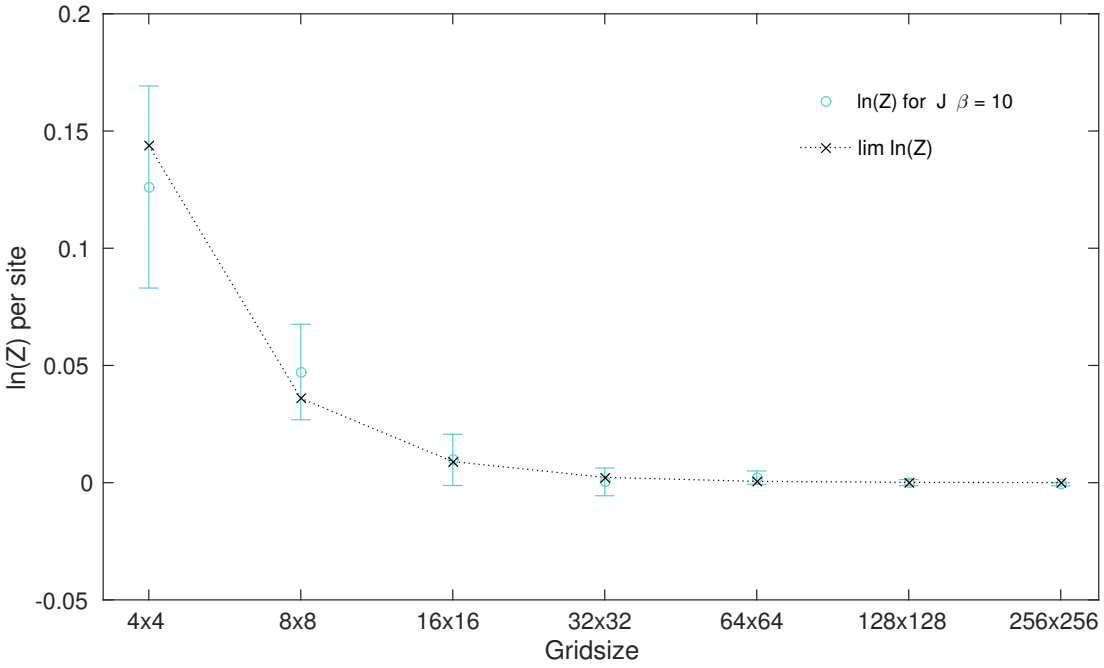


Figure 7.18: $ln(Z)$ for the $q = 10$ Potts model computed by the nested sampling algorithm versus grid size N.

For a proper comparison of the two methods, NESA and TI, some important points need to be considered. Thermodynamic integration yields the values of the partition function only at the values of the $J\beta$ steps explicitly determined in the start parameters. But looking at figure 4.1 and remembering equation 4.6, we see that only terms contribute to the sum, if not all bonds are active. The probability for a bond to be active is given in equation 2.9. It follows, that above a certain $J\beta$ value, there will be nearly no contributions to the sum any more. Hence it is sufficient to calculate the partition function $\ln(Z)$ with thermodynamic integration up to a value of e.g. $J\beta = 10$ for the investigation of the $q = 10$ system. Above $J\beta = 10$ no significant variation in the result of $\ln(Z)$ will occur.

The most time consuming element of both methods is the SW update. Remember that for nested sampling the number of required SW steps equals the number of likelihood values, whereas in thermodynamic integration for each $J\beta$ value, multiple SW updates are performed. Additionally to the plain SW update, nested sampling requires for each update the determination of a new bond number (see section 6.6.1) and further the minimal likelihood values after each update need to be ordered.

For the comparison of NESA and TI the respective parameters have been chosen in a way that the resulting error bars of $ln(Z)$ at $J\beta = 10$ are of approximately the same size (see figure 7.19). For system sizes ranging from 4×4 to 32×32 the correlation times are computed for all $J\beta$ values (details on the computation can be found in section 4.1.1). This enables us to correct the respective variances for TI at every $J\beta$ value. NESA used a sample of 100 walkers and 50 prior mass samples in all following computations.
The results for $ln(Z)$ computed via NESA and via TI are displayed in figure 7.19. The respective computational time required for TI as well as for NESA is given in figure 7.20. From a linear fit ($y = a\,x + b$) of the logarithmic values obtained for TI we find

$$a = \mathbf{1.94 \pm 0.07} \quad \text{and} \quad b = \mathbf{-2.83 \pm 0.37} \tag{7.14}$$

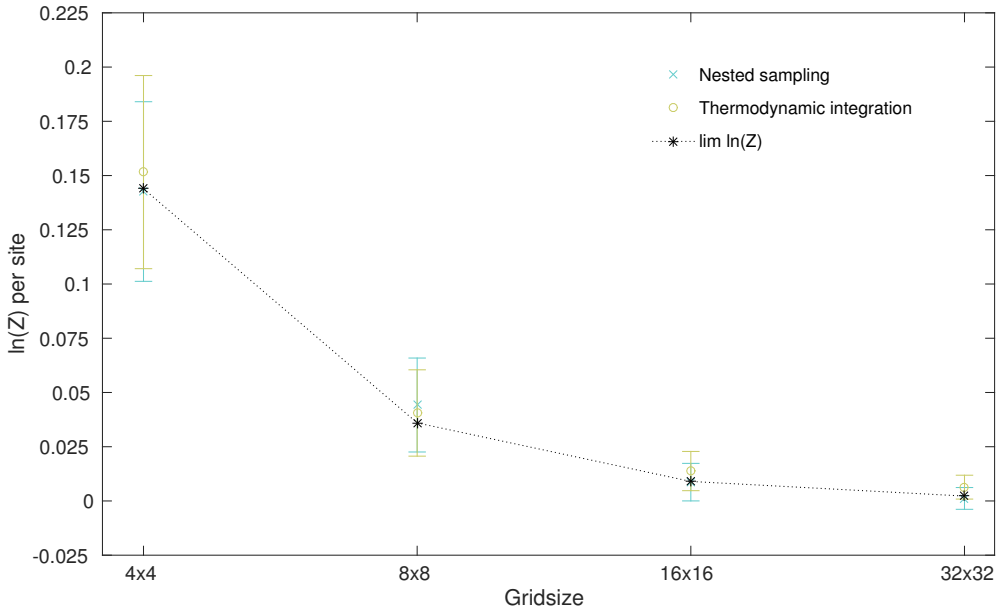


Figure 7.19: $ln(Z)$ of the q=10 Potts model computed by thermodynamic integration and nested sampling in dependence of the grid-size.

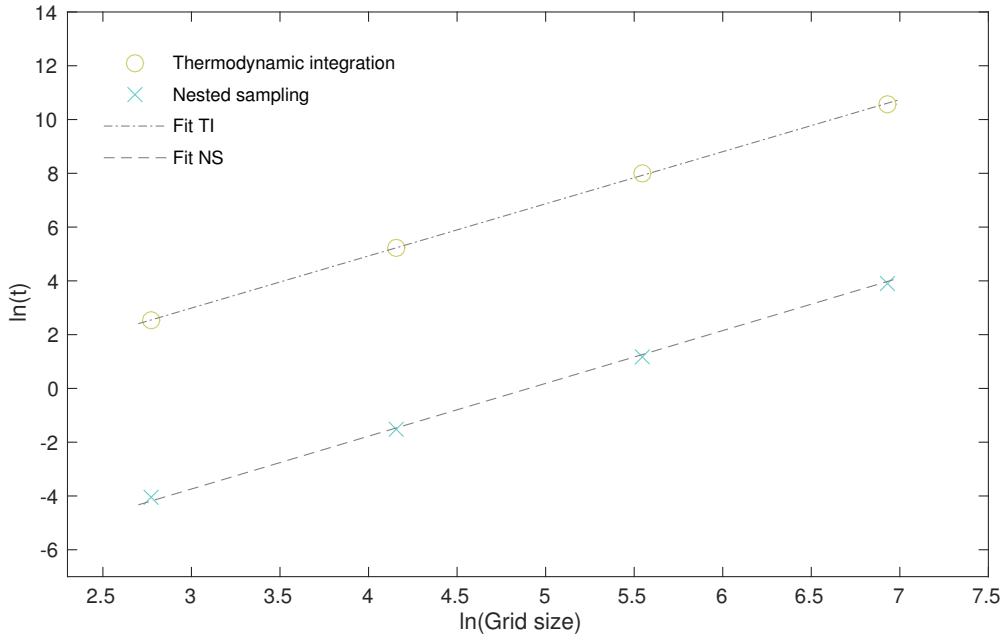


Figure 7.20: Investigating the q=10 Potts model, the logarithmic time versus the logarithmic grid size exhibits a similar scaling for both methods. The time required for the Z_π evaluation is not included in the time for NESA, because it only scales linearly with N.

Figure 7.21 shows the number of SW updates for each algorithm for grid-sizes from 4×4 to 32×32. Being the most time consuming element of both procedures, the plots of TI and NESA (without the number of SW updates for the preceding TI computation) exhibit the same structure as the respective graphs of computational time.

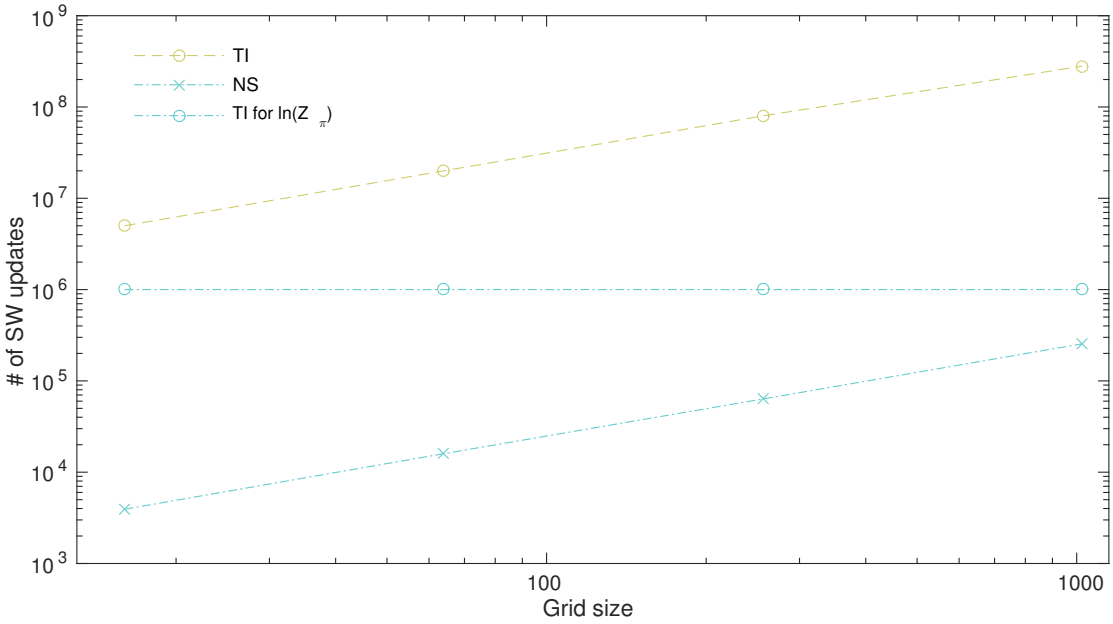


Figure 7.21: Number of SW updates required for the results of TI and NESA. The SW updates for the NESA run and the additional TI to evaluate $ln(Z_\pi)(\beta = ln(2))$ are indicated separately.

Assuming that the SW update is the most time consuming part, the total time of both algorithms is given via

$$t_{\text{total}} = n_{\text{SW}}\, t_{\text{SW}} \tag{7.15}$$
$$= e^{b_n}\, N^{a_n}\; e^{b_t}\, N^{a_t} \tag{7.16}$$
$$= e^{b}\, N^{a} \tag{7.17}$$

Here n_{SW} denotes the number of SW updates and t_{SW} denotes the time per SW update. Both quantities exhibit a power law dependence on the system size.
Again the corresponding logarithmic values $ln(t_{SW})$ and $ln(n_{SW})$ in dependence on $ln(N)$ are linearly fitted. The obtained values for the exponents are listed in table 7.2

Table 7.2: Values of the exponents in equation 7.16 determined by a linear fit of the logarithmic values of t_{SW} and n_{SW}. The last two lines depict the exponents found earlier for the fits of the total time.

a) Fit of # of SW updates	a_n	b_n
TI	0.97 ± 0.07	12.76 ± 0.37
NESA	1.001 ± 0.002	5.51 ± 0.12
b) Fit of t per SW update	a_t	b_t
TI	0.97 ± 0.05	-15.59 ± 0.24
NESA	0.96 ± 0.04	-15.14 ± 0.26
c) Combination of a) and b)	$a = a_n + a_t$	$b = b_n + b_t$
TI	1.94 ± 0.12	-2.83 ± 0.61
NESA	1.96 ± 0.04	-9.63 ± 0.38
d) Fit of total time	a	b
TI	1.94 ± 0.07	-2.83 ± 0.37
NESA	1.96 ± 0.03	-9.63 ± 0.25

The results show that for both methods the number of SW updates roughly scales linearly with the system size N. Also the time per SW update exhibits a approximately linear scaling with N. The scaling of the total time in dependence on the grid size computed via the individual components yields the same results as we found earlier by means of fitting the total time.

Finally we find that both methods show the same scaling within the error bars. In each case the total time scales approximately quadratic with N. From the results in equation 7.13 and 7.14 we find that the prefactor e^b leads to a lower total computational time for the NESA algorithm by a factor of 898 (483, 1669). Again it needs to be noted that for this evaluations the time for the $ln(Z_\pi)$ computation has not been included and the above factor is only valid for system greater $\approx 32 \times 32$.

7.5 Analysis of the parallel nested sampling algorithm

In this section we implement the parallel NESA algorithm (as presented in section 6.5) and analyse it for system sizes from 4×4 to 64×64. For the case of $k = 1$ a sample of $K = 100$ walkers is employed. If $k > 1$, K is scaled with the number of cores according to equation 6.39. The computations for $k = 1, 2, 4$ are performed

on a processor including 4 cores.
For the evaluation of each $ln(Z)$ value, 100 sequences of prior samples are used. The samples X are drawn from the respective Beta distribution and afterwards their logarithmic value is computed. The error of $ln(Z)$ is evaluated according to equation 6.33.

Figure 7.22 shows the wall-clock time of the parallel NESA evaluations for the analysed grid-sizes. The wall-clock time depicts the real time, which passes by for the user, while computing a task. It includes also the time needed for e.g. input and output operations.

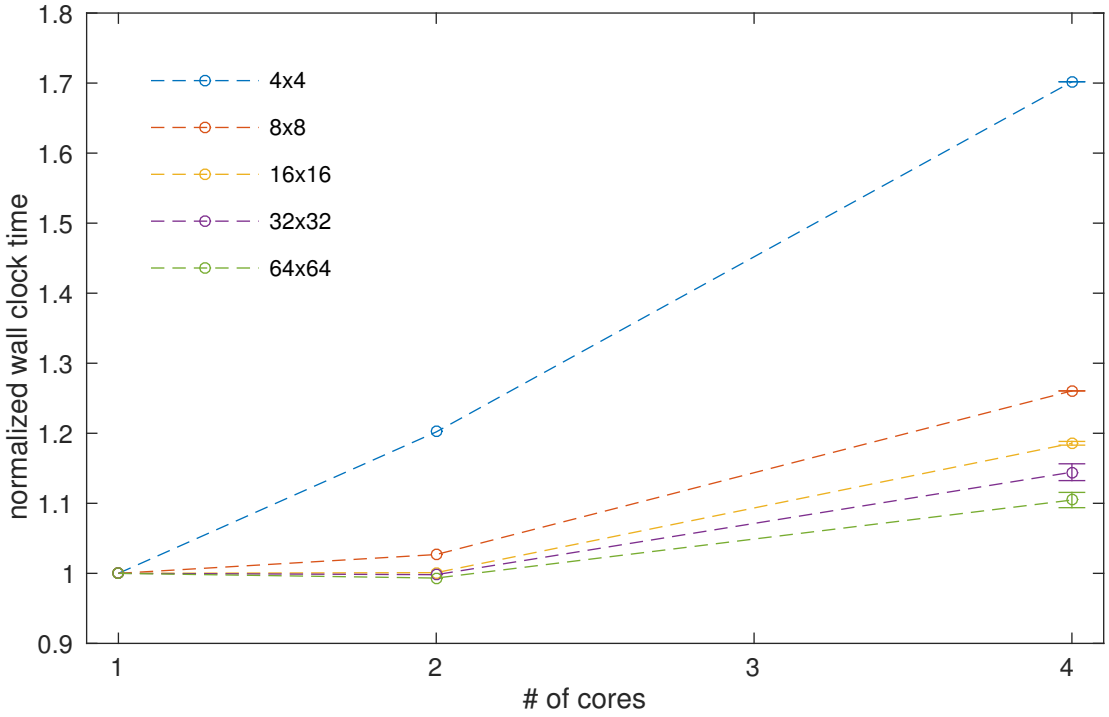


Figure 7.22: Normalized wall clock time required by parallel NESA for the $q = 10$ Potts model for various grid-sizes in dependence of the number of employed cores k.

The values displayed in figure 7.22 are normalized with respect to the case of $k = 1$. Each value is the mean of five independent runs. The respective standard deviation of the time is only drawn for the measurements at $k = 4$ for clarity. One can observe that for increasing grid-sizes the increase in wall clock time, over the number of cores, shrinks. For the 64×64 system an increase of about 10 percent for

4 cores is measured. The slight kink in the plot is presumably caused by reaching the full capacity of the processor, when employing 4 cores.
The accumulated time, of all involved cores, actually spent for processing the code, is referred to as CPU time. When processing code in parallel this value will be larger than the wall-clock time. Figure 7.23 displays the CPU time of the parallel nested sampling algorithm for different grid sizes. Again the values are normalized with respect to the time at $k = 1$. The mean values and the standard deviation are evaluated, again from the 5 independent runs used above. The CPU time scales roughly linearly with the number of cores. The reason that the real computational effort increases with larger k is due to the higher number of walkers K.

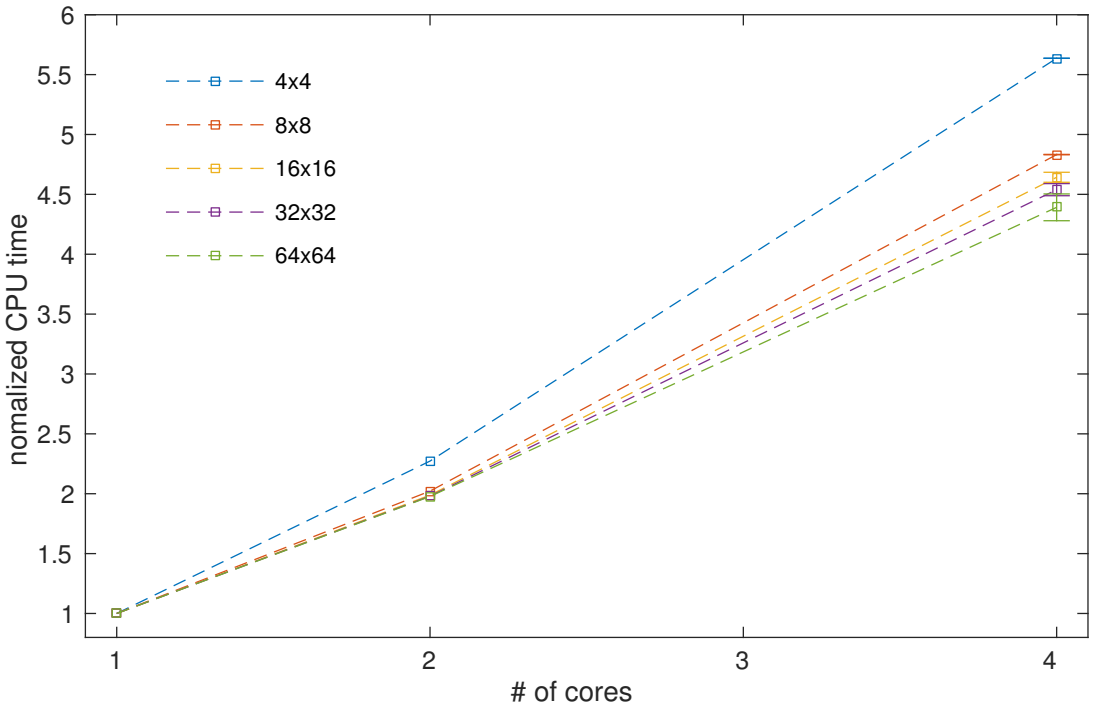


Figure 7.23: Normalized CPU time required by parallel NESA for the $q = 10$ Potts model for various grid-sizes in dependence of the number of employed cores k.

The results for $ln(Z)$ are given in figure 7.24. The indicated $ln(Z)$ values are the average of 5 independent samples and the error bars are the mean value of the corresponding standard deviations. Although the number of walkers in the NESA runs has been scaled to yield constant variance, we find a slight decrease in the error of $ln(Z)$ for all sizes.

We can conclude, that the parallel NESA algorithm does not yield a distinct improvement for the problem of the 10 state Potts model. The decreasing trend in variation as well as the trend to constant values for the wall-clock time for larger systems suggest that for systems with a more structured phase space, the parallel NESA could be advantageous. Because with parallel NESA we are able to simulate more walkers within the same time, which is advantageous for multi-modal likelihood functions, where we need enough walkers per mode if we employ the present clone approach (see section 6.2).

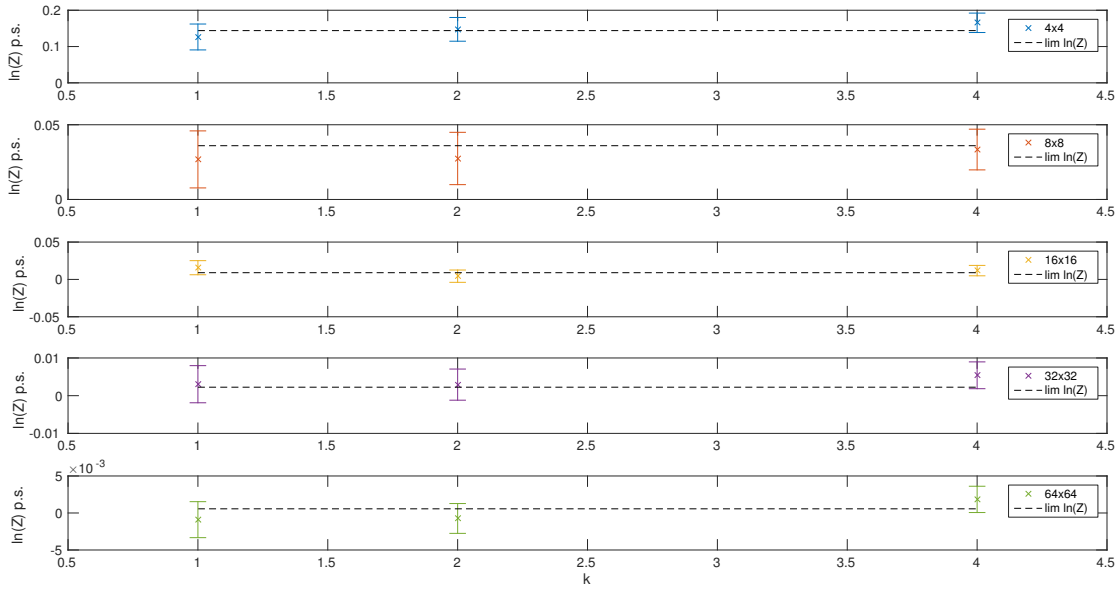


Figure 7.24: $ln(Z)$ per site versus the number k of employed cores for the q=10 Potts model computed for various grid-sizes by means of the parallel NESA algorithm.

8 Conclusion

In this thesis the application of nested sampling on physical problems was studied. For this purpose the evaluation of the partition function of the two dimensional Potts model was investigated.
As alternative method the established thermodynamic integration has been employed. A correlation time analysis of this method, displayed a massive slowing down near the critical temperature. In the temperature range, where the prior normalization, required for the nested sampling algorithm, is evaluated, however, the correlation times are small and pose no significant slowing down. The determination of correlation times for nested sampling yielded negligible values.

Furthermore expressions to determine thermodynamic quantities from the results of a single nested sampling run, without employing numerical derivatives were obtained. The results for the energy were confirmed by a comparison with the results from a multicanonical simulation.
Both methods exhibit a power law scaling with increasing grid-sizes. Fits of the CPU time yielded roughly a quadratic exponent for both methods. Though, due to a smaller prefactor, the nested sampling algorithm is about three orders of magnitude faster than thermodynamic integration. For small system sizes the time for the additional thermodynamic integration computation to calculate the prior normalization, which scales only linearly with system size, accounts for most of the computational time.
The time per Swendsen Wang update for each of the methods exhibits approximately the same value and the same scaling, which is reasonable, because for both methods the cluster identification during each update is the most time consuming task. On the other hand the same scaling in the number of Swendsen Wang updates is in the first sight stunning, because nested sampling does not require to increase the number of updates, due to the negligible correlation time. However, where thermodynamic integration needs to increase the number of Swendsen Wang

updates for larger systems due to rising correlation times, nested sampling needs more updates to reach the maximum likelihood value. This increase seems to scale similarly for both methods. Eventually the reason for the time difference is, that for each expectation value obtained during thermodynamic integration, we need to evaluate multiple Swendsen Wang updates. An important advantage of nested sampling is, that the partition function for all temperatures is available from the results of a single run.
The partition function of a 10 state Potts model was evaluated via nested sampling up to a grid-size of 256×256, where the total time for the largest system was about 57 h. Employing thermodynamic integration for the same evaluation would already take about 4 years.

Finally a parallelised version of the nested sampling algorithm was investigated. The parallel implementation, employing 1 to 4 cores, was tested for system sizes ranging from 4×4 to 64×64. The wall clock times as well as the respective results for the partition function did not reveal a relevant improvement for the application of the parallelised version to the Potts model. However, parallel nested sampling has to use more walkers, which is the reason why there is no CPU time gain in the parallel implementation, but leads to a better sampling of the phase space, while keeping the wall clock time constant. Hence when dealing with problems involving a multiple peaked likelihood, the parallel implementation will become advantageous.

The independence of the algorithm on temperature, makes nested sampling able to deal with first order phase transitions. The method poses a promising challenger for the multicanonical and multibondic algorithms, which are state of the art algorithms for dealing with Potts models.
In our opinion nested sampling depicts a high potential algorithm for applications in statistical physics and due to its uniqueness it deserves a place in a physicists standard repertoire of simulation techniques.

9 Appendix

9.1 Handling sums of large numbers

Determining certain quantities for problems in statistical physics often require to evaluate exponential functions with large numbers as arguments. For instance the computation of the partition function (equation 2.8) or the computation of expectation values of observables involve such evaluations due to the Boltzmann factor $e^{-\beta H}$. Consider a Potts model with $q = 2$ and size 100×100. The energy in the case of total disorder yields $H = 0$, whereas the ground-state energy yields $H = -2 \times 10^4$. For intermediate β values the computation of the Boltzmann factor, in the order of $\exp(10^4)$, leads to overflow errors. It is therefore preferable to work with the respective logarithmic values.
Therefore we need a technique to evaluate sums, where the individual contributions x_i are given in their logarithmic value $ln\, x_i$, i.e.

$$\sum_i x_i = \sum_i e^{ln\, x_i}$$

Note that we need to deal in our case only with positive summands x_i. Calculating the sum straightforward, would infer with our intent to never calculate the exponentials explicitly. Let us investigate the general case of a sum of two terms $\Omega = x_n + x_m$, where x_n and x_m depict positive numbers. Assuming we know $ln\, x_n$ and $ln\, x_m$, then Ω can be retrieved via

$$\Omega \;=\; \max(x_n, x_m)\left(1 + \frac{\min(x_n, x_m)}{\max(x_n, x_m)}\right)$$

and the logarithm of Ω reads as

$$ln(\Omega) \;=\; \max(ln\, x_n, ln\, x_m) + ln\left[\, 1 + e^{(\,\min(ln\, x_n, ln\, x_m) - \max(ln\, x_n, ln\, x_m)\,)}\,\right]$$

The argument of the exponential function is constructed to be negative, therefore only underflow errors will occur, when the difference becomes large. These values are set to zero and pose therefore no problem for the resulting $ln(\Omega)$. For sums with multiple summands this approach leads to

$$ln(\Omega) \;=\; \max_i(ln\, x_i) + ln\left[\, 1 + e^{\sum_i (\, ln\, x_i - \max_j (ln\, x_j)\,)}\,\right]$$

For an optimal evaluation of the sum in the exponent, the summands should be ordered in increasing order to prevent floating point errors due to finite precision.

With this approach it is never necessary to calculate either of the numbers x_n, x_m or Ω during the evaluation of the sum. (compare [5] for the presented information in this section).

9.2 Beta and Gamma function

The Beta function is defined by

$$B(\alpha, \rho) \coloneqq \int_0^1 p^{\alpha-1}\,(1-p)^{\rho-1}\,dp \tag{9.1}$$

The Gamma function reads as

$$\Gamma(\alpha) \;\coloneqq\; \int_0^\infty t^{\alpha-1}\,e^{-t}\,dt \tag{9.2}$$

$$\Gamma(\alpha) \;=\; (\alpha-1)\,\Gamma(\alpha-1) \tag{9.3}$$

$$\Gamma(m) \;=\; (m-1)!\;\text{ for m} \in \mathbb{N} \tag{9.4}$$

Both functions are related via

$$B(\alpha, \rho) = \frac{\Gamma(\alpha)\,\Gamma(\rho)}{\Gamma(\alpha+\rho)} \tag{9.5}$$

For more information the reader is referred to [26].

Bibliography

[1] Fork join model, October 2014. Page Version ID: 604532727.

[2] Stuart Aitken and Ozgur E. Akman. Nested sampling for parameter inference in systems biology: application to an exemplar circadian model. *BMC Systems Biology*, 7(1):72, July 2013.

[3] Clive F. Baillie and Paul D. Coddington. Comparison of cluster algorithms for two-dimensional Potts models. *Physical Review B*, 43(13):10617–10621, May 1991.

[4] Bernd A. Berg. Introduction to Multicanonical Monte Carlo Simulations. *arXiv:cond-mat/9909236*, September 1999. Fields Inst.Commun.26:1,2000.

[5] Bernd A Berg. Multicanonical simulations step by step. *Computer Physics Communications*, 153(3):397–406, July 2003.

[6] Nikolas S. Burkoff, Csilla Varnai, Stephen A. Wells, and David L. Wild. Exploring the Energy Landscapes of Protein Folding Simulations with Bayesian Computation. *Biophysical Journal*, 102(4):878–886, February 2012.

[7] Robert G. Edwards and Alan D. Sokal. Generalization of the Fortuin-Kasteleyn-Swendsen-Wang representation and Monte Carlo algorithm. *Physical Review D*, 38(6):2009–2012, September 1988.

[8] F. Feroz and J. Skilling. Exploring Multi-Modal Distributions with Nested Sampling. *arXiv:1312.5638 [astro-ph, physics:physics, stat]*, pages 106–113, 2013. AIP Conference Proceedings, Volume 1553, pp. 106-113 (2013).

[9] C. M. Fortuin and P. W. Kasteleyn. On the random-cluster model: I. Introduction and relation to other models. *Physica*, 57(4):536–564, February 1972.

[10] Vivek K. Gore and Mark R. Jerrum. The Swendsen Wang Process Does Not Always Mix Rapidly. *Journal of Statistical Physics*, 97(1-2):67–86, October 1999.

[11] R. Wesley Henderson and Paul M. Goggans. Parallelized Nested Sampling. (Conference proceedings for the 33rd International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering), 2014.

[12] Wolfhard Janke. *Multibondic cluster algorithm for Monte Carlo simulations of first-order phase transitions*. 1995.

[13] Helmut G. Katzgraber. Introduction to Monte Carlo Methods. *arXiv:0905.1629 [cond-mat, physics:physics]*, May 2009. arXiv: 0905.1629.

[14] David P Landau and K Binder. *A guide to Monte Carlo simulations in statistical physics*. Cambridge University Press, Cambridge; New York, 2009.

[15] David MacKay. Nested sampling explanatory illustrations, April 2004.

[16] Stefano Martiniani, Jacob D. Stevenson, David J. Wales, and Daan Frenkel. Superposition Enhanced Nested Sampling. *Physical Review X*, 4(3):031034, August 2014.

[17] Pia Mukherjee, David Parkinson, and Andrew R. Liddle. A Nested Sampling Algorithm for Cosmological Model Selection. *The Astrophysical Journal*, 638(2):L51–L54, February 2006. arXiv: astro-ph/0508461.

[18] Iain Murray, David MacKay, Zoubin Ghahramani, and John Skilling. Nested sampling for Potts models. pages 947–954, 2005.

[19] M. E. J. Newman and R. M. Ziff. Fast Monte Carlo algorithm for site or bond percolation. *Physical Review E*, 64(1):016706, June 2001.

[20] Heng Qiao and P. Pal. Generalized nested sampling for compression and exact recovery of symmetric Toeplitz matrices. In *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 443–447, December 2014.

[21] Kari Rummukainen. Multicanonical methods and cluster algorithms, 2008.

[22] John Skilling. Nested sampling for general Bayesian computation. *Bayesian Analysis*, 1(4):833–859, December 2006. Mathematical Reviews number (MathSciNet) MR 2282208.

[23] John Skilling. Nested Samplings Convergence. In *AIP Conference Proceedings*, volume 1193, pages 277–291. AIP Publishing, December 2009.

[24] John Skilling. Bayesian computation in big spaces-nested sampling and Galilean Monte Carlo. In *AIP Conference Proceedings*, volume 1443, pages 145–156. AIP Publishing, May 2012.

[25] Robert H. Swendsen and Jian-Sheng Wang. Nonuniversal critical dynamics in Monte Carlo simulations. *Physical Review Letters*, 58(2):86–88, January 1987.

[26] von der Linden, Dose, and von Toussaint. *Bayesian Probability Theory Applications in the Physical Sciences.* Cambridge Univ Pr, 2014.

[27] Udo von Toussaint. Bayesian inference in physics. *Reviews of Modern Physics*, 83(3):943–999, September 2011.

[28] S. Lewis W. Vanderbauwhede. Implementing data parallelisation in a Nested-Sampling Monte Carlo algorithm. pages 512–518, 2013.

[29] F. Y. Wu. The Potts model. *Reviews of Modern Physics*, 54(1):235–268, January 1982.

List of abbreviations

MC	Monte Carlo
MCMC	Markov chain Monte Carlo
MH	Metropolis Hastings
SW	Swendsen Wang
FK	Fortuin Kasteleyn random cluster model
TI	Thermodynamic integration
MUCA	Multicanonical simulation
NESA	Nested sampling
PDF	Probability density function
CPU	Central processing unit

List of Figures

2.1 Fortuin-Kasteleyn representation 7

4.1 Thermodynamic integration for a 3×3, $q = 5$ system 18
4.2 Thermodynamic integration for a 3×3, $q = 5$ system with suboptimal parameters 18
4.3 $\ln(Z)$ for a 3×3, $q = 5$ Potts system evaluated via TI 19
4.4 Correlation times for a 8×8, $q = 10$ 20
4.5 Correlation times for a 8×8, $q = 10$, in $J\beta$ range around 2 21
4.6 τ_{int} results for a $q = 10$ Potts model evaluated for different grid-sizes 23

5.1 Comparison of the multicanonical and the canonical distribution . . 24

6.1 Principle of the nested sampling algorithm. 29
6.2 Relation of $L(\mathbf{x})$ and $L(X)$ in a 2 dimensional parameter space. . . 31
6.3 Update of the walker with the lowest likelihood 32
6.4 Possible ways to evaluate the estimate of Z 33
6.5 Degeneracy of configurations with a certain number of bonds for a 3×3, $q = 2$ system. 34
6.6 Spawning of threads with OpenMP 48
6.7 Truncated binomial distribution of the new number of bonds D' . . 53
6.8 Decision tree for evaluating the probability of the two displayed cases. 54
6.9 Time comparison of ways for choosing the new bond number 55
6.10 Time and partition function per site versus the number of $J\beta$ steps 56
6.11 Time and partition function per site versus the number of SW updates at $J\beta = ln(2)$ 57
6.12 Standard deviation of the partition function for the $q = 10$ Potts model versus the number of SW updates 58
6.13 CPU time per site and $ln(Z_\pi)$ per site for a $q = 10$ Potts model and $J\beta_{max} = ln(2)$ in dependence on the grid-size 59

6.14 Standard deviation of the $ln(Z_\pi)$ for the $q = 10$ Potts model versus grid size 59
6.15 Exact solution of the partition function for $3x3$ systems with the number of colours $q = 2$ and $q = 5$. 60
6.16 Exact partition function computed in spin and bond variables for the 3×3 and $q = 2$ Potts model. 61
6.17 Distributions of the integrated correlation time for a 3×3 system . 63
6.18 Correlation times of a 16×16 Potts model with $q = 10$ and a constraint on the likelihood. 64
6.19 $ln(Z)$ for the 16×16 system with standard deviation from 10 runs and the error evaluated from 100 prior samples. 64

7.1 Positions of walkers for each step n during a single nested sampling run for the Gaussian likelihood function 67
7.2 Likelihood values from a single NESA run with 10 walkers placed at their true X value and placed at three samples of prior sequences. 68
7.3 $ln(Z)$ results of 20 independent nested sampling runs for k=1,4,12 . 70
7.4 $ln(Z)$ of the Gaussian likelihood function in dependence of the number of walkers K for $k = 1$. 71
7.5 Standard deviation of $ln(Z)$ in dependence of K. 71
7.6 The increase of CPU time over increasing number of walkers K . . . 72
7.7 $ln(Z)$ for a 4×4, q=10 Potts model displayed in dependence on the number of NESA steps 73
7.8 Likelihood over the logarithmic prior mass of the 4×4, $q = 10$ Potts system for different temperatures 75
7.9 Summands of equation 4.6 for a 4×4 and $q = 10$ system at different temperatures 76
7.10 Likelihood function versus the prior mass for the 128×128, $q = 10$ system 77
7.11 $ln(Z)$ for a 128×128, q=10 Potts model displayed in dependence on the number of NESA steps 77
7.12 $ln(Z)$ versus the number of prior sequences s 78
7.13 $ln(Z)$ of a 32×32 and q=10 system versus the number of walkers K 79
7.14 $ln(Z)$ of a 16×16, q=10 system versus $J\beta$ 80
7.15 $ln(Z)$ of a 128×128, q=10 system versus $J\beta$ 80

7.16 Energy per site versus $J\beta$ evaluated by means of NESA and MUCA for the 20×20 and $q = 10$ Potts system. 84
7.17 Breakdown of the total CPU time versus N for the nested sampling evaluation of the $q = 10$ Potts model. 86
7.18 $ln(Z)$ for the $q = 10$ Potts model computed by the nested sampling algorithm versus grid size N. 86
7.19 $ln(Z)$ of the q=10 Potts model computed by thermodynamic integration and nested sampling in dependence of the grid-size. 88
7.20 Investigating the q=10 Potts model, the logarithmic time versus the logarithmic grid size exhibits a similar scaling for both methods. The time required for the Z_π evaluation is not included in the time for NESA, because it only scales linearly with N. 88
7.21 Number of SW updates required for the results of TI and NESA . . 89
7.22 Normalized wall clock time required by parallel NESA for the $q = 10$ Potts model for various grid-sizes in dependence of k 91
7.23 Normalized CPU time required by parallel NESA for the $q = 10$ Potts model for various grid-sizes in dependence of k 92
7.24 $ln(Z)$ per site versus the number k of employed cores for the q=10 Potts model computed for various grid-sizes by means of the parallel NESA algorithm. 93

List of Tables

4.1 Comparison of τ_{int} at the infinite critical point for $q = 3$ Potts systems. For further discussion see text 22
4.2 τ_{int} at the infinite critical point for $q = 10$ Potts systems. 23

6.1 Integrated correlation times $\bar{\tau}_{int}$. See text for further explanation. . 62

7.1 $\ln(Z)$ for a 16×16 model computed by TI for parameters near phase transition. The second row shows the results from [18]. 84
7.2 Values of the exponents in equation 7.16 determined by a linear fit of the logarithmic values of t_{SW} and n_{SW}. The last two lines depict the exponents found earlier for the fits of the total time. 90