

# Scaling properties of a parallel implementation of the multicanonical algorithm

Johannes Zierenberg\*, Martin Marenz, Wolfhard Janke

Institut für Theoretische Physik, Universität Leipzig, Postfach 100920, D-04009 Leipzig, Germany

## ARTICLE INFO

### Article history:

Received 18 July 2012

Received in revised form

6 December 2012

Accepted 7 December 2012

Available online 14 December 2012

### Keywords:

Parallel

Multicanonical

Ising

Potts

## ABSTRACT

The multicanonical method has been proven powerful for statistical investigations of lattice and off-lattice systems throughout the last two decades. We discuss an intuitive but very efficient parallel implementation of this algorithm and analyze its scaling properties for discrete energy systems, namely the Ising model and the 8-state Potts model. The parallelization relies on independent equilibrium simulations in each iteration with identical weights, merging their statistics in order to obtain estimates for the successive weights. With good care, this allows faster investigations of large systems, because it distributes the time-consuming weight-iteration procedure and allows parallel production runs. We show that the parallel implementation scales very well for the simple Ising model, while the performance of the 8-state Potts model, which exhibits a first-order phase transition, is limited due to emerging barriers and the resulting large integrated autocorrelation times. The quality of estimates in parallel production runs remains of the same order at the same statistical cost.

© 2012 Elsevier B.V. All rights reserved.

## 1. Introduction

Monte Carlo simulations are an important tool to investigate a wide range of theoretical models with respect to their statistical properties such as phase transitions, structure formation and more. Throughout the last two decades, umbrella sampling algorithms like the multicanonical [1,2] or the Wang–Landau [3] algorithm have been proven to be very powerful for investigations of statistical phenomena, especially first-order phase transitions, for lattice and off-lattice models. They have been applied to a variety of systems with rugged free-energy landscapes in physics, chemistry and structural biology [4].

Due to the fact that computer performance increases mainly in terms of parallel processing on multi-core architectures, a parallel implementation is of great interest, if the additional cores bring a benefit to the required simulation time. We present the scaling properties of a simple and straightforward parallelization of the multicanonical method, which has been reported in a similar way in [5] without much detail to the performance. This parallelization considers independent Markov chains, keeping communication to a minimum. Thus, it can be added on top of the multicanonical algorithm without much modification or system-dependent considerations and is also suitable for systems with simple energy calculations. Similar to this parallelization, there have been previous reports for the Wang–Landau algorithm [6,7], which needed a little more adaptation to the algorithm.

## 2. Multicanonical algorithm

The multicanonical method allows us to sample a system over a range of canonical ensembles at the same time. This is possible, because the statistical weights are modified in such a way that the simulation reaches each configuration energy of a chosen interval with equal probability, resulting in a flat energy histogram. To this end, the canonical partition function, in terms of the density of states  $\Omega(E)$ , is modified in the following way:

$$Z_{\text{can}} = \sum_{\{x_i\}} e^{-\beta E(\{x_i\})} = \sum_E \Omega(E) e^{-\beta E}$$

$$\rightarrow Z_{\text{MUCA}} = \sum_{\{x_i\}} W(E(\{x_i\})) = \sum_E \Omega(E) W(E). \quad (1)$$

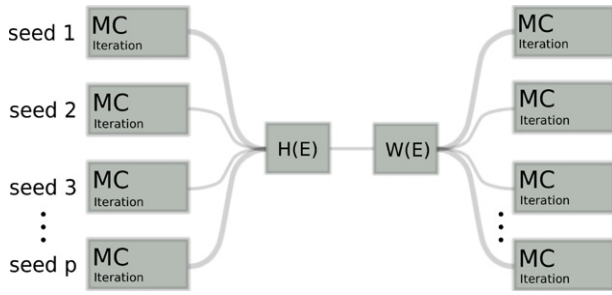
In order that each energy state occurs with the same probability, as requested above, the statistical weights have to equal the inverse density of states  $W(E) = \Omega^{-1}(E)$ . After an equilibrium simulation with those weights, it is possible to reweight to all canonical ensembles with a Boltzmann energy distribution covered by the flat histogram. This can be done for example by time-series reweighting, where in the average each measured observable is multiplied with its desired weight and divided by the weight with which it was measured:

$$\langle O \rangle_\beta = \frac{\langle O_i e^{-\beta E_i} W^{-1}(E_i) \rangle_{\text{MUCA}}}{\langle e^{-\beta E_i} W^{-1}(E_i) \rangle_{\text{MUCA}}}. \quad (2)$$

Of course, the density of states and consequently the weights that yield a flat energy histogram are usually not known in

\* Corresponding author.

E-mail address: [zierenberg@itp.uni-leipzig.de](mailto:zierenberg@itp.uni-leipzig.de) (J. Zierenberg).



**Fig. 1.** Scheme of the parallel implementation of the multicanonical algorithm on  $p$  cores. After each iteration with independent Markov chains but identical weights, the histograms are merged, the new weights are estimated and the weights are distributed onto all processes again.

advance. Therefore the weights have to be obtained iteratively. In the most simple way consecutive weights are obtained from the last weights and the current energy histogram,  $W^{(n+1)}(E) = W^{(n)}(E)/H^{(n)}(E)$ . More sophisticated methods exist, where the full statistics of previous iterations is used for a stable and efficient approximation of the density of states [2]. All our simulations use this recursive version with logarithmic weights in order to avoid numerical problems.

#### Parallel version

The idea of this parallel implementation, similar to [5], is to distribute the time consuming generation of statistics on  $p$  independent processes. All processes perform equilibrium simulations with identical weights  $W_i^{(n)} = W^{(n)}$ ,  $i = 1, \dots, p$ , but with different random number seeds, resulting in similar but independent energy histograms  $H_i^{(n)}(E)$ . The histograms are merged after each iteration and one ends up with  $H^{(n)}(E) = \sum_i H_i^{(n)}(E)$ . According to the weight modification of choice, the collected histogram is processed together with the previous weights in order to estimate the consecutive weights  $W^{(n+1)}$ . The new weights are distributed onto all processes, which run equilibrium simulations again. That way, the computational effort may be distributed on several cores, allowing us to generate the same amount of statistics in a fraction of the time. It is important to notice that a modification of the program only influences the histogram merging and the distribution of the new weights, see Fig. 1. The iterations are independent copies run in parallel and the weight modification is performed on the master process as in the non-parallelized case.

### 3. Systems and implementation issues

We consider two discrete two-dimensional spin systems, namely the well known Ising model and the  $q$ -state Potts model with  $q = 8$ , where the Ising model can be mapped onto the  $q = 2$  Potts model. The Ising model exhibits a second-order phase transition at  $\beta_0 = \ln(1 + \sqrt{2})/2$  and the 8-state Potts model exhibits a first-order phase transition at  $\beta_0 = \ln(1 + \sqrt{8})$ . The spins are located on a square lattice with side length  $L$  and interact only between nearest neighbors. In the case of the Ising model, the interaction is described by the Hamiltonian

$$\mathcal{H}^{(\text{Ising})} = -J \sum_{\langle i,j \rangle} s_i s_j, \quad (3)$$

where  $J$  is the coupling constant and  $s_i, s_j$  can take the values  $\{-1, 1\}$ . For the  $q$ -state Potts model, where each site assumes values from  $\{0, \dots, q-1\}$ , the nearest-neighbor interaction is described by

$$\mathcal{H}^{(\text{Potts})} = -J \sum_{\langle i,j \rangle} \delta(s_i, s_j), \quad (4)$$

where  $\delta(s_i, s_j)$  is the Kronecker-Delta function which is only non-zero in the case  $s_i = s_j$ .

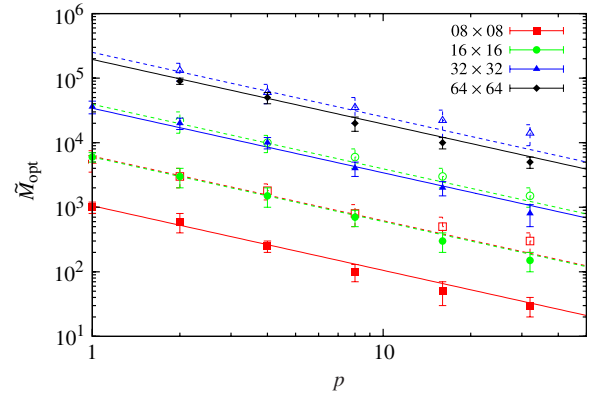
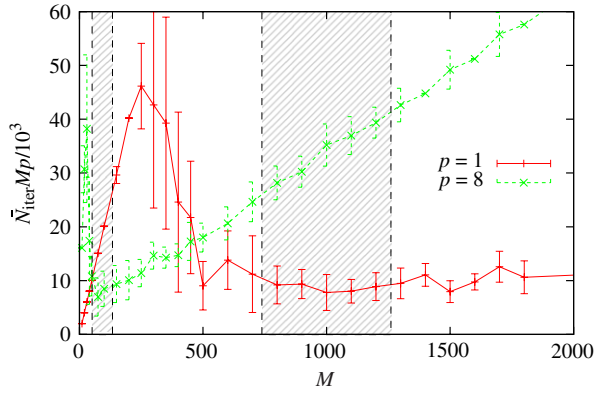
In both cases the number of discrete energy states is proportional to the number of lattice sites  $V = L^2$ , such that the width of the energy range increases quickly with system size. The simulations in this study start at infinite temperature, i.e.,  $\beta = 1/T = 0$  with quite narrow energy histograms. Because an estimation of successive weights is only possible for energies with non-zero histogram entries, the number of iterations may be very large for wide energy ranges. In order to ensure faster convergence, our implementation includes after each estimation of weights a correction function, which linearly interpolates the logarithmic weights at the boundaries of the sampled region (with a range of  $L$  bins), allowing the next iteration to sample a larger energy region. The MUCA weights are converged if the last iteration covered the full energy range and all histogram entries are within half and twice the average histogram entry. Between convergence of the weights and the final production run, the systems are thermalized again in order to yield correct estimates of the observables. In both cases, each sweep includes  $V$  numbers of spin updates.

### 4. Performance and scaling

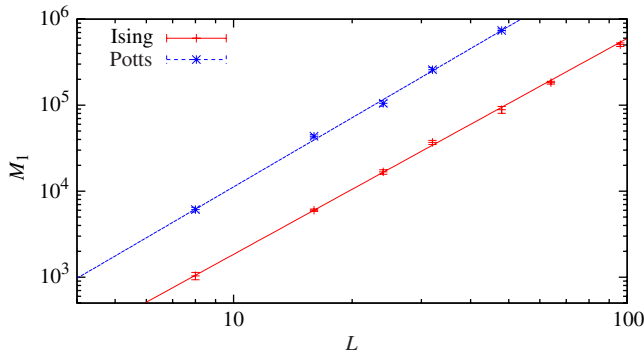
In order to estimate the performance and the speedup of the parallel algorithm appropriately, we performed the analysis in two steps. First, we estimated the optimal number of sweeps per iteration and core, which we will refer to as  $M$ . To this end, we performed parallel MUCA simulations over a wide range of  $M$  for different lattice sizes  $L$  and number of cores  $p$ . The simulations were thermalized once in the beginning on every core, continuing the next iteration with the last state of the previous iteration on that core. This violates the equilibrium condition a little, as no intermediate thermalization phase was applied and part of the iteration was needed to reach equilibrium. This is accepted in order to compare the performance equally without an additional parameter to optimize next to  $M$ . Furthermore, the physical results were not influenced, because each Markov chain was thermalized before the final production run. We determined the mean number of iterations until convergence to a flat histogram  $\bar{N}_{\text{iter}}$  as the average over 10 simulations at different initial seeds. Plotting the total number of sweeps  $\bar{N}_{\text{iter}} M p$  versus  $M$ , we can estimate the optimal number of sweeps per iteration and core  $\tilde{M}_{\text{opt}}$  as the minimum of this function (see Fig. 2(left)). For a small number of cores, this curve has a rather broad minimum, introducing a rough estimate. If, on the other hand, we stretch the curves along the  $x$ -axis with the number of cores, the outcomes look quite similar.

Selected results of the estimation of  $\tilde{M}_{\text{opt}}$  are shown in Fig. 2(right). We see that for different lattice sizes and spin models the dependence on the number of cores may be described by a  $1/p$  power law, where the amplitudes seem to depend on the system size and the number of states (notice that the Potts model curves nearly coincide with those of the Ising model with 4 times system size). In order to measure the performance equally, it is convenient to describe  $\tilde{M}_{\text{opt}}$  by a function of system size  $L$  and number of cores  $p$ ,  $\tilde{M}_{\text{opt}}(L, p) \approx M_{\text{opt}}(L, p) = M_1(L)/p$ , where  $M_1$  is the interpolated optimal  $M$  for one core. Therefore, we estimated  $\tilde{M}_{\text{opt}}$  for the square lattice sizes 8, 16, 24, 32, 48, 64, and 96 (the latter two only for Ising) with  $p \leq 32$  and fitted for fixed size  $M_{\text{opt}}(L, p) = M_1(L)/p$ . The obtained  $M_1(L)$  were plotted over  $L$  and fitted with a power law (see Fig. 3). In the end, the optimal number of sweeps per core and iteration were systematically described by the functions

$$\begin{aligned} M_{\text{opt}}^{(\text{Ising})}(L, p) &= 5.7(5) \times L^{2+0.51(4)} \frac{1}{p} \\ M_{\text{opt}}^{(8\text{Potts})}(L, p) &= 24(4) \times L^{2+0.67(6)} \frac{1}{p}, \end{aligned} \quad (5)$$



**Fig. 2.** (Left) Example for the estimation of  $M_{\text{opt}}$  for the  $8 \times 8$  Ising model with 1 and 8 cores. The minimum of the total statistics  $\bar{N}_{\text{iter}}Mp$  with respect to  $M$  yields  $\bar{M}_{\text{opt}}$ . (Right) Plot of  $\bar{M}_{\text{opt}}$  versus the number of cores  $p$  used in the parallel MUCA simulation for the Ising (filled symbols) and the 8-state Potts (open symbols) model. The system sizes are  $8 \times 8$  (red squares),  $16 \times 16$  (green circles),  $32 \times 32$  (blue triangles) and  $64 \times 64$  (black diamonds). Fitted to the data points is the power-law dependence (5).



**Fig. 3.** The optimal number of sweeps for a single core  $M_1(L)$  together with its fit. Each data point is interpolated by fitting  $M_{\text{opt}}(p) = M_1/p$  to the estimated  $\bar{M}_{\text{opt}}$  for  $p \leq 32$ .

which can be justified, considering that a random walk through energy space has to depend on the total system size and the number of spin states. This power-law behavior corresponds roughly to the scaling of the multicanonical tunneling times in previous works [1,8]. The explicit functional dependence (5) is characteristic for our specific implementation and will be the basis for our performance study (including larger lattice sizes). It is interesting to notice the prefactor ratio between 8-state Potts and Ising (which is a 2-state Potts model) of about 4, corresponding to the increase in the number of spin states.

Afterwards, we performed parallel MUCA simulations with different numbers of cores for each system size, using  $M_{\text{opt}}(L, p)$  in order to compare the optimal performance at each degree of parallelization. To this end, we considered the speedup for  $p$  cores,

defined in terms of real time  $t_p$  until convergence of the MUCA weights,

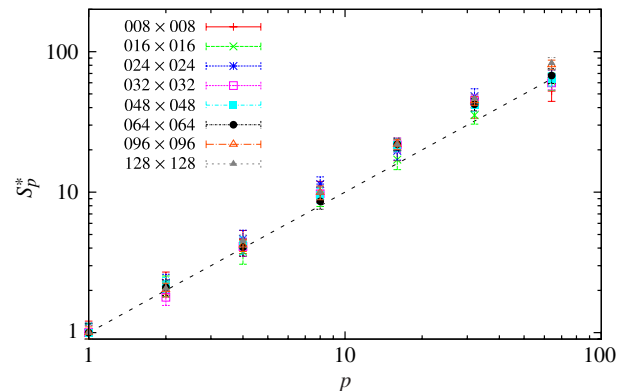
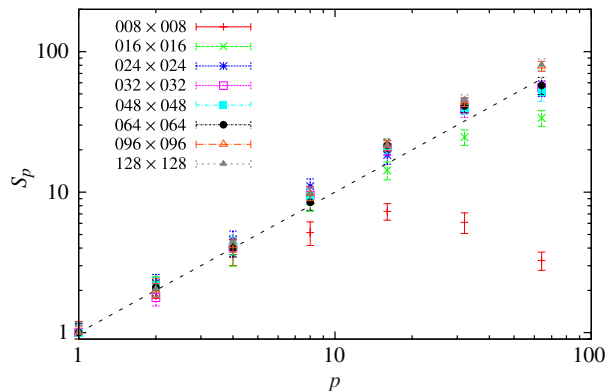
$$S_p = \frac{t_1}{t_p}, \quad (6)$$

as well as the time-independent statistical speedup, defined in terms of total number of sweeps on each core until convergence  $\bar{N}_{\text{iter}}M_{\text{opt}}(L, p)$ ,

$$S_p^* = \frac{[\bar{N}_{\text{iter}}M_{\text{opt}}(L, 1)]_1}{[\bar{N}_{\text{iter}}M_{\text{opt}}(L, p)]_p}, \quad (7)$$

where the subscript indicates the number of cores used. In order to estimate the mean performance, we averaged the required time and number of sweeps over 32 independent PMUCA simulations for each data point. The results for the Ising model are shown in Fig. 4, revealing that the parallel implementation of MUCA results in a linear speedup up to 64 cores already for system sizes  $L \geq 24$ . In case of small system sizes  $L = 8$ , the small  $M_{\text{opt}}$  leads to convergence of MUCA weights within milliseconds, which is difficult to measure precisely. In addition, it can be seen that the statistical speedup scales linearly for all system sizes investigated. This means that the total required number of sweeps until convergence does not increase with the number of cores (compare also Fig. 2(left)). This indicates no slowdown of the parallelization other than from communication, which is kept to a minimum.

In case of the 8-state Potts model, the performance does not scale as linearly with the number of cores, see Fig. 5, but is still satisfying. The reason for the drop in performance may be found in the first-order aspect [9] of the Potts model. In the transition from the disordered to an ordered phase the model undergoes a



**Fig. 4.** Performance for the Ising model over different system sizes expressed by (left) the speedup factor and (right) the time-independent statistical speedup factor.

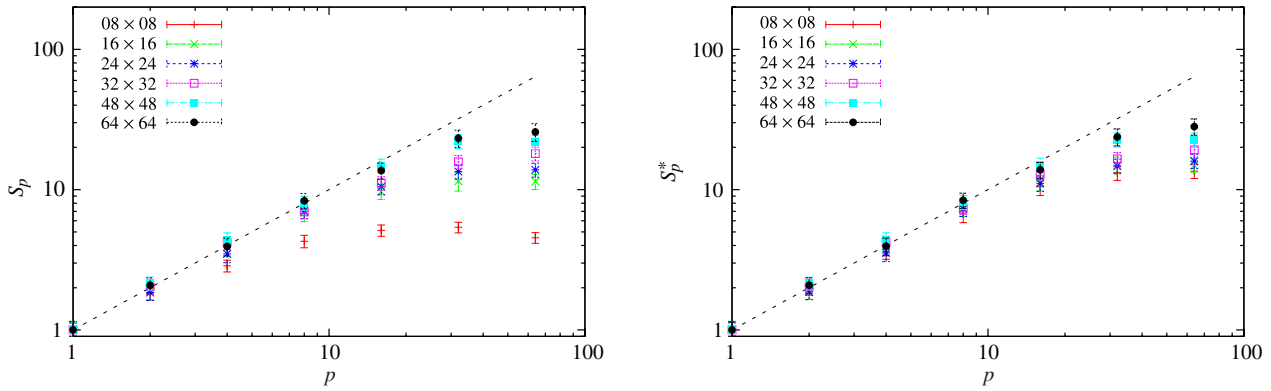


Fig. 5. Performance for the 8-state Potts model over different system sizes expressed by (left) the speedup factor and (right) the time-independent statistical speedup factor.

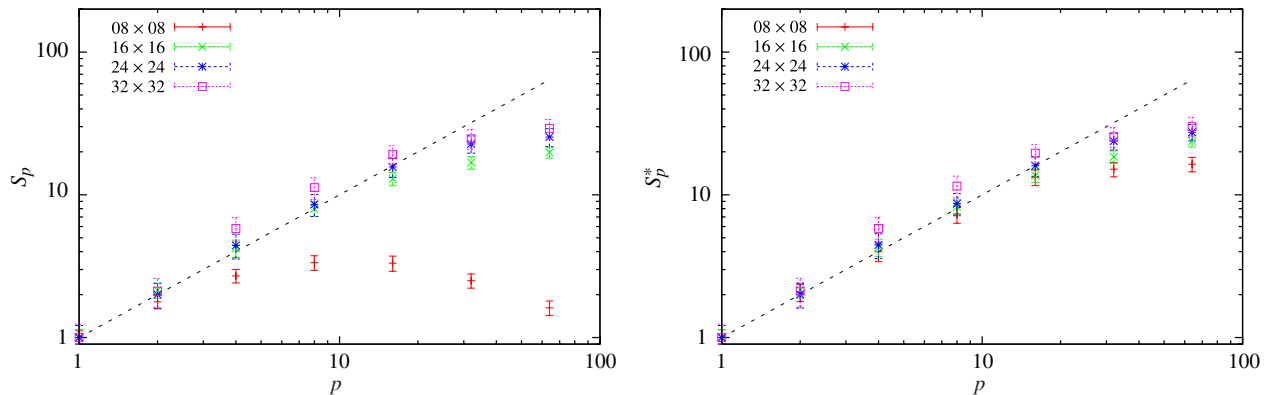


Fig. 6. Performance for the multimagnetic simulation of the Ising model at  $\beta = (3/2)\beta_0$  over different system sizes expressed by (left) the speedup factor and (right) the time-independent statistical speedup factor.

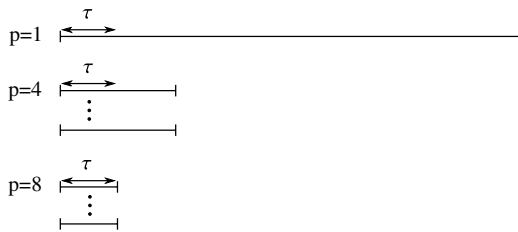


Fig. 7. Scheme of the number of sweeps per iteration and core in comparison to the integrated autocorrelation time  $\tau$ . If the number of sweeps per core gets smaller than the integrated autocorrelation time of the Markov chain, the convergence of the MUCA weights gets worse and the performance drops.

droplet–strip transition [10]. Previous work on multimagnetic simulations of the Ising model [11] showed that such a transition (and also the droplet–condensation transition) is accompanied by “hidden” barriers, which are not directly reflected in the multimagnetic or multicanonical histograms and hence are difficult to overcome. We applied the parallel version of the multicanonical method also to a multimagnetic simulation of the Ising model at  $\beta = (3/2)\beta_0$ , determined  $M_{opt}(p)$  for the lattice sizes  $L = 8, 16, 24, 32$  and measured the speedup factor. The result can be seen in Fig. 6 and shows the same drop in performance as we can observe for the 8-state Potts model.

With this picture in mind, the drop in performance may be explained by the fact that with increasing  $p$  we decreased the number of sweeps per iteration and thus decreased the chance to efficiently cross emerging barriers. When reducing the number of sweeps per iteration as a consequence of parallelization, this reaches a point where the number of sweeps are of the order of the integrated autocorrelation time  $\tau$  and each Markov chain is strongly correlated, see also Fig. 7. Exemplary measurements of

$\tau$  in the 8-state Potts model for different lattice sizes revealed that it was of the order of  $M_{opt}(L, 8)$  to  $M_{opt}(L, 16)$ , verifying the drop in performance for  $p \geq 16$ . This gives a limit of parallelization depending on the barriers and the associated autocorrelation times of the system.

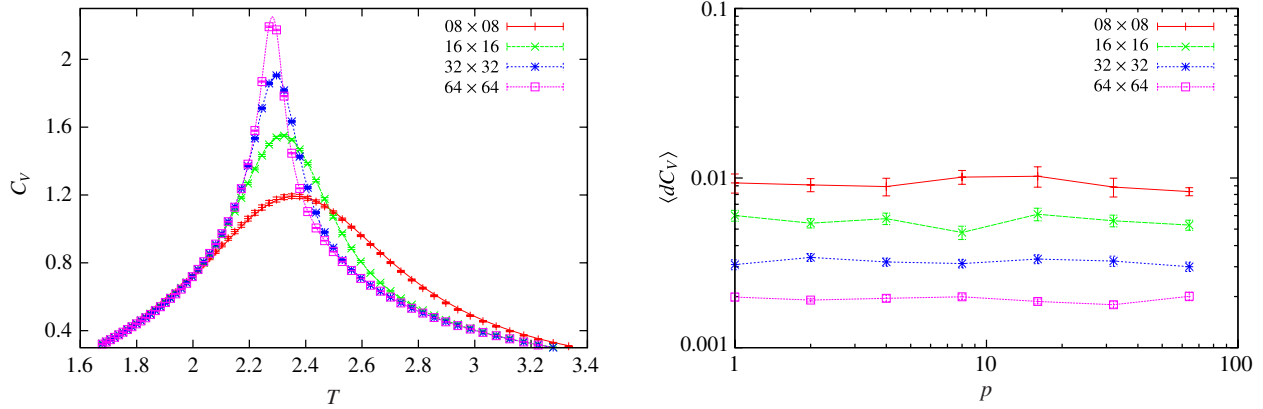
From a practical point of view, when simulating complex systems, it may be advantageous to introduce short thermalization phases between iterations. Exemplary investigations of the 8-state Potts model with intermediate thermalization showed that the number of iterations can be reduced significantly while introducing additional computation time.

### 5. Quality

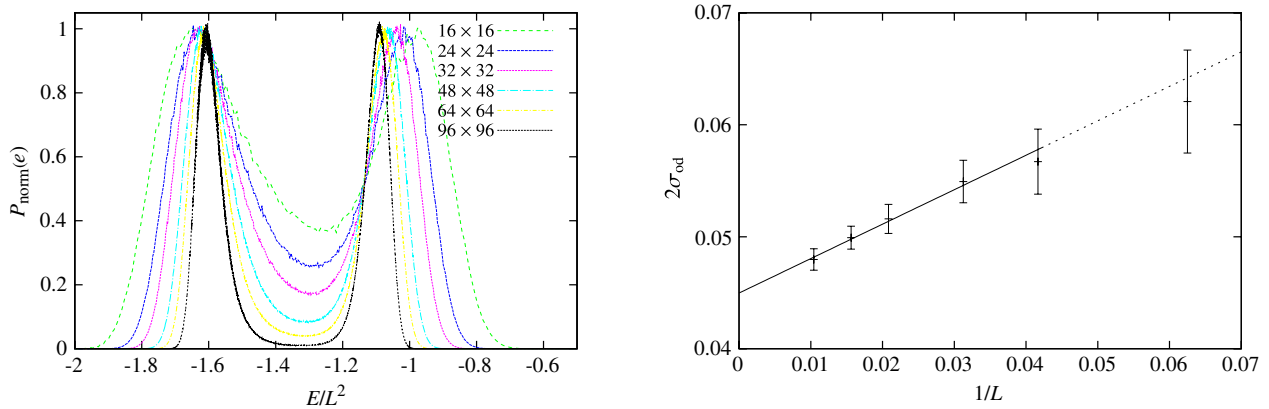
The parallel MUCA weight recursion can be extended by a parallel production run, acquiring data with independent Markov chains. This allows equally good estimation of observables for a constant number of total measurements, if it is ensured that each Markov chain samples the desired phase space appropriately. For the Ising model, we considered the relative deviation from the exact result  $O_{ex}$  [12] and averaged over a temperature range around the critical temperature

$$\langle dO \rangle = (1/N) \sum_{\beta_i} \frac{|O(\beta_i) - O_{ex}(\beta_i)|}{O_{ex}(\beta_i)}. \tag{8}$$

In this case the range was chosen to be  $\beta \in [3/10, 6/10]$  with  $N = 300$  steps. Fig. 8 shows the average deviation of the specific heat  $\langle dC_V \rangle$  ( $C_V = \beta^2(\langle E^2 \rangle - \langle E \rangle^2)/V$ ) for different sizes of the Ising model. The error was estimated by averaging over 16 independent runs. In each simulation there was an additional thermalization phase after the MUCA-weight convergence and the final production run was performed with  $30 \times M_{opt}(L, p)$  measurement points



**Fig. 8.** (Left) Reweighted specific heat from an Ising simulation with 64 cores and the exact results as well as (right) its relative deviation from the exact solution over the number of cores with constant total number of measurements.



**Fig. 9.** (Left) Reweighted normalized mean energy histograms and (right) the scaling of the order–disorder interface tension for the 8-state Potts model simulated on 64 cores.

and 50 sweeps between measurements. The decrease of the relative deviation with increasing lattice size comes from this choice. From Fig. 8(right) it can be seen that, for a given system size, the relative deviations remain constant within the statistical error for all  $p$ .

In order to verify the quality of the parallel simulation of the 8-state Potts model, we estimated the scaling of the order–disorder interface tension  $\sigma_{\text{od}}$  and compared it to analytic results [13]. The interface tension can be approximated in terms of the probability density of the histograms close to the transition temperature [14],

$$\sigma_{\text{od}} = \lim_{L \rightarrow \infty} \frac{1}{2L} \ln \left( \frac{P_{\text{max}}}{P_{\text{min}}} \right)_{\beta_{\text{eqh}}} . \quad (9)$$

This requires us to first find the inverse temperature  $\beta_{\text{eqh}}$  for which the reweighted energy histogram shows two equally high peaks (see Fig. 9(left)) and then to estimate the ratio between the maximum and minimum. Fig. 9(right) shows a rough scaling of the order–disorder interface tension for several system sizes up to  $96 \times 96$ , simulated with 64 cores. The fit to the larger system sizes yields an infinite lattice limit  $\sigma_{\text{od}} \approx 0.045$ , which is consistent with the exact result [13] and verifies that the parallel implementation yields correct results.

## 6. Conclusion

We presented a straightforward parallel implementation of the multicanonical algorithm and showed that its scaling properties with the number of cores are very good for the Ising model and adequate for the 8-state Potts model. The latter one suffers from

emerging barriers at the first-order phase transition, resulting in large integrated autocorrelation times. The parallelization profits from a minimal amount of communication because histograms are merged at the end of each iteration. This is the main reason why it is difficult to adapt this method to weight recursions of the Wang–Landau type where the weights are changed after each spin update. Since it would be interesting to find a similar approach for those weight recursions, not relying on shared memory, we are currently investigating this problem. The major advantage of the implementation employed here lies in the fact that no greater adjustment to the usual implementation is necessary and that additional modifications may be carried along. Thus, it can be easily generalized to complex systems, e.g. (spin) glasses or (bio-) polymers, and allows good convergence if it is ensured that the number of sweeps per core is greater than the integrated autocorrelation times.

## Acknowledgments

We are grateful for support from the Leipzig Graduate School of Excellence GSC185 “BuildMoNa” and the Deutsch-Französische Hochschule (DFH-UFA) under grant no. CDFA-02-07. This project was funded by the European Union and the Free State of Saxony.

## References

- [1] B.A. Berg, T. Neuhaus, Phys. Lett. B 267 (1991) 249; B.A. Berg, T. Neuhaus, Phys. Rev. Lett. 68 (1992) 9.
- [2] W. Janke, Physica A 254 (1998) 164.
- [3] F. Wang, D.P. Landau, Phys. Rev. Lett. 86 (2001) 2050; F. Wang, D.P. Landau, Phys. Rev. E 64 (2001) 056101.

- [4] W. Janke (Ed.), Rugged free-energy landscapes, Lect. Notes Phys. 736 (2008) 203;  
A. Mitsutake, Y. Sugita, Y. Okamoto, Biopolymers (Peptide Science) 60 (2001) 96.
- [5] V.V. Slavin, Low Temp. Phys. 36 (2010) 243;  
A. Ghazisaeidi, F. Vacondio, L.A. Rusch, J. Lightwave Technol. 28 (2010) 79.
- [6] L. Zhan, Comput. Phys. Comm. 179 (2008) 339.
- [7] J. Yin, D.P. Landau, Comput. Phys. Comm. 183 (2012) 1568.
- [8] W. Janke, B.A. Berg, M. Katoot, Nuclear Phys. B 382 (1992) 649.
- [9] W. Janke, in: B. Dünweg, D.P. Landau, A.I. Milchev (Eds.), Computer Simulations of Surfaces and Interfaces, Kluwer, Dordrecht, 2003, p. 111.
- [10] V. Martin-Mayor, Phys. Rev. Lett. 98 (2007) 137207.
- [11] A. Nußbaumer, E. Bittner, T. Neuhaus, W. Janke, Europhys. Lett. 75 (2006) 716;  
A. Nußbaumer, E. Bittner, T. Neuhaus, W. Janke, Phys. Procedia 7 (2010) 52;  
A. Nußbaumer, E. Bittner, W. Janke, Phys. Rev. E 77 (2008) 041109;  
A. Nußbaumer, E. Bittner, W. Janke, Prog. Theor. Phys. Suppl. 184 (2010) 400.
- [12] B. Kaufman, Phys. Rev. 76 (1949) 1232.
- [13] C. Borgs, W. Janke, J. Phys. I France 2 (1992) 2011.
- [14] K. Binder, Phys. Rev. A 25 (1982) 1699;  
W. Janke, Nucl. Phys. B (Proc. Suppl.) 63A-C (1998) 631.