ELSEVIER

Physica A 212 (1994) 261-276

PHYSICA A

Efficient parallelization of the 2D Swendsen-Wang algorithm

R. Hackl [a], H.-G. Matuttis [a,b], J.M. Singer [a], Th. Husslein [a], I. Morgenstern [a]

[a] *Institut für Physik II, Universität Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany*
[b] *HLRZ, c/o KFA Jülich, Postfach 1913, D-52428 Jülich, Germany*

Received 10 July 1994

Abstract

We established a fast Swendsen-Wang algorithm for the two-dimensional Ising model on parallel computers with a high efficiency. On an Intel paragon with 140 processors we reached spin update times of only 14 ns with an efficiency of 89%. This algorithm was used to examine the non-equilibrium relaxation of magnetization and energy in large Ising systems of a size up to 17920 × 17920 spins. Nevertheless we observed still a strong finite-size effect for the magnetization. We assume both magnetization and energy decay to behave like $(t+\Delta)^{-\lambda}\mathrm{e}^{-bt}$ in an infinitely large system. Thus, for long times magnetization and energy show an exponential, asymtotic time-dependence, implying a critical dynamic exponent z of zero.

1. Introduction

Although or even because the Ising model is already solved in two dimensions analytically, it is one of the numerically best investigated models. In spite of its simplicity physicists can gain a lot of insight and experience from it that proves to be fruitful dealing with more complex models.

Hence, especially in the 1980's, the Ising model was explored on more and more powerful computers and architectures. It turned out, however, that at the critical temperature consecutive spin configurations are highly correlated if single spin flip dynamics, like Metropolis or Heat Bath, are used. This effect is known as 'critical slowing down'. So, the Ising model resisted a numerical examination at the critical point. Hence, significantly more precise results could not be achieved just by an increase in computer power. Only the cluster dynamics invented by Swendsen and Wang resolved this problem [1].

SSDI 0378-4371(94)00200-2

The supercomputer architectures of the 80's, the vector computers, seem to reach a similar limit. An enhancement in computational power is possible only with an enormous increase in energy and costs.

Therefore, the new parallel computers, especially the MIMD-systems, are expected to become a promising alternative although they are right at the beginning of their development. The Intel Paragon with 140 nodes for example has a theoretical peak performance of 11.5 GFLOPs and the new 9076 SP 2 of IBM is announced to have a peak performance of 30 GFLOPs with 256 processors.

An algorithm has to fulfill two requirements to use such a parallel architecture efficiently. First it needs a good sequential algorithm as for any single processor machine. Secondly, there must be an efficient way of exchanging relevant data between computation nodes.

The first part will be called 'local algorithm' in the following, as the processor uses only data stored in the local memory, the second part 'global algorithm'. In the latter part, both communication of global data and their processing takes place.

In contrast to the Metropolis algorithm which is not very difficult to parallelize, the Swendsen-Wang algorithm was tackled by several teams in order to find an efficient way of data exchange [2–5], but only M. Flanigan & P. Tamayo succeeded in implementing a communication method – they called it 'relaxation' –, which yields an enhanced computational speed even with several hundreds of processors.

The most reasonable way to parallelize the Swendsen-Wang algorithm is a geometrical parallelization, i.e. cutting up the total system into several local systems, each processed by one node. So far, two ways of decomposition have been used: strip geometry and square geometry. The square geometry is more favorable than strip geometry because the ratio of boundary length to interior area is smaller.

2. The local algorithm

On these local systems we first tried to develop a properly tuned local algorithm which we have already described in [5]. It turned out that the most important points one has to care for in order to optimize performance, are a fast generation of random numbers and an advantageous use of the cache. The latter is not straight forward because read- and write-accesses to the ownerlist, where we store the information of cluster relation, are hardly to predict due to indirect addressing. The cache handling is a crucial factor especially at the Intel 860 RX processor used in the Intel iPSC/860 Hypercube because this processor has a data cache of 4 kbyte in each set only.

As the results produced with different random number generators did not show any relevant difference, we chose the CONG[1] [3] which appeared to be the fastest one. The bit shift random number generator [6] proved to be only slightly slower.

Fig. 1a shows the update time per spin for quadratic systems of various sizes from

[1] multiplication by 16807.

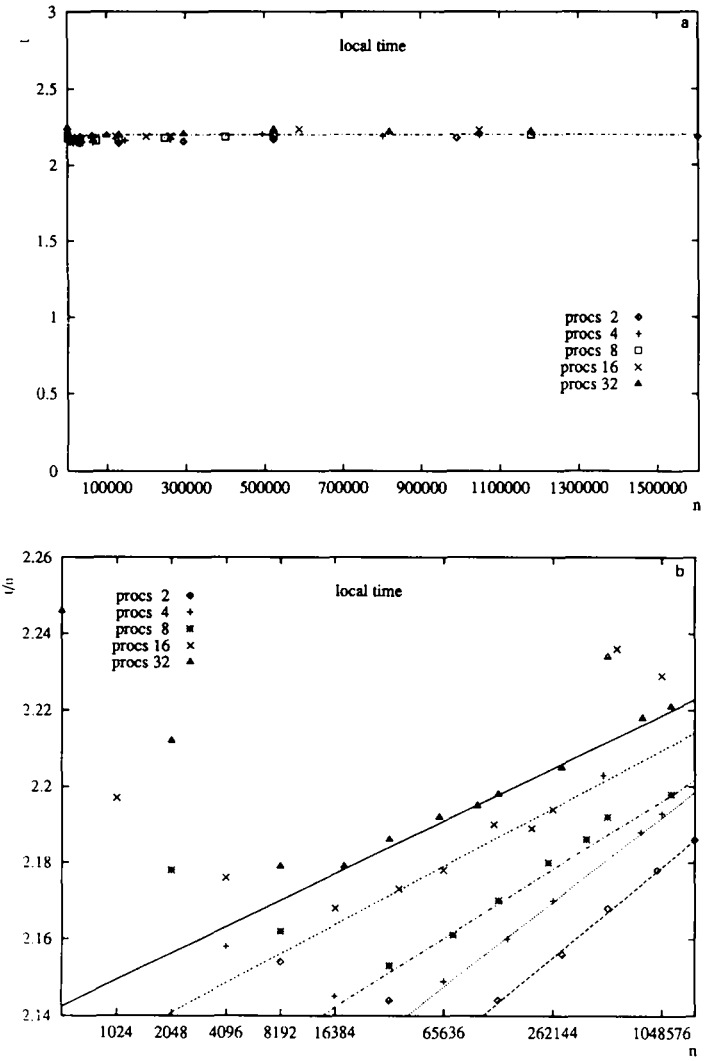

Fig. 1. Local update times per spin versus (a) system size, (b) logarithmic system size.

$n = 128 \times 128$ up to 6144×6144. simulated on an Intel Hypercube. It seems to be roughly constant: $t_{\text{local}}/n = 2.2\,\mu\text{s/spin}$.

A closer view, however, reveals a systematic increase for both small and large grids. In fact, one should expect that the local time behaves like

$$t_{\text{local}} = \alpha n + \beta n \ln n + \gamma \sqrt{n}/p$$

und thus,

$$t_{\text{local}}/n = \alpha + \beta \ln n + \gamma\, \mathrm{e}^{-(0.5 \ln n + \ln p)}$$

where α, β and γ are size-independent constants and p is the number of processors used. The second term is due to iterative accesses to the ownerlist in order to determine the root cluster, which grow in number when the number of clusters increases. The third term originates in the number of spins on the y-boundaries which is of order $\sqrt{n}/p$ in a quadratic system (the x-boundaries are treated during the communication period). This assumption is corroborated by Fig. 1b where the logarithmic behavior can clearly be seen for systems with size n greater than 16000. For small systems the expected exponential decay can not be confirmed beyond doubt because of high statistical fluctuations, but there is no disagreement either.

Surprisingly, the local time seems to be dependant on the number of processors p even for large systems. The reason is that the total system was divided up into p strips of size $l \times (l/p)$, each strip processed by one node. Therefore, the local systems are not quadratic, but their shape depends on the number of nodes. A smaller p causes wider strips and yields a more complex information about the cluster relation in the ownerlist that results in a higher coefficient β. At the same time the total number of local clusters and, hence, the coefficient α decrease, because less random numbers have to be generated.

An additional astonishing point is that the data for system sizes with a linear dimension of 1024 spins or a multiple of it, are higher than expected. This is due to the fact that two labels adjacent in y-direction are stored in memory locations separated by 4096 bytes. As this is just the length of a cache set, they map into the same location in the cache. This results in numerous cache-misses, especially, because the i860 RX has a two-set data cache only.

From this examination we learn that it does not really make sense to state precise update times without mentioning size and form of the system they have been measured with.

We ported our Swendsen-Wang algorithm for a system of size 1024×1024 onto four different parallel computer architectures: An Intel Hypercube, an Intel Paragon, a CM5 of TMC, and an IBM SP1 and reached update times per spin of 2.3 μs/spin, 1.85 μs/spin, 3.9 μs/spin and 0.85 μs/spin, respectively.

3. The communication algorithm

To establish the information of global clusters – clusters extending over more than one processor – we implemented three different communication procedures which we called host-procedure (HP) [5], relaxation-procedure (RP) [4] and binary-tree-procedure (BTP) [2].

For the HP all nodes send spins, labels and the ownerlists of the boundaries to one processor, the so-called host. There the necessary information about the relation of the global clusters is figured out in a process similarly to the local algorithm and stored in the host-ownerlist. The host-ownerlist is finally sent back to the nodes which can decide by this information whether the local part of a global cluster must be flipped or not.

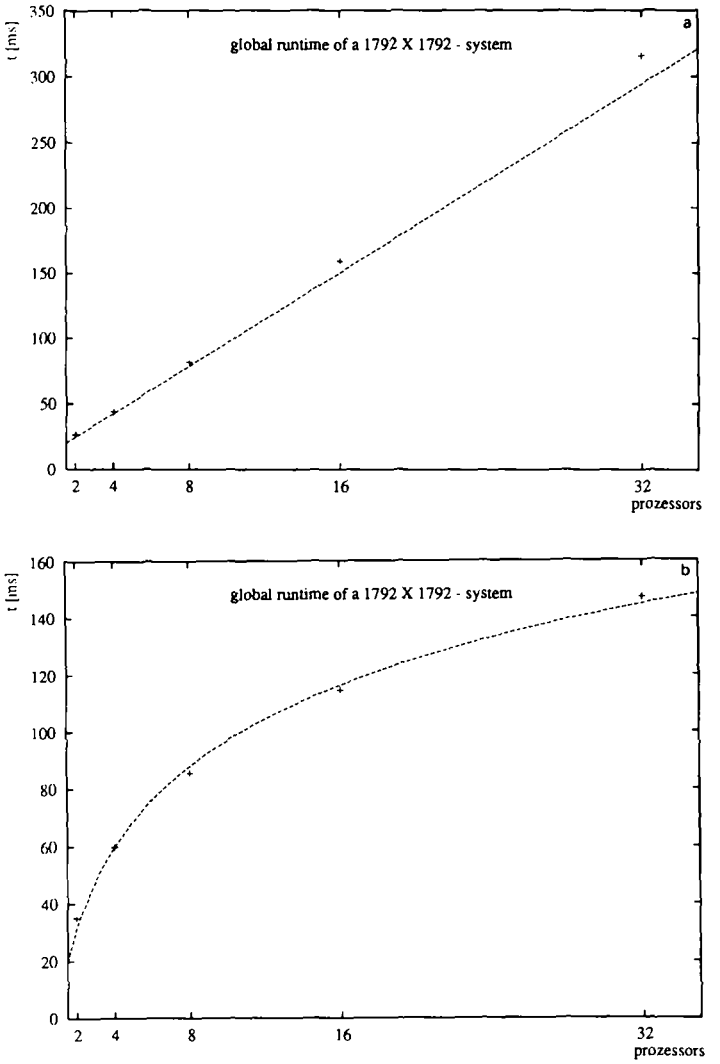

Fig. 2. Global update times with HP (a) , BTP (b) versus number of processors.

For the HP the communication time is proportional to the number of nodes p plus some p-independent time to prepare the information for communication and to evaluate the returned data (Fig. 2a).

The RP was introduced by Flanigan & Tamayo. During this procedure adjacent processors repeatedly exchange information about global clusters and relabel their local boundaries until a fixed state is reached. It turns out that the communication time is roughly proportional to the number of processors p with strip geometry. If the system is split up into quadratic local systems, however, the communication time is proportional to $\sqrt{p}$ only [4].

If we use a strip geometry with n strips, we can group them into $n/2$ pairs of

Table 1
Total update-times per spin for a 6144 × 6144-system on various computers

Machine (processors)	HP	RP	BTP
Hypercube (32)	99.8 ns	78 ns	82 ns
Paragon (128)	-	18 ns	29 ns
CM5 (64)	106 ns	57 ns	71 ns

neighboring strips. After having determined the global clusters of each pair, we end up with $n/2$ strips of cluster distributions. Repeating this process $\log_2 n$ times yields the information about all clusters. This has now to be evaluated in the reverse order to determine whether a local part of a global cluster has to be flipped or not. Theoretically the communication time should be proportional to $\ln p$. This could be confirmed on the Intel Hypercube for up to 32 nodes (Fig. 2b). We are not sure about the reason of the deviation at the value for $p = 2$. But we think that this is a machine dependent effect and not a built-in effect of the algorithm because a similar behavior is found for times of message-broadcasts to several processors.

For the Intel Paragon, however, this is no longer true. As the nodes are arranged in a two-dimensional grid, communication between not directly connected nodes takes longer and prevents the transmitting processors in between from working. This might be changed by the next release of the operating system (OS 1.2), that takes advantage of the coprocessors to communicate data.

So, we can expect the BTP to be the best suited communication procedure, but only if all nodes are homogeneous with respect to communication, as it should be the case on the IBM SP1 due to its switch technology. Unfortunatelly we could only use up to eight processors on such a machine. Hence, no reliable relation between the communication time and the number of processors could be established. With eight nodes the RP and BTP still perform equally well. (See Table 1.)

4. Non-equilibrium relaxation

Based on these algorithms we could investigate the relaxation decay of both magnetization and energy at the critical temperature for a completely magnetized system (all spins pointing up) at $t = 0$, i.e. $M(t = 0) = 1$ [2], with a very high statistical precision. Recently, the relaxation of the magnetization was examined by Kertész & Stauffer [3], Tamayo [7] and Hackl et al. [5]. While Kertész & Stauffer proposed an exponential decay, Tamayo found power-law behavior with his data lying clearly above those of Kertész & Stauffer for times higher than 35 (measured in units of system updates). Tamayo was corroborated by Hackl et al., who got the same results within error bars.

[2] All physical quantities are stated by their value per spin.

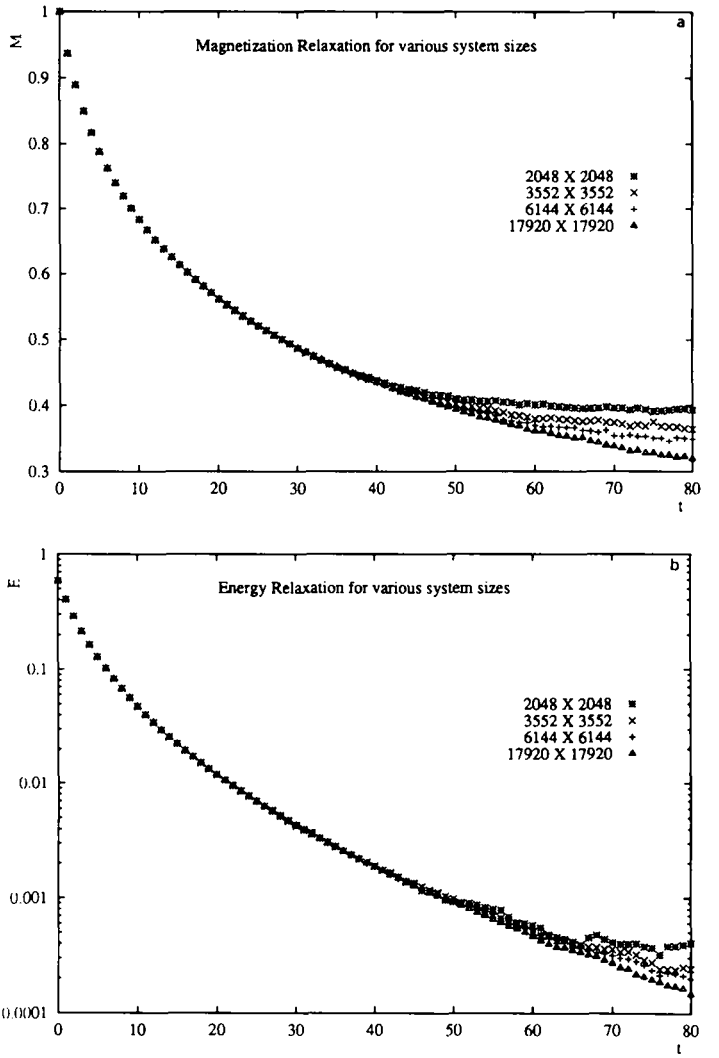

Fig. 3. (a) Magnetization relaxation , (b) energy relaxation, for differently sized systems.

In the case of energy relaxation Ito & Kohring observed a logarithmic singularity, i.e. an asymptotic energy decay of the form $E(t) - E_{eq} \sim t^{-\lambda} \ln t$, for the Wolff dynamics [8].

For the investigation of the non-equilibrium relaxation behavior one has to take into consideration finite-size effects. We simulated systems of size 2048×2048, 3552×3552, 6144×6144 and 17920×17920. For the magnetization we found a similar behavior as Kertész & Stauffer. For small times the data of all sizes coincide. But when time increases, the decay flattens with smaller system sizes (Fig. 3a).

Thus, to reach reliable conclusions for larger times huge systems must be studied. All data presented below refer to the 17920×17920-system.

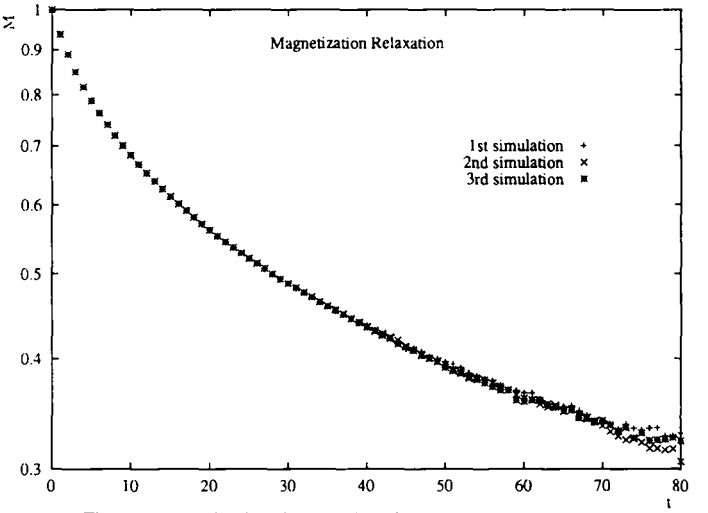

Fig. 4. Magnetization data produced by several simulations.

On the other hand, in the case of the energy decay finite-size effects are comparably small (Fig. 3b).

Usually one is only interested in the long-time behavior of systems. Unfortunately, the statistical fluctuations increase rapidly with time (Fig. 4). Therefore, a prediction for the long-time behavior is hard to derive from the data at large times. Instead, we tried to describe the decay for moderate t as precisely as possible and found that this time-dependency holds for long times, too. To gain small statistical errors we averaged 460 independent relaxation runs.

Figs. 5a and 5b show the effective $z_{eff}(t)$, defined as usual by

$$z^{m}_{eff}(t) = -\frac{\beta}{\nu} \ln\left(\frac{t+1}{t}\right) \Big/ \ln\left(\frac{M(t+1)}{M(t)}\right)$$

and

$$z^{e}_{eff}(t) = (\alpha - 1) \ln\left(\frac{t+1}{t}\right) \Big/ \ln\left(\frac{E(t+1) - E_{eq}}{E(t) - E_{eq}}\right)$$

respectively, where $\alpha = 0$, $\beta = 1/8$ and $\nu = 1$ are the critical exponents for two dimensions and $E_{eq} = -\sqrt{2}$ is the equilibrium energy of an infinite lattice. We can observe z_{eff} decrease linearly in $1/t$ for $4 \leq t \leq 20$, indicating a decay in the form $M(t) \sim (t + \Delta_M)^{-\lambda_M}$ for magnetization and $E(t) - E_{eq} \sim (t + \Delta_E)^{-\lambda_E}$ for energy. According to this ansatz

$$z_{eff}(t) \sim -\frac{1}{\lambda} \frac{\ln(1 + 1/t)}{\ln\left(1 + 1/(t + \Delta)\right)}$$

should asymptotically approach the curve

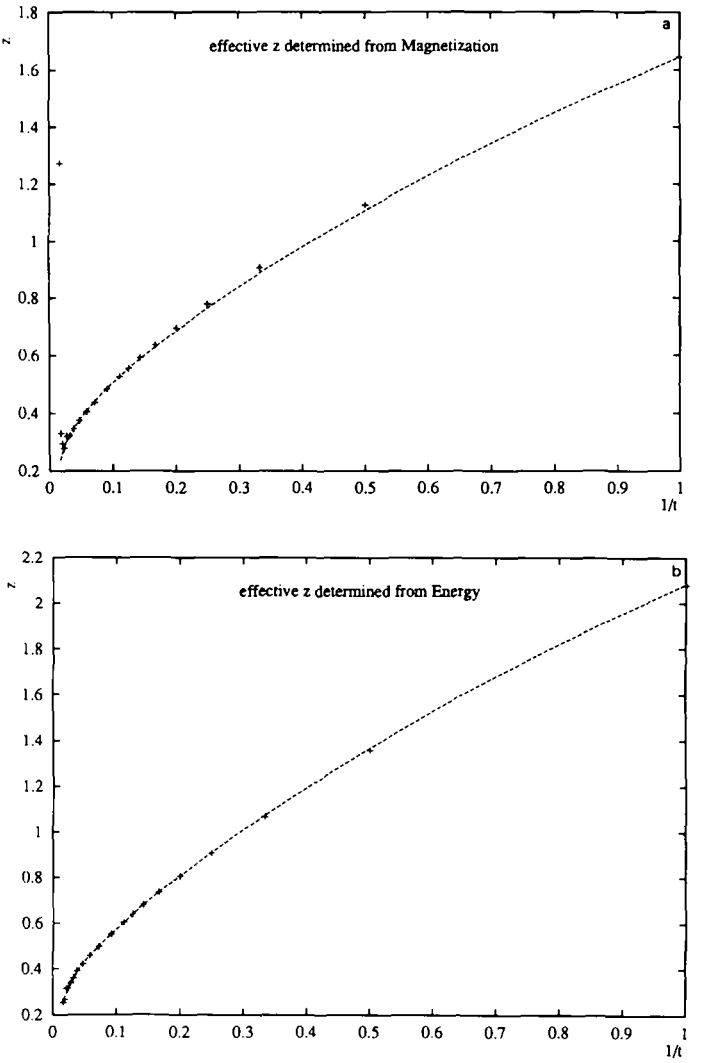

Fig. 5. Effective z from (a) magnetization versus reciprocal time, (b) from energy versus reciprocal time.

$$\frac{\beta}{\nu}\,\frac{1}{\lambda_M}(1+\Delta_M/t) \quad \text{or} \quad (\alpha-1)\,\frac{1}{\lambda_E}(1+\Delta_E/t),$$

respectively, as $1/t$ goes to zero.

However, for $t > 20$ there is a significant deviation from this linear behavior in z_{eff}, telling that the ansatz above is not the whole story. The same can be seen in the results of Tamayo, too. But with his error bars it could not be decided whether this is a significant effect or just a statistical noise.

To explain this we define the function

$$\tilde{b}(t) = \lambda_M \ln\left(\frac{t+1+\Delta_M}{t+\Delta_M}\right) + \ln\left(\frac{M(t+1)}{M(t)}\right)$$

Assuming a time-dependence of the magnetization of the form $M(t) \sim (t+\Delta_M)^{-\lambda_M}/f_M(t)$, we arrive at

$$\tilde{b}(t) = -\ln\left(\frac{f_M(t+1)}{f_M(t)}\right)$$

The energy can be treated in the same way, of course. Looking at Figs. 6a and 6b we see that $\tilde{b}(t)$ is constant for both magnetization and energy in the range $10 \leq t \leq 30$, with

$$b_M = 0.0043 \pm 0.0010, \qquad b_E = 0.0281 \pm 0.0020$$

For larger times the fluctuations become too strong. Thus, no trend can be recognized. The condition, however, for $\tilde{b}(t)$ to be constant must be satisfied by the choice of $f(t)$ as

$$f_M(t) = \mathrm{e}^{b_M t} \quad \text{and} \quad f_E(t) = \mathrm{e}^{b_E t}$$

respectively. Therefore, we propose a decay of magnetization and energy of the form

$$M(t) \sim (t+\Delta_M)^{-\lambda_M}\,\mathrm{e}^{-b_M t}$$

and

$$E(t) - E_{eq} \sim (t+\Delta_E)^{-\lambda_E}\,\mathrm{e}^{-b_E t}$$

However, so far we did not really tell how to determine the coefficients Δ and λ, which we used to attain Figs. 6a and 6b. It turns out not to be a good idea to evaluate Figs. 5a and 5b because the influence of the e^{-bt}-term is too strong to be neglected. At $t = 20$ it is already 0.92 for the magnetization and even 0.57 for the energy. To resolve this problem we define the functions

$$h_M(t) = -\frac{1}{M(t)}\frac{\mathrm{d}M(t)}{\mathrm{d}t}$$

and

$$g_M(t) = \frac{-1}{h_M(t+2) - h_M(2)}$$

and analogously the pair $h_E(t)$ and $g_E(t)$ for the energy. According to the ansatz above we find

$$h_M(t) = \frac{\lambda_M}{t+\Delta_M} + b_M$$

and

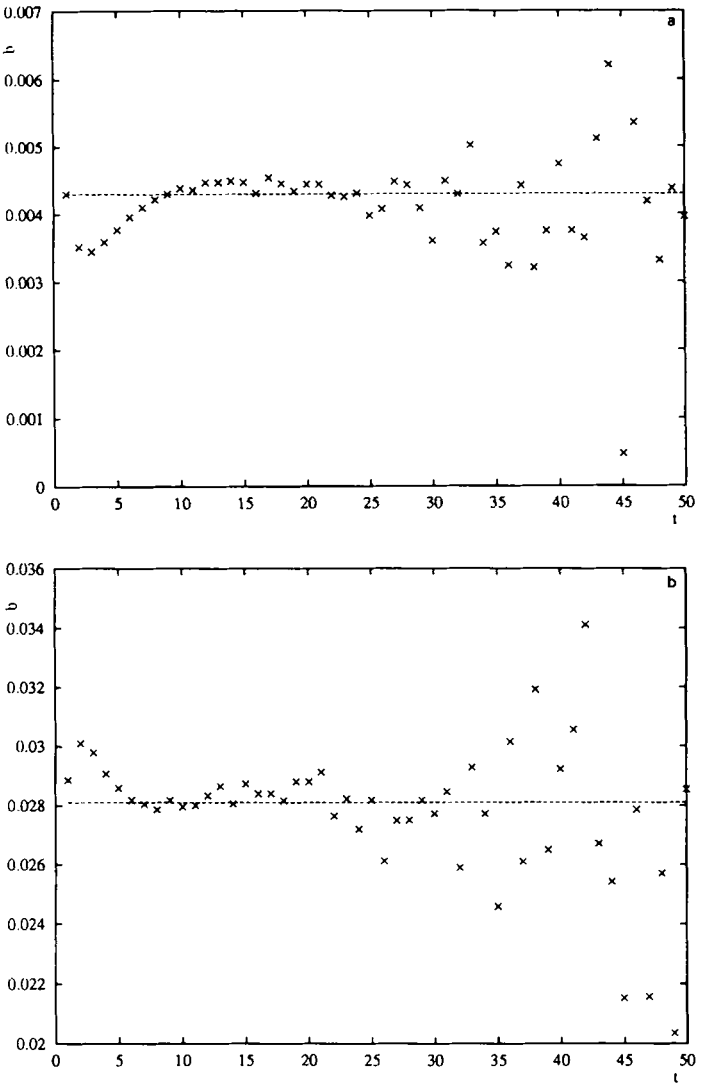

Fig. 6. $b(t)$ (a) from magnetization versus time, (b) from energy versus time.

$$g_M(t) = \frac{1}{\lambda_M}\left[\frac{(\Delta_M + 2)^2}{t} + \Delta_M + 2\right]$$

These equations explain the intention for the definition of h and g. By canceling out b we are left with only two unknown coefficients in g. Although the transformations above are numerically subtle, plotting g versus $1/t$ results in a nice straight line (Figs. 7a and 7b). Evaluating slope and offset yields the parameters λ and Δ,

$$\lambda_M = 0.29 \pm 0.03, \qquad \lambda_E = 2.26 \pm 0.08$$
$$\Delta_M = 4.5 \pm 0.6, \qquad \Delta_E = 5.9 \pm 0.25$$

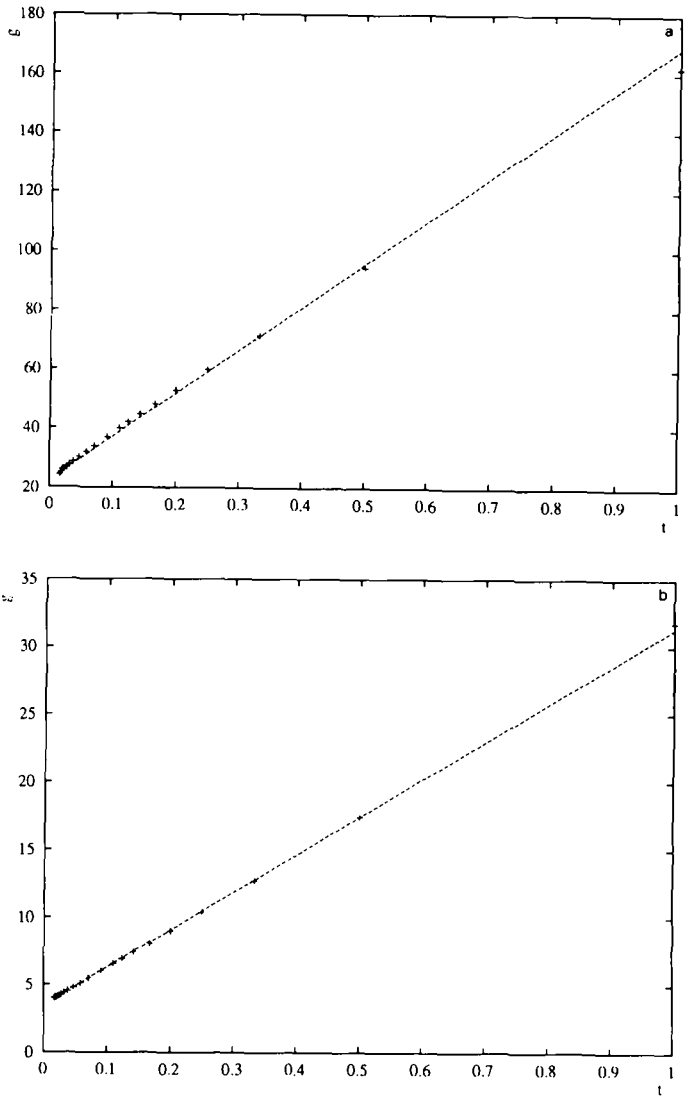

Fig. 7. $g(t)$ (a) from magnetization versus reciprocal time, (b) from energy versus reciprocal time.

Furthermore, it proved to be numerically advantageous to determine b not from $\tilde{b}(t)$. Considering that we know $M(t=0)=1$ and $E(t=0)=-2$, we better use

$$b_M(t) = -\frac{1}{t}\left[\ln\left(\frac{M(t)}{M(0)}\right) + \lambda_M \ln\left(\frac{t+\Delta_M}{\Delta_M}\right)\right]$$

and

$$b_E(t) = -\frac{1}{t}\left[\ln\left(\frac{E(t)-E_{eq}}{E(0)}\right) + \lambda_E \ln\left(\frac{t+\Delta_E}{\Delta_E}\right)\right]$$

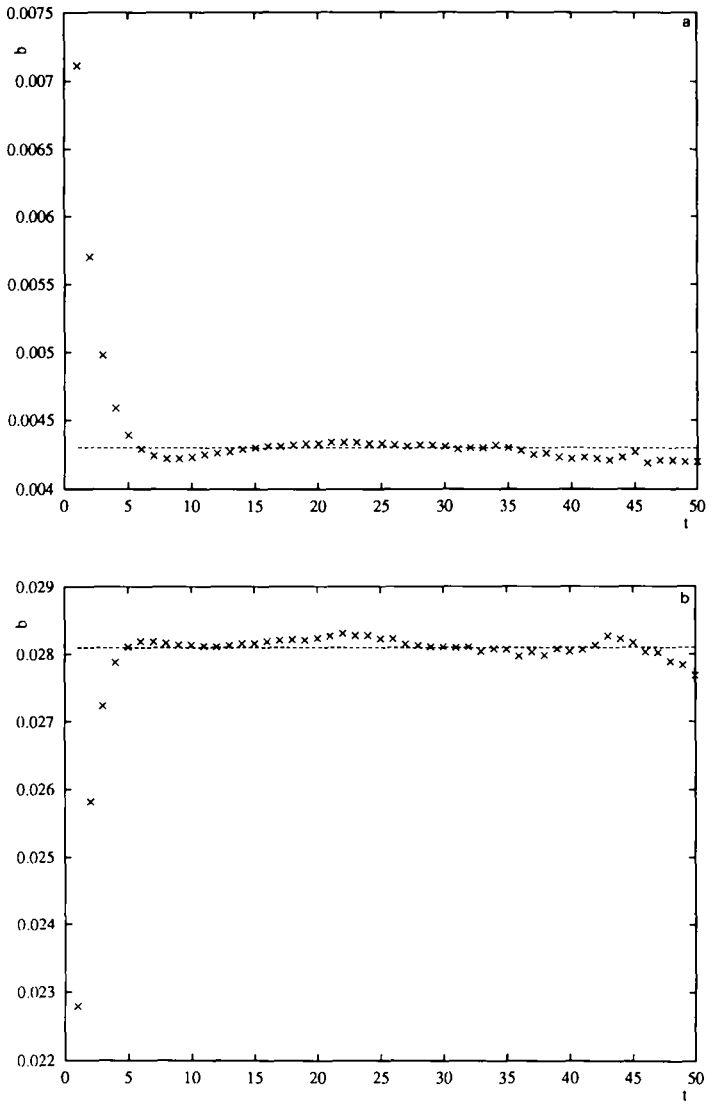

Fig. 8. $b(t)$ (a) from magnetization, (b) from energy, versus time.

instead. Figs. 8a and 8b show $b(t)$ to be constant within a wide range of t with only moderate fluctuations.

Taking into consideration that the statistical fluctuations increase rapidly for $t > 50$, this ansatz fits very well the data of the simulations from $t = 0$ on (Figs. 9a and 9b). At about $t = 50$ the theoretical curve leaves the data points in Fig. 9a. This had to be expected because we know from Fig. 3a that the magnetization data are strongly influenced by finite-size effects.

As for power-law behavior the exponents λ_M and λ_E can be expressed by the dynamical critical exponent z, i.e. $\lambda_M = \frac{\beta}{\nu z}$ and $\lambda_E = \frac{(1-\alpha)}{z}$, we can determine z from

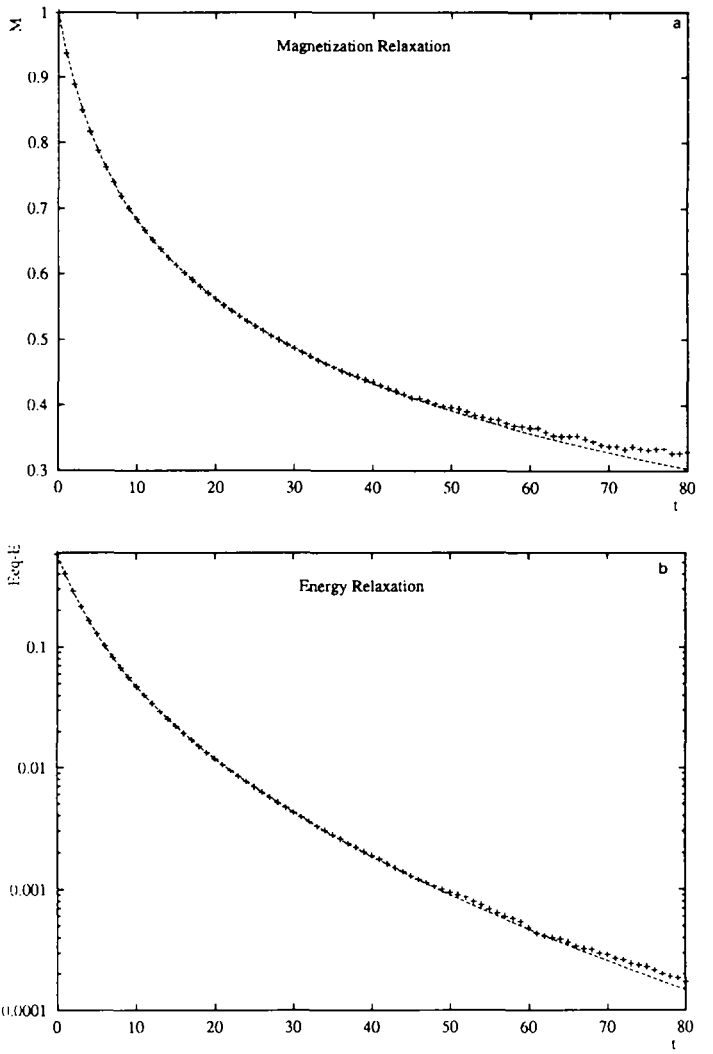

Fig. 9. Magnetization decay (a), energy decay (b), and proposed theoretical curves.

the magnetization and energy as 0.43 and 0.44, respectively. Although Tamayo's data agree excellently with ours, he got $z = 0.25 \pm 0.05$ assuming a pure power-law. This is reasoned by the influence of the $e^{-b_M t}$-factor.

The proposed behavior is not so surprising if we remember that for the spin correlation in spin glasses the Kohlrausch decay, i.e. $s(t)s(0) \sim e^{-bt^x}/t^\lambda$, was already established, which is very similar except for the additional parameter x [10].

These results do not indicate any logarithmic singularity for the energy decay in the case of Swendsen-Wang dynamics in contrast to Wolff dynamics. This is in agreement with Siegert & Stauffer who did not find an $\ln(t)$-factor in the energy relaxation of the 2D and 3D Glauber dynamics either [11]. To reveal a $t^{-\lambda}\ln(t)$-behavior one usually

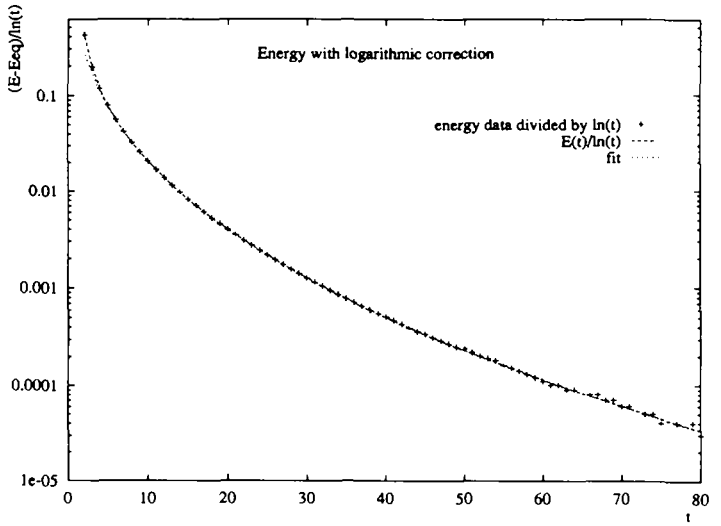

Fig. 10. Decay of $(E(t) - E_{eq})/\ln(t)$ and fit-curves.

plots the data divided by $\ln t$.

Fig. 10 displays the energy data in the form $(E(t) - E_{eq})/\ln(t)$ together with two fit curves, i.e. the ansatz from above divided by $\ln t$

$$\frac{2-\sqrt{2}}{\Delta_E^{-\lambda_E}} \frac{(t+\Delta_E)^{-\lambda_E} e^{-b_E t}}{\ln t}$$

with the coefficients determined above and a pure fit ansatz (not divided by $\ln t$)

$$\frac{0.00396\, e^{20\tilde{b}}}{(\tilde{\Delta}+20)^{-\tilde{\lambda}}} (t+\tilde{\Delta})^{-\tilde{\lambda}} e^{-\tilde{b}t}$$

with $\tilde{\Delta} = 2.54$, $\tilde{\lambda} = 2.3$ and $\tilde{b} = 0.030$. The second curve is normalized by the first factor in order to pass through the data point at $t = 20$. Although the two curves are based on slightly different parameter sets, they are hardly to distinguish in the range $5 < t \leq 80$ and both fit the data very nicely in this range.

This picture (Fig. 10) teaches not to trust a single fit blindly. More important is an overall consistency in all kinds of data presentation. This is not the case for the second curve, because $\tilde{b}(t)$ in Fig. 11, determined in an analoguos manner as for Fig. 6b, but with the parameters $\tilde{\Delta}$ and $\tilde{\lambda}$ and the energy data divided by $\ln t$, varies much stronger than in Fig. 6b although the y-range is twice as large as in Fig. 6b, and $\tilde{b}(t)$ does not at all look like a constant. In contrast, the power-times-exponential-ansatz for magnetization and energy (represented by the dashed lines) fits all the data displayed in Figs. 5a–9b reasonably well. Therefore, we favor our ansatz without a logarithm for describing the relaxation of the energy over the ansatz including the theoretically expected logarithm.

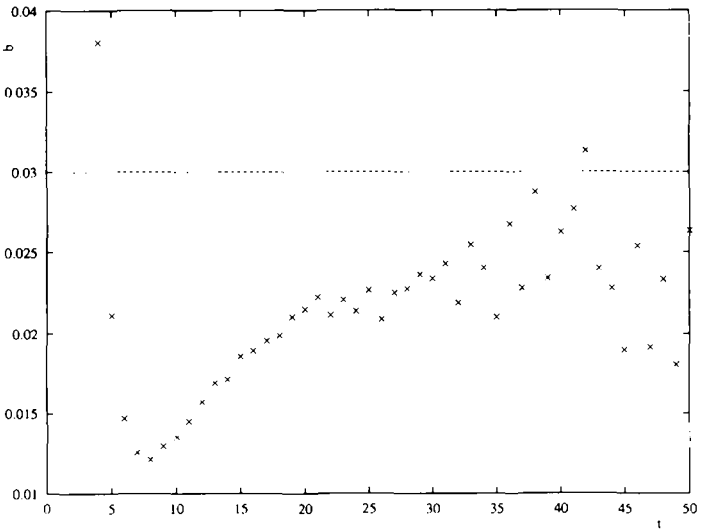

Fig. 11. $\tilde{b}(t)$ from $(E(t) - E_{eq})/\ln(t)$-data.

Acknowledgement

We thank the ZAM at the KFA-Jülich for extensive computation time and support by its staff. Representatively for many helpful members one of us (R.H.) wants to mention Jutta Docter who made working at the paragon a pleasure by her continuous advice. Also D. Stauffer has to be gratefully acknowledged who initiated the second part of this article and was always ready to give advice when we had problems.

References

[1] R.H. Swendsen and J. Wang, Nonuniversal Critical Dynamics in Monte Carlo Simulations Phys. Rev. Lett. 58 (1987) 86–88.
[2] D.W. Heermann and A.N. Burkitt, Parallel Algorithms in Computational Science(Springer, 1991).
[3] J. Kertész and D. Stauffer, Swendsen-Wang Dynamics on Large 2D Critical Ising Models, Int. J. Mod. Phys. C 3 (1992) 1275–1279.
[4] M. Flanigan and P. Tamayo, A Parallel Cluster Labeling Method for Monte Carlo Dynamics, Int. J. Mod. Phys. C 3 (1992) 1235–1249.
[5] R. Hackl, H.-G. Matuttis, J.M. Singer, T. Husslein and I. Morgenstern, Parallelisation of the 2D Swendsen-Wang Algorithm, Int. J. Mod. Phys. C 4 (1993) 1117–1130.
[6] N. Ito, Monte Carlo Study of The Ising Model, Thesis (1990) p. 45.
[7] P. Tamayo, Magnetization relaxation to equilibrium on large 2D Swendsen-Wang Ising models, Physica A 201 (1993) 543–546.
[8] N. Ito and G. Kohring, Non-equilibrium critical relaxation of the Wolff dynamics of the Ising model Physica A 201 (1993) 547–556.
[9] C. Münkel and D.W. Heermann, J. Adler, M. Gofman and D. Stauffer, The dynamical critical exponent of the two-, three-, and five-dimensional kinetic Ising model, Physica A 193 (1993) 540–552.
[10] A.T. Ogielski, Phase Transitions and Equilibrium Dynamics in Strongly Random Ising Spin Systems, in: Heidelberg Colloquium on Glassy Dynamics, edited by J.L.van Hemmen and I.Morgenstern (Springer, 1986) p. 202.
[11] M.Siegert and D.Stauffer, Energy Relaxation in Ising Models, Physica A 208 (1994) 540.