Numerical simulation study of the dynamical behavior of the Niedermayer algorithm

J. Stat. Mech. (2010) P04012

(http://iopscience.iop.org/1742-5468/2010/04/P04012)

Journal of Statistical Mechanics: Theory and Experiment
An IOP and SISSA journal

Numerical simulation study of the dynamical behavior of the Niedermayer algorithm

D Girardi and N S Branco

Departamento de Física, Universidade Federal de Santa Catarina, 88040-900, Florianópolis, SC, Brazil
E-mail: daniel@fisica.ufsc.br and nsbranco@fisica.ufsc.br

Received 10 February 2010
Accepted 18 March 2010
Published 16 April 2010

Online at stacks.iop.org/JSTAT/2010/P04012
doi:10.1088/1742-5468/2010/04/P04012

Abstract. We calculate the dynamic critical exponent for the Niedermayer algorithm applied to the two-dimensional Ising and XY models, for various values of the free parameter E_0. For $E_0 = -1$ we regain the Metropolis algorithm and for $E_0 = 1$ we regain the Wolff algorithm. For $-1 < E_0 < 1$, we show that the mean size of the clusters of (possibly) turned spins initially grows with the linear size of the lattice, L, but eventually saturates at a given lattice size $\widetilde{L}$, which depends on E_0. For $L > \widetilde{L}$, the Niedermayer algorithm is equivalent to the Metropolis one, i.e., they have the same dynamic exponent. For $E_0 > 1$, the autocorrelation time is always greater than for $E_0 = 1$ (Wolff) and, more important, it also grows faster than a power of L. Therefore, we show that the best choice of cluster algorithm is the Wolff one, when comparing against the Niedermayer generalization. We also obtain the dynamic behavior of the Wolff algorithm: although not conclusively, we propose a scaling law for the dependence of the autocorrelation time on L.

Keywords: classical Monte Carlo simulations, correlation functions (theory)

Contents

1. Introduction

Numerical simulations have been widely used in the study of physical systems, especially in the last few decades. The field of statistical mechanics, among others, has benefited a great deal from the use of this technique. In particular, Monte Carlo methods allowed for a precise determination of thermodynamic parameters in a variety of models, both classical and quantum. Excellent reviews on these methods can be found in [1, 2].

In recent years, this field has seen a fast development of new algorithms, which aim to make the simulation more efficient, both in time and in memory, as well as broadening its application to more complex systems. As examples of these developments, we can recall: the calculation of the density of states through flat histograms, which allows obtaining information at any temperature from one single simulation, independent of temperature [3]; the use of bitwise operations and storage, which increases by a great deal the speed of the update process and saves memory (with the drawback that this procedure can be used only with specific models) [4]; and the introduction of cluster algorithms, which updates collections of spins, decreasing the autocorrelation time and almost eliminating critical slowing down [1, 2, 5, 6].

In this work we will focus on this last issue. In fact, critical slowing down is a serious drawback, which makes simulation of systems at, or near, critical points very inefficient. This phenomenon is measured through the scaling of the autocorrelation time, τ, with the linear size of the lattice, L, assumed to be in the form $\tau \sim L^z$, for points at the critical region. The popular Metropolis algorithm, for example, when applied to the Ising model in two dimensions, presents $z \sim 2.17$ [7]. Algorithms which update clusters of spins (the so-called cluster algorithms) have a much lower value of z: this is the case for the Swendsen–Wang [5] and Wolff [6] algorithms, for which z is approximately zero for the two-dimensional Ising model [8, 9].

An alternative to (and generalization of) these last two cluster algorithms, the Niedermayer algorithm, was introduced some time ago [10] but, to the best of our

J. Stat. Mech. (2010) P04012

knowledge, has never had its dynamic behavior studied in detail. In this work, we calculate the dynamic exponent for this algorithm, applied to the Ising and XY models, for some values of the free parameter E_0 (see below), in order to determine the best choice of this parameter.

This work is organized as follows. In section 2 we present the Niedermayer algorithm and relate it to the Metropolis and Wolff ones. In section 3 we review some features connected to the autocorrelation time and the dynamic exponent z, in section 4 we present and discuss our results, and in section 5 we summarize the results.

2. The Niedermayer algorithm

The Niedermayer algorithm was introduced some time ago and is an alternative to Wolff and Swendsen–Wang cluster algorithms. The idea is to build clusters of spins and accept their updating as a single entity, one hopes in a more efficient way, when compared to these last two algorithms. In this work, we have chosen to build the clusters according to the Wolff criterion (they can be constructed according to the Swendsen–Wang rule but the results will not differ qualitatively in two dimensions, and in higher dimensions the Wolff algorithm is superior to the Swendsen–Wang one). It works as follows, for the Ising model (the generalization of this algorithm for the XY model is presented in the appendix): a spin in the lattice is randomly chosen to be the first spin of the cluster. This spin is called the *seed*. First neighbors of this spin may be considered part of the cluster, with a probability

$$P_{\mathrm{add}}(E_{ij}) = \begin{cases} 1 - \mathrm{e}^{K(E_{ij}-E_0)}, & \text{if } E_{ij} < E_0, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where $K = J/kT$, T is the temperature, J is the exchange constant, and E_{ij} is the energy between nearest-neighbor spins in units of J (i.e., $E_{ij} = -s_i s_j$; $s_i, s_j = \pm 1$). The free parameter E_0 controls the size of the clusters and the acceptance ratio of their updating, as seen below. First neighbors of added spins may be added to the cluster, according to the probability given above. Each spin has more than one chance of being part of the cluster, since it may have more than one first neighbor in it. When no more spins can be added, all spins in the cluster are flipped with an acceptance ratio A. Assuming that at the frontier of the cluster there are m bonds linking parallel spins and n bonds linking anti-parallel spins, A satisfies

$$\frac{A(a \to b)}{A(b \to a)} = \left[\mathrm{e}^{2K} \left(\frac{1 - P_{\mathrm{add}}(-J)}{1 - P_{\mathrm{add}}(J)} \right) \right]^{n-m}, \tag{2}$$

where $a \to b$ represents the possible updating process, from the 'old' (a) to the 'new' (b) state, which differs from the flipping of all spins in the cluster, and $b \to a$ represents the opposite move. This expression ensures that detailed balance is satisfied [2].

Now we must consider three cases:

(i) For $-1 \leq E_0 < 1$, only spins in the same state as the seed may be added to the cluster, with probability $P_{\mathrm{add}} = 1 - \mathrm{e}^{-K(1+E_0)}$. The acceptance ratio (equation (2)) cannot be chosen to be 1 always and is given by $A = \mathrm{e}^{-K(1-E_0)(m-n)}$ if $n < m$ (i.e., if the energy increases when the spins in the cluster are flipped), or by $A = 1$ if $n > m$ (i.e., if the

J. Stat. Mech. (2010) P04012

energy decreases when the spins in the cluster are flipped). If $E_0 = -1$, we obtain the Metropolis algorithm, since only one-spin clusters are possible and the acceptance ratio is $A = \mathrm{e}^{-K\Delta E}$ for positive ΔE and 1 otherwise, where $\Delta E = 2(m - n)$ is the difference in energy when the spin is flipped, in units of J.

(ii) For $E_0 = 1$, again only spins in the same state can take part in the cluster, with probability $P_{\mathrm{add}} = 1 - \mathrm{e}^{-2K}$. Now, the acceptance ratio can be chosen to be 1, i.e., the cluster of parallel spins is always flipped. This is the celebrated Wolff algorithm.

(iii) For $E_0 > 1$, spins anti-parallel to the seed may be part of the cluster, with probability $P_{\mathrm{add}} = 1 - \mathrm{e}^{K(1-E_0)}$, while spins in the same state of the seed have a probability $P_{\mathrm{add}} = 1 - \mathrm{e}^{-K(1+E_0)}$ of being added to the cluster. The acceptance ratio is again always 1. Note that for $E_0 \gg 1$ nearly all spins will be in the cluster and the algorithm will be clearly inefficient (in fact, it will not be ergodic for $E_0 \to \infty$). Therefore, we expect that, if the optimal choice of E_0 is greater than 1, it will not be much greater than this value.

Our goal here is to do a systematic study of the Niedermayer algorithm, in order to establish the optimal value for E_0, at least for the two models addressed in this text.

3. The autocorrelation time and dynamic exponent

One possible way to access the dynamic behavior of a numerical algorithm is to measure the autocorrelation time, τ, of some convenient physical quantity, which is obtained from the dependence of the autocorrelation function, $\rho(t)$, on the time t. Here, time is measured in Monte Carlo steps (MCS); one MCS is defined as the attempt to flip N spins, where N is the number of spins in the (finite) lattice being simulated (in our case, $N = L^2$, where L is the linear size of the lattice). In fact, a rescaling of the time is necessary, when dealing with cluster algorithms [2] and comparing the results for different values of E_0. The relation between 'time' in MCS, t_{MCS}, and the 'time' taken to build and possibly flip a cluster, t, is

$$t_{\mathrm{MCS}} = t\frac{\langle n\rangle}{N}, \tag{3}$$

where $\langle n\rangle$ is the mean size of the clusters. Note that, for Metropolis, $\langle n\rangle = 1$ and 1 MCS is the 'time' taken to try to flip N spins, as usual.

In this work, this rescaling has been done and all times are expressed in MCS. The function $\rho(t)$ is defined as

$$\begin{aligned}\rho(t) &= \int \left[\Phi(t') - \langle\Phi\rangle\right]\left[\Phi(t'+t) - \langle\Phi\rangle\right] \mathrm{d}t' \\ &= \int \left[\Phi(t')\Phi(t'+t) - \langle\Phi\rangle^2\right] \mathrm{d}t',\end{aligned} \tag{4}$$

where $\Phi(t)$ is some physical quantity. Of course, time is a discrete quantity in the simulations; therefore, we have to discretize the previous equation, which leads to [2]

$$\rho(t) = \frac{1}{t_{\max} - t}\sum_{t'=0}^{t_{\max}-t}\left[\Phi(t')\Phi(t'+t)\right] - \frac{1}{(t_{\max}-t)^2}\sum_{t'=0}^{t_{\max}-t}\Phi(t') \times \sum_{t'=0}^{t_{\max}-t}\Phi(t'+t). \tag{5}$$

The autocorrelation function is expected to behave, as a function of time, as [2]

$$\rho(t) = A\mathrm{e}^{-t/\tau}, \tag{6}$$

at least in its simplest form. It is known that, in some cases, more than one exponential term is required [11]; we will comment on this later. Usually, one can measure τ from the slope of an adjusted straight line in a semi-log plot of the autocorrelation function versus time. However, the autocorrelation function is not well behaved for long times, due to bad statistics (this is evident from equation (5), since few 'measurements' are available for long times). Therefore, one has to choose the region where the straight line will be adjusted very carefully and it turns out that the value of τ so obtained is strongly dependent on this choice. Alternatively, one can integrate $\rho(t)$, assuming a single-exponential dependence on (past and forward) time, and obtain

$$\tau = \frac{1}{2}\int_{-\infty}^{\infty} \frac{\rho(t)}{\rho(0)}\,\mathrm{d}t, \tag{7}$$

with

$$\rho(t) \equiv \mathrm{e}^{-|t|/\tau}. \tag{8}$$

Equation (7), when discretized, leads to [12]

$$\tau = \frac{1}{2} + \sum_{t=1}^{\infty} \frac{\rho(t)}{\rho(0)}. \tag{9}$$

Of course, the sum in equation (9) cannot be carried out for large values of t. It has to be truncated at some point; we use a cutoff (see [12] and references therein), defined as the value in time where the noise in the data is clearly greater than the signal itself. With the value of τ obtained as explained above, we made the integral of $\rho(t)/\rho(0)$ from the value of the cutoff to infinity. A criterion for accepting the cutoff is that the value of this integral is smaller than the statistical uncertainty in calculating τ. Since the value that we obtain for τ is underestimated, this criterion is a safe one.

Whenever possible, we fitted the autocorrelation time to the expected behavior, namely $\tau \sim L^z$, in the critical region, where z is the dynamic exponent. A point worth mentioning is that the autocorrelation functions of different quantities may behave in different ways. A typical example is shown in figure 1, where both the magnetization and the energy autocorrelation functions are depicted as functions of time, for the Niedermayer algorithm with $E_0 = 0.3$ and linear sizes $L = 16$ (main graph) and $L = 256$ (inset). Note the abrupt drop of the magnetization autocorrelation function for small times and $L = 16$. This is an indication that this function is not properly described by a single exponential. On the other hand, the energy autocorrelation time follows a straight line even for the smallest times. Therefore, we should calculate τ from the latter, for $L = 16$, using equation (9). However, when L is increased, the picture changes and now the magnetization autocorrelation function is well described by a single exponential (for small and intermediate values of time), as depicted in the inset of figure 1. Whenever a crossover like this is present, we measure the dynamic exponent from the behavior for large values of L and for the function which is well described by a single exponential for this range of L, using equation (9). But note that, for intermediate values of t, the slopes of both curves in figure 1 (main graph and inset) appear to be the same. However, we have

doi:10.1088/1742-5468/2010/04/P04012

J. Stat. Mech. (2010) P04012

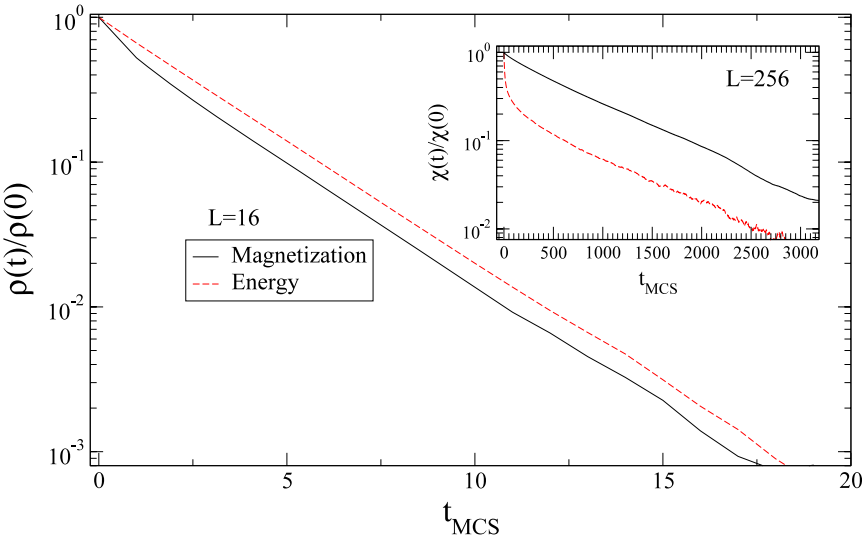

Figure 1. Magnetization and energy autocorrelation functions versus time (in MCS) for the Niedermayer algorithm with $E_0 = 0.3$ (see the text). The main graph represents the behavior for linear size $L = 16$, while the inset applies to $L = 256$.

already commented on the drawback of calculating τ from the slope of the autocorrelation function on a semi-log graph. As final notes, we would like to mention that we used helical boundary conditions and 20 independent runs (each with a different seed for the random number generator) were made for each E_0 and L. For each seed, at least 4×10^6 trial flips were made, in order to calculate the autocorrelation functions and their respective autocorrelation times. The values that we quote are the averages of the values obtained for each seed of the random number generator and the uncertainty in τ is the standard deviation of these 20 values.

4. Results and discussion

4.1. Ising model

We first present our results for the Ising model and leave to section 4.2 the discussion of the results for the XY model.

As already discussed, the case $E_0 = -1$ corresponds to the Metropolis algorithm. At the critical temperature, the autocorrelation time scales with L as $\tau \sim L^z$, with $z = 2.1665 \pm 0.0012$ [7]. We have simulated this case only as a test for our algorithm. The value that we found for z is consistent with the one quoted above and the scaling law is obeyed, even for the smallest values of L that we simulated. Note also that, for the Metropolis algorithm ($E_0 = -1$), it is the magnetization autocorrelation time which is well described by a single exponential.

The first non-trivial value of E_0 that we simulated was -0.9. In figure 2 the autocorrelation times for the magnetization are depicted as a function of L. We note that, for this value of E_0, only the autocorrelation function for the magnetization is well described by a single exponential. The initial decay of the corresponding function for the

J. Stat. Mech. (2010) P04012

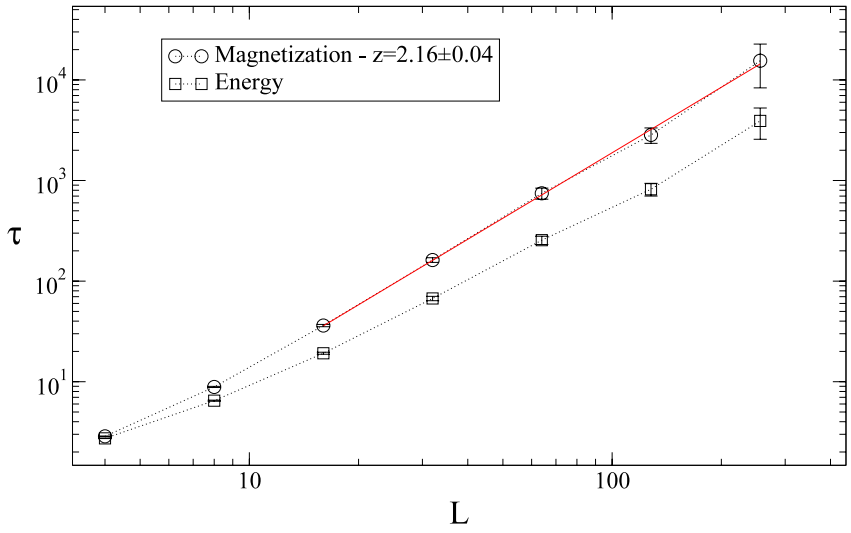

Figure 2. Log–log graphs of the magnetization (○) and energy (□) autocorrelation time (in MCS) versus linear size L for the Niedermayer algorithm with $E_0 = -0.9$. The quoted value for z is obtained from the slope of an adjusted straight line for the magnetization autocorrelation time for $L \geq 16$ (see the text). The dotted line is just a guide to the eye.

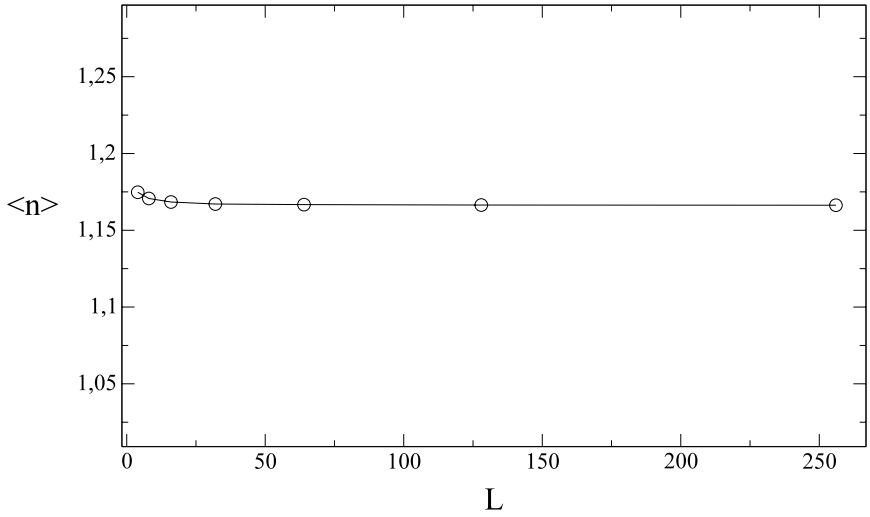

Figure 3. Mean size of the clusters of possibly flipped spins as a function of the linear size L for $E_0 = -0.9$.

energy has an abrupt drop for small times. Therefore, it is not a reliable quantity to extract the autocorrelation time from. The value of z was obtained from the curve for the magnetization and its value is $z = 2.16 \pm 0.04$, which is, within error bars, the same value as for the Metropolis algorithm.

In figure 3 the behavior of the mean size of the clusters of spins, $\langle n \rangle$, is shown, as a function of L. For this value of E_0, it seems that $\langle n \rangle$ does not change with L. We will see shortly that in fact it initially grows with L and eventually saturates at some value of L, which we call $\widetilde{L}$.

doi:10.1088/1742-5468/2010/04/P04012

J. Stat. Mech. (2010) P04012

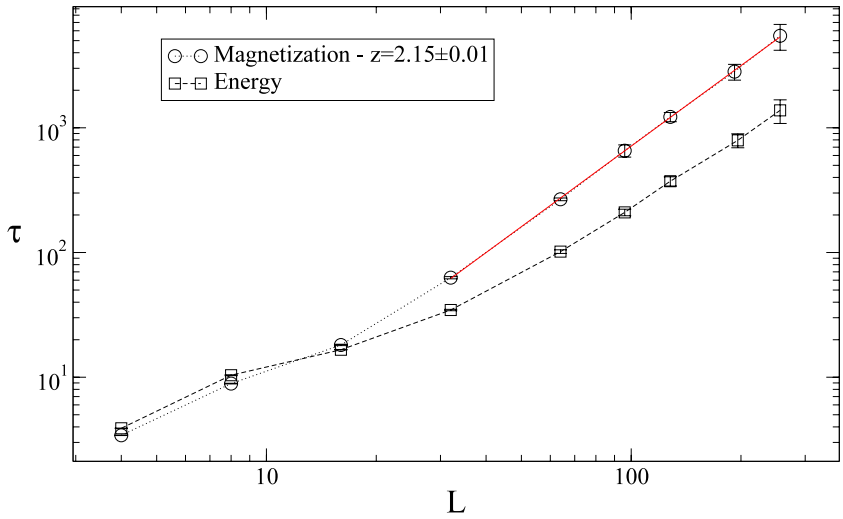

Figure 4. Log–log graphs of magnetization (○) and energy (□) autocorrelation time (in MCS) versus linear size L for the Niedermayer algorithm with $E_0 = 0.0$. The quoted value for z is obtained from the slope of an adjusted straight line for the magnetization autocorrelation time, for values of L beyond the point where the crossover takes place. The dotted line is just a guide to the eye.

The overall picture does not change for $E_0 = -0.5$: the magnetization autocorrelation function is well described by a single-exponential law and the autocorrelation time was calculated from it. The dynamic exponent is $z = 2.12 \pm 0.03$, still consistent with the Metropolis value (the error bars that we quote are all one standard deviation; the intersection with the expected value for the Metropolis algorithm, for this case, is obtained assuming two standard deviations for the error). Since the picture for $E_0 = -0.5$ does not change from the one for $E_0 = -0.9$, we will not depict the graphs for the former.

For $E_0 = 0$, a crossover clearly takes place, as shown in figure 4: for small L, the energy autocorrelation times are larger than their magnetization counterparts, while the situation is reversed for larger L (this behavior is more evident for $E_0 = 0.3$; we showed the corresponding graph in figure 1 above and will comment on it below). The value of z is obtained from the slope of an adjusted straight line for the magnetization autocorrelation function, for values of L beyond the point where the crossover takes place. It reads $z = 2.15 \pm 0.01$ in this case, again compatible with the Metropolis value. The behavior of $\langle n \rangle$ is shown in figure 5: it grows initially with L but eventually saturates at $\widetilde{L} \sim 15$. For small values of L it is the autocorrelation function for the energy which is well described by a single exponential, while the corresponding function for the magnetization shows an abrupt drop for small times. The situation is reversed for $L > \widetilde{L}$.

This picture is maintained for $E_0 > 0.0$, with the value of $\widetilde{L}$ increasing with E_0 and the crossover taking place at larger and larger values of L. The dynamic exponent z is given by 2.16 ± 0.03 and 2.12 ± 0.04 for $E_0 = 0.3$ and 0.5, respectively. Both are compatible with the value for the Metropolis algorithm.

In figure 1 we show the change in the behavior of the autocorrelation functions for the magnetization and the energy. Note that the crossover mentioned above is connected also to the possibility of describing the autocorrelation function using a single exponential:

J. Stat. Mech. (2010) P04012

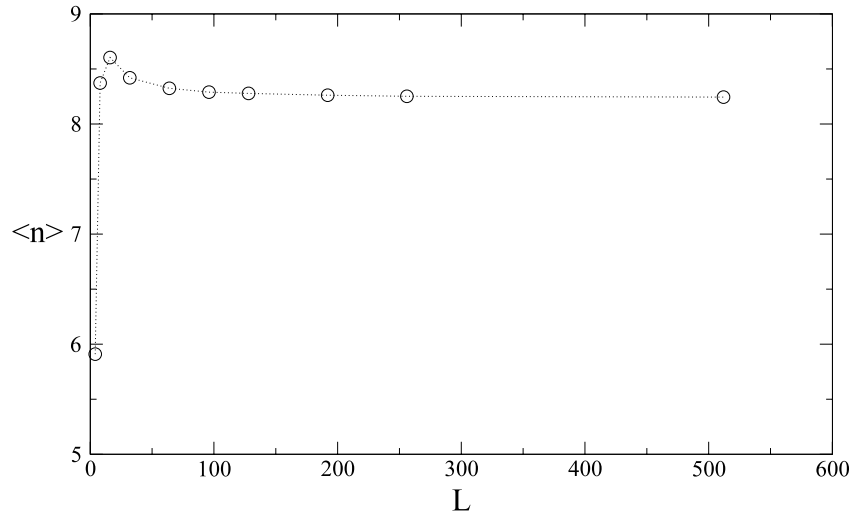

Figure 5. Mean size of the clusters of possibly flipped spins as a function of the linear size L for $E_0 = 0.0$.

this is accomplished with the energy autocorrelation function for small values of L and for its magnetization counterpart for larger values of L.

Finally, for $E_0 = 0.7$ and 0.9 the crossover happens at values of L large enough to prevent a reliable estimate of z. It is necessary to go to values of L well above our present computational capabilities to be able to extract z from the graphs.

Nevertheless, the overall trend is well determined: for $0 \leq E_0 < 1$, the dynamic behavior is the Metropolis one but this behavior sets in only for large enough L. The size of the clusters of turned spins increases with E_0 but eventually saturates for $L = \widetilde{L}$, where $\widetilde{L}$ increases with E_0. For $L > \widetilde{L}$, the relative size of the clusters (i.e., the ratio $\langle n \rangle / L^2$) decreases and, in this sense, the algorithm is like a single-spin one (Metropolis, in our case), explaining the value of its dynamic exponent. Therefore, the Wolff algorithm (corresponding to $E_0 = 1$) is still the best choice, as compared to the Niedermayer algorithm with $E_0 < 1$.

We postpone the discussion of the Wolff algorithm and go to $E_0 > 1$. In this case, spins in different states may be part of the same cluster, although with a smaller probability than spins in the same state, and a cluster will always be flipped (see (iii) above). For $E_0 \gg 1$, almost all spins in the finite lattice will take part in the cluster and the algorithm will not be optimal (in fact, it would not even be ergodic for $E_0 \to \infty$). Therefore, if the Niedermayer algorithm is more efficient than Wolff's, it should be for E_0 close to 1. We, therefore, studied the cases $E_0 = 1.1$ and 1.05. The results are qualitatively equivalent and in figure 6 we show both. Note that the growth of τ with L is faster than a power law for both values of E_0. In the inset, we show the corresponding graph for $E_0 = 1.1$: a crossover is also present but the value where it takes place decreases with E_0 and for $E_0 = 1.1$ it is not seen. Since the value of the autocorrelation time is already greater then for the Wolff algorithm, for a given L, and it grows faster than a power law with L, again the optimal algorithm is Wolff's.

We single out the discussion of the Wolff algorithm ($E_0 = 1$) in order to compare with other values of E_0. Although our intention is not to calculate a precise value of z

J. Stat. Mech. (2010) P04012

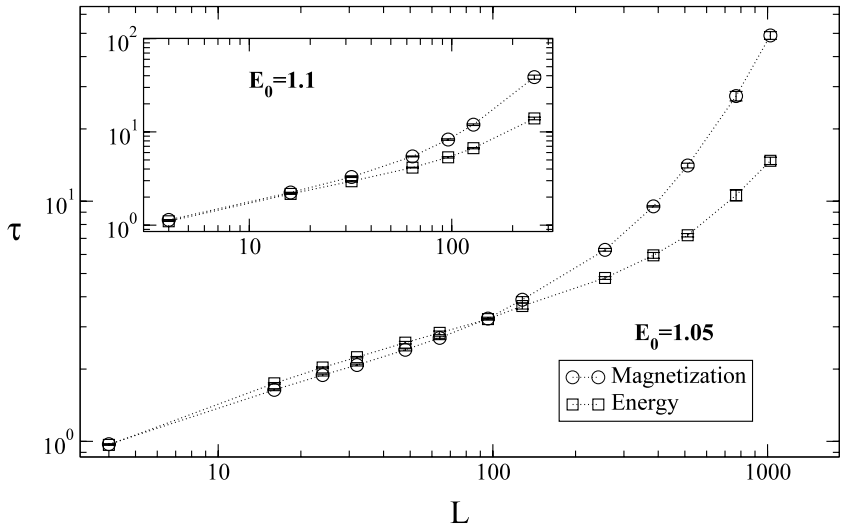

Figure 6. Log–log graphs of magnetization and energy autocorrelation time (in MCS) versus linear size L for the Niedermayer algorithm with $E_0 = 1.05$ and 1.1 (inset). In both graphs the autocorrelation time τ is plotted as function of the linear size L. The dotted line is just a guide to the eye.

for this algorithm, we have adjusted the autocorrelation time for the energy (in this case, it is this function which is well described by a single exponential for small values of the time) as a function of L for three different functions, namely

$$\tau = \begin{cases} AL^z \\ A \ln L + C \\ A(\ln L)^z + C. \end{cases} \tag{10}$$

The first function is the usual scaling law assumed for the autocorrelation function at the critical point. We can see in figure 7 that there is no indication that the best adjusted curve will eventually be a straight line in a log–log plot. Since previous calculations tend to point to a value of z close to zero for the Wolff algorithm in two dimensions, one cannot exclude the possibility of a logarithmic dependence, which is the case of the second function in the above equation. The fitting is better than for the power law but it is not a satisfactory one either. Moreover, it tends to deviate from the data for large enough L. The third function is an ad hoc assumption, which proved to be the best fit to our data, as can be seen in figure 7. The parameters of the function are obtained from a non-linear fitting:

$$\tau = A(\ln L)^z + C, \tag{11}$$

with $A = 0.21 \pm 0.01$, $z = 1.50 \pm 0.02$ and $C = 0.47 \pm 0.03$. We have no theoretical explanation for this behavior. The constant C, however, is a finite-size correction. The behavior in equation (11) is expected to hold true for *large* enough L and the logarithmic dependence makes the scaling region reachable only for very large values of L. For this region, one would expect a simpler law, namely $\tau = A(\ln L)^z$; however, for intermediate or small values of L, the constant C acts as a finite-size correction. A similar scaling law was found for the exponential relaxation time for the Swendsen–Wang algorithm [13].

J. Stat. Mech. (2010) P04012

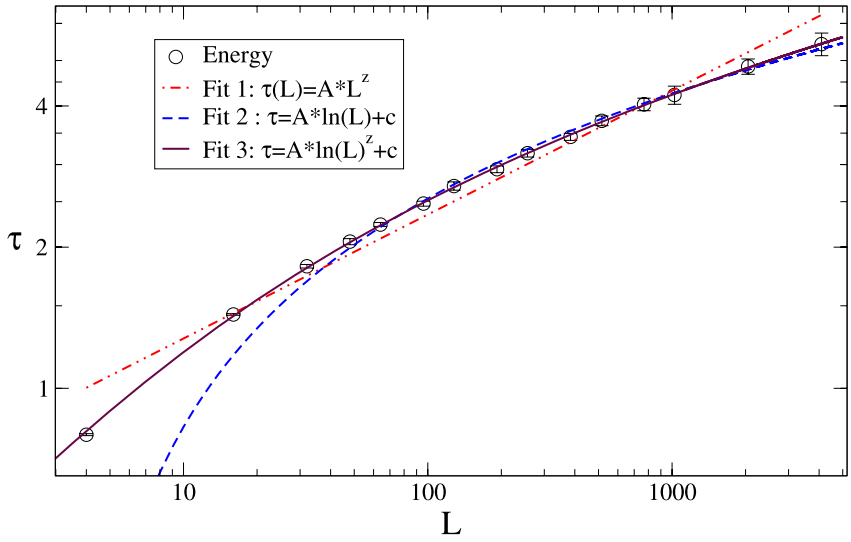

Figure 7. Log–log graph of the energy autocorrelation time (in MCS) as a function of L for the Wolff algorithm. The three fitted curves proposed in (10) are shown.

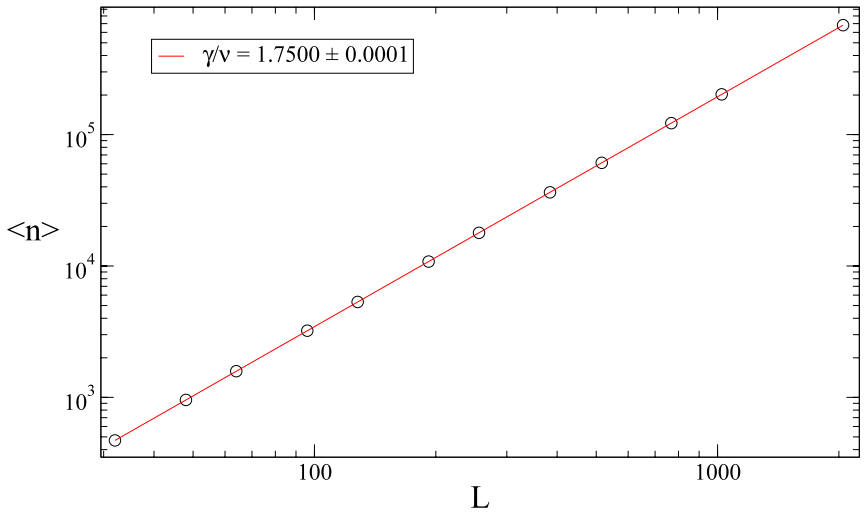

Figure 8. Log–log graph of the mean size of turned clusters as function of L. The slope is an evaluation of γ/ν.

We also depict the mean size of the clusters of turned spins, $\langle n \rangle$, as a function of L in figure 8. The slope of the straight line is 1.7500 ± 0.0001, which is, as expected [2], the value for the ratio γ/ν. Note that, in contrast to what happens for $E_0 < 1$, there is no saturation of $\langle n \rangle$ with L. This seems to explain why Wolff and Niedermayer algorithms are in different dynamic universality classes.

4.2. The XY model

We have applied the Niedermayer algorithm in the study of the dynamic behavior of the XY model as well. The generalization of this algorithm to continuous models is outlined

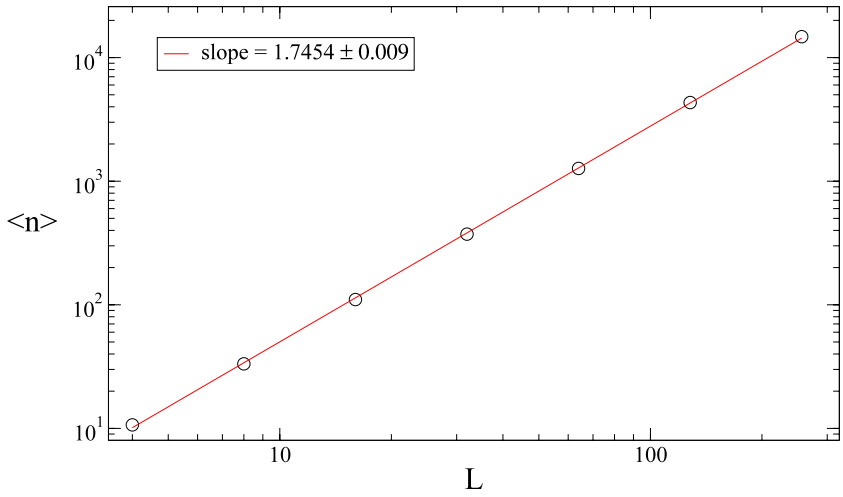

Figure 9. Log–log plot of the mean size of the clusters of flipped spins for the XY model versus the linear size of the lattice. The slope of the curve just misses the expected value for $2-\eta$, 7/4 [15].

in the appendix. Although we have studied three values of E_0, our results are conclusive and lead to an overall picture which is analogous to the one for the Ising model. We have used the value $k_{\mathrm{B}}T_{\mathrm{c}}/J = 0.8865$ for the transition temperature of the two-dimensional XY model. This value is only 0.7% off from the most recent evaluation of $k_{\mathrm{B}}T_{\mathrm{c}}/J$ for this model [14].

The mean size of the clusters of flipped spins for the Wolff algorithm as a function of the linear size of the lattice is depicted in figure 9. The slope of the straight line is 1.7454 ± 0.009, which is slightly different from the expected value for $2-\eta$ for this model at $k_{\mathrm{B}}T_{\mathrm{c}}/J$, 7/4 [15]. In fact, the small discrepancy may be due to the fact that we are not using the (unknown) exact value for the transition temperature.

The autocorrelation times for the magnetization and energy for the Wolff algorithm are shown in figure 10. We have not tried to fit the data but it is evident that the energy autocorrelation time grows with L slower than a power law. The *decrease* in the magnetization autocorrelation time has been observed previously (in fact, an oscillation was observed in an algorithm which mixed Wolff's and Swendsen–Wang's procedures but the overall picture is qualitatively similar to ours; see [16]).

We have simulated also the cases $E_0 = 0$ and -0.5. The mean size of clusters of flipped spins saturates and the value of saturation increases with E_0 (see figure 11). Therefore, one expects the same picture as for the Ising model: in particular, the dynamic behavior for L large enough is the Metropolis one. This is confirmed for $E_0 = 0.0$ explicitly, where the dynamic exponent measured is $z = 1.916 \pm 0.004$ (see figure 12). Recalling our reasoning for the Ising model for $E_0 < 1$, we can infer that the value just quoted for z is an evaluation of the dynamic exponent for the Metropolis algorithm applied to the two-dimensional XY model. Since, to the best of our knowledge, there is no previous evaluation of z for this model and for the Metropolis algorithm, we have made a crude evaluation of z for this case and obtained the value 1.89 ± 0.03, which is in agreement, within error bars, with the value that we obtained for $E_0 = 0.0$ for large L. Clearly, the Wolff algorithm is the most efficient, as compared against the Niedermayer algorithm

J. Stat. Mech. (2010) P04012

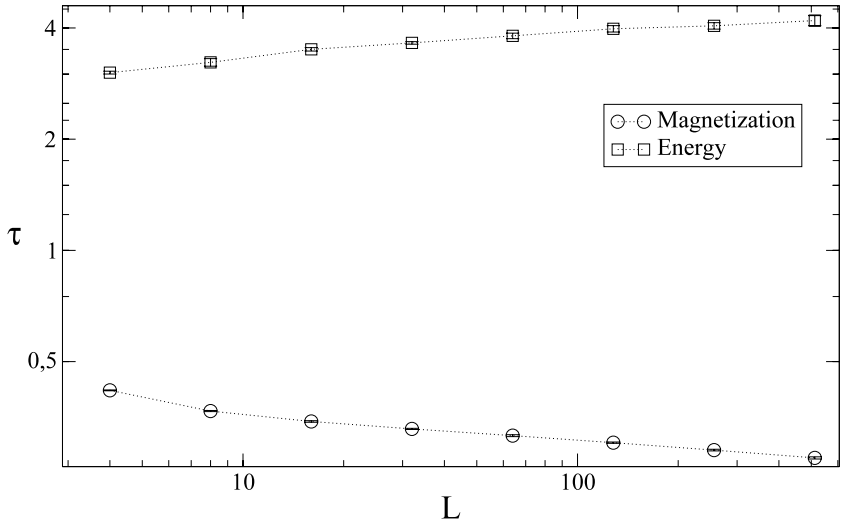

Figure 10. Autocorrelation times for the magnetization (circle) and energy (square) for the Wolff algorithm applied to the two-dimensional XY model.

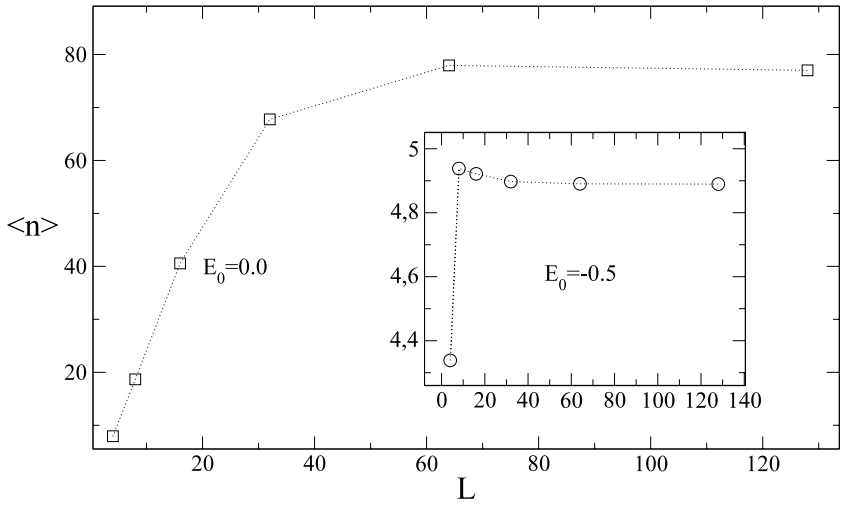

Figure 11. Mean size of the clusters of flipped spins for $E_0 = -0.5$ (inset) and $E_0 = 0$ (main graph) for the two-dimensional XY model.

with the two values of E_0 quoted above. We have also simulated one example of $E_0 > 1$, namely $E_0 = 1.05$. The behavior is qualitatively the same as for the Ising model; see figure 13. Note that the growth of the autocorrelation time is faster than a single power law; in fact, it is well fitted by an exponential. Therefore, for the XY model also the best choice is $E_0 = 1$ (Wolff algorithm), as compared to the Niedermayer algorithm.

5. Summary

We have studied the dynamic behavior of the Niedermayer algorithm applied to the two-dimensional Ising and XY models. Our main goal is to compare its efficiency with the

J. Stat. Mech. (2010) P04012

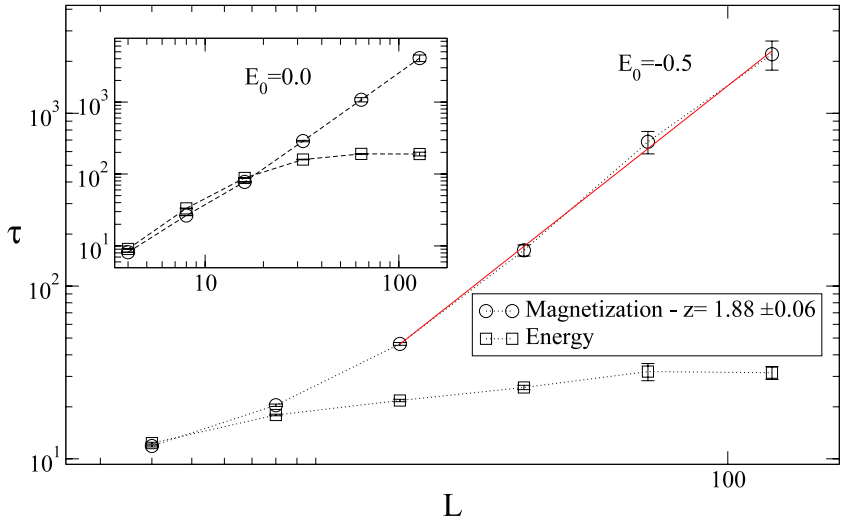

Figure 12. Autocorrelation time for the magnetization and energy as a function of L for $E_0 = -0.5$ (main graph) and $E_0 = 0$ (inset) for the two-dimensional XY model.

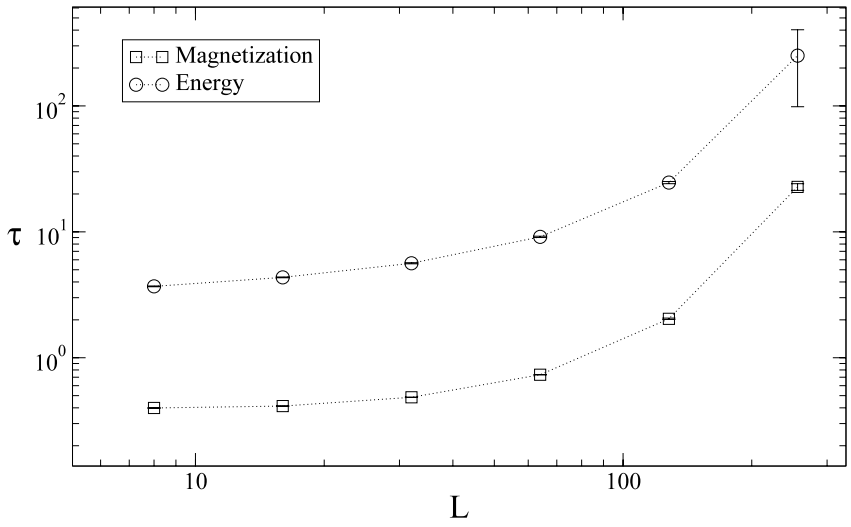

Figure 13. Autocorrelation time for the magnetization and energy as a function of L for $E_0 = 1.05$ for the two-dimensional XY model.

Wolff algorithm. The latter is a particular case of the Niedermayer algorithm, such that a parameter governing the size of the flipped clusters, E_0, assumes the value 1.

We show that, for $-1 < E_0 < 1$, the dynamic behavior eventually recovers the Metropolis ($E_0 = -1$) one. This behavior is linked to the saturation of the mean size of the clusters, which happens for all $E_0 < 1$, leading to a *decrease* of the relative size of these clusters when L increases.

For the Wolff algorithm and the Ising model, we propose a scaling function for the autocorrelation time for the magnetization. This choice is an ad hoc one but fits the data very well and does not coincide with any function proposed so far in the literature. We were not able to make a fitting with the same statistical quality for the XY model.

For $E_0 > 1$, the values of the autocorrelation times are greater than those for the Wolff algorithm and grow faster than a power law with L.

Therefore, at least for these two models, the Wolff algorithm is superior to Niedermayer's.

Acknowledgments

The authors would like to thank the Brazilian agencies FAPESC, CNPq, and CAPES for partial financial support.

Appendix

The Hamiltonian for the XY model can be written as

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} \vec{s_i} \cdot \vec{s_j} \tag{A.1}$$

where J is the coupling constant and $\vec{s_i}$ is the spin of site i, represented by a unit vector in any direction in the xy plane.

To start the cluster we randomly choose a preferred direction $\hat{n}$ and a spin $\vec{s_i}$. This spin is the first one of the cluster. Neighbors of $\vec{s_i}$ are added to a cluster with probability

$$P_{\text{add}} = \begin{cases} 1 - \mathrm{e}^{KE_{ij}(1+E_0)}, & \text{if } E_{ij} < E_0, \\ 0, & \text{otherwise}, \end{cases} \tag{A.2}$$

where in the XY model, $E_{ij} = -(\vec{s_i} \cdot \hat{n})(\vec{s_j} \cdot \hat{n})$. Note that, if $E_0 \leq 1$, only sites that have the same component in the direction $\hat{n}$ as $\vec{s_i}$ may be added to the cluster. The procedure for the construction of the cluster continues as described for the Ising model. After the cluster is built the new directions of the spins are given by a reflection with respect to the axis perpendicular to $\hat{n}$.

The acceptance ratio for the XY model is slightly different from the one for the Ising model. We cannot define the energy difference (ΔE) as the number of spins parallel and anti-parallel to the cluster. So we must calculate the energy before and after the cluster is flipped. In this case we define the acceptance ratio, for $E_0 \leq 1$, as

$$A(a \to b) = \begin{cases} \mathrm{e}^{-(\Delta E/2)K(1-E_0)}, & \text{if } \Delta E > 0 \\ 1, & \text{if } \Delta E < 0, \end{cases} \tag{A.3}$$

where a and b have the same meaning as before and ΔE is the difference in energy between configurations a and b, in units of J. As we can see, for $E_0 = -1$ we regain the Metropolis algorithm with $A = \mathrm{e}^{-K\Delta E}$ and for $E_0 = 1$ we regain the Wolff algorithm with $A = 1$ for all clusters. These choices ensure that detailed balance is obeyed.

The generalization for $E_0 > 1$ is analogous to the one described above and, again, spins with different signs for the component along $\hat{n}$ may also be part of the cluster, and $A = 1$ always.

References

[1] Landau D P and Binder K, 2005 *A Guide to Monte Carlo Simulations in Statistical Physics* 2nd edn (New York: Cambridge University Press)
[2] Newman M E J and Barkema G T, 2001 *Monte Carlo Methods in Statistical Physics* (Oxford: Oxford University Press)
[3] Wang F G and Landau D P, 2001 *Phys. Rev. Lett.* **86** 2050
[4] de Oliveira P M C, 1991 *Computing Boolean Statistical Models* (London: World Scientific)
[5] Swendsen R H and Wang J-S, 1987 *Phys. Rev. Lett.* **58** 86
[6] Wolff U, 1989 *Phys. Rev. Lett.* **62** 361
[7] Nightingale M P and Blöte H W, 1996 *Phys. Rev. Lett.* **76** 4548
[8] Baillie C F and Coddington P D, 1991 *Phys. Rev.* B **43** 10617
[9] Coddington P D and Baillie C F, 1992 *Phys. Rev. Lett.* **68** 962
[10] Niedermayer F, 1988 *Phys. Rev. Lett.* **61** 2026
[11] Wansleben S and Landau D P, 1991 *Phys. Rev.* B **43** 6006
[12] Salas J and Sokal A D, 1997 *J. Stat. Phys.* **87** 1
[13] Du J, Zheng B and Wang J-S, 2006 *J. Stat. Mech.* P05004
[14] Arisue H, 2007 *Prog. Theor. Phys.* **118** 855
[15] Butera P and Percini M, 2008 *Physica* A **387** 6293
[16] Edwards R G and Sokal A D, 1989 *Phys. Rev.* D **40** 1374

J. Stat. Mech. (2010) P04012