

Single-Cluster Monte Carlo Dynamics for the Ising Model

P. Tamayo,¹ R. C. Brower,² and W. Klein³

Received July 27, 1989; revision received September 7, 1989

We present an extensive study of a new Monte Carlo acceleration algorithm introduced by Wolff for the Ising model. It differs from the Swendsen–Wang algorithm by growing and flipping single clusters at a random seed. In general, it is more efficient than Swendsen–Wang dynamics for $d > 2$, giving zero critical slowing down in the upper critical dimension. Monte Carlo simulations give dynamical critical exponents $z_w = 0.33 \pm 0.05$ and 0.44 ± 0.10 in $d = 2$ and 3, respectively, and numbers consistent with $z_w = 0$ in $d = 4$ and mean-field theory. We present scaling arguments which indicate that the Wolff mechanism for decorrelation differs substantially from Swendsen–Wang despite the apparent similarities of the two methods.

KEY WORDS: Cluster acceleration; critical slowing down; Swendsen–Wang algorithm; Wolff algorithm; Fortuin–Kasteleyn mapping; Coniglio–Klein clusters.

1. INTRODUCTION

Considerable attention has been given in recent years to efficient Monte Carlo methods in statistical physics. Recently, Swendsen and Wang⁽¹⁾ (SW) introduced a new dynamics which effectively reduces the dynamical exponent z and the effects of critical slowing down. This dynamics, based on the Fortuin–Kasteleyn map for site-bond percolation and the Potts model,^(2,3) flips whole clusters. This increases the relaxation rate while preserving

¹ Physics Department and Center for Polymer Studies, Boston University, Boston, Massachusetts.

² Department of Electrical, Computer and Systems Engineering and Physics Department, Boston University, Boston, Massachusetts.

³ Physics Department and Center for Polymer Studies, Boston University, Boston, Massachusetts.

detailed balance and ergodicity. Although this dynamics is useful and generalizations already exist,⁽⁴⁻⁷⁾ there are still open questions about the dynamic universality classes⁽⁸⁾ it introduces.⁽⁹⁾

Here we study a single-cluster Monte Carlo algorithm introduced by Wolff⁽¹⁰⁾ for the $O(N)$ spin models as a variation on the SW scheme for the Ising model. The main difference for the Ising case is that in each Monte Carlo step only one cluster is formed and flipped with probability 1. In the original SW algorithm all percolation clusters are formed and flipped with probability 1/2. This algorithm is an application to dynamics of the Leath cluster growth method.⁽¹¹⁾ The single percolation cluster is grown starting from a site selected at random. The clusters formed are the same Coniglio–Klein droplets⁽³⁾ used by SW dynamics, but they are sampled with an additional weight factor equal to the cluster volume. The effect of this procedure is to choose bigger clusters, which generally reduce auto-correlation times and increase the efficiency.

In this paper, we present numerical results for $d=2, 3, 4$ and mean-field simulations and scaling arguments that give insight into the decorrelation mechanisms involved. Our main results are summarized in Table I.

2. SINGLE-CLUSTER MONTE CARLO DYNAMICS

In a recent paper, Wolff considered two distinct ideas for generalizing Swendsen–Wang cluster dynamics. One idea involved a new embedding scheme for the continuous spin $O(N)$ models similar to one introduced for the ϕ^4 theory⁽⁶⁾ and the other idea modified the method of updating the percolation clusters. It is this modified update procedure which we wish to study in the simplest case of the Ising model. We begin with a brief review of the basic formalism for cluster dynamics of the Ising model.

We will study relaxation rates by finite-size effects on hypercubic lattices of linear size L and volume L^d in dimensions $d=2, 3$, and 4. The standard Ising probability distribution is

$$P_{\text{I}}(\sigma_i) = \frac{1}{Z_{\text{I}}} \exp \left[K \sum_{\langle i,j \rangle} (\sigma_i \sigma_j - 1) \right] \quad (1)$$

where $K=J/k_{\text{B}}T$ and J is the coupling constant. $P_{\text{I}}(\sigma_i)$ can also be expressed as a product over bonds,

$$P_{\text{I}}(\sigma_i) = \frac{1}{Z_{\text{I}}} \prod_{\langle i,j \rangle} [p \delta_{\sigma_i \sigma_j} + (1-p)] \quad (2)$$

where $p = 1 - e^{-2K}$. The key step to constructing the cluster dynamics is to

consider a joint probability distribution for the Ising spins σ_i and site bond percolation variables n_{ij} ,

$$P_{\text{Joint}}(\sigma_i, n_{ij}) = \frac{1}{Z_{\text{Joint}}} \prod_{\langle i, j \rangle} [p \delta_{\sigma_i \sigma_j, 1} \delta_{n_{ij}, 1} + (1-p) \delta_{n_{ij}, 0}] \quad (3)$$

which has the property that summing over the percolation variables (n_{ij}) gives the Ising model and summing over the Ising variables (σ_i) gives the so-called random percolation model,

$$P_{\text{RC}}(n_{ij}) = \frac{1}{Z_{\text{RC}}} q^{N_{\text{clust}}} \prod_{n_{ij}=1} p \prod_{n_{ij}=0} (1-p) \quad (4)$$

The factor $q^{N_{\text{clust}}}$ is written correctly for a q -state Potts model, where N_{clust} is the number of clusters, although in the Ising model $q = 2$.

The Swendsen–Wang cluster dynamics operates on the joint model to update the cluster distribution (or percolation variables) for fixed spins followed by an update of the spins for fixed bonds. In each percolation step the bonds are set ($n_{ij} = 1$) with probability

$$p_{ij} = 1 - e^{-K(1 + \sigma_i \sigma_j)} \quad (5)$$

which results in the correct distribution of clusters. We define n_c , the average number of clusters containing c up (or c down) spins on a lattice of size N . These distributions are analogous to the standard n_s distribution of pure $q = 1$ percolation, except that we do not drop the incipient infinite cluster. Hence the normalized cluster probability distribution is

$$P(c) = \frac{n_c}{N_{\text{clust}}} \quad (6)$$

since $\sum_c n_c = N_{\text{clust}}$. The SW dynamics updates all clusters drawn from this distribution, whereas Wolff's dynamics consists of updating a single cluster drawn from the distribution

$$P_{\text{Wolff}}(c) = \frac{|c| n_c}{N} \quad (7)$$

weighted by the volume of the cluster relative to the SW algorithm.

In detail, the Wolff algorithm consists of the following steps:

1. Pick a site i_0 at random.
2. Grow a percolation cluster from i_0 by throwing bonds to nearest neighbors with probability $p_{i_0, j} = 1 - \exp[-\beta J(1 + \sigma_{i_0} \sigma_j)]$, and continue iteratively until no new peripheral sites are generated.

3. Flip with probability one the spins in the cluster just formed, take measurements, and repeat the cycle.

The resulting dynamics is ergodic and obeys detailed balance.⁽¹⁰⁾ It differs in two respects from SW dynamics: (1) It flips a single a cluster at a time before repercolating the lattice and (2) it draws this cluster from the probability distribution $P_{\text{Wolff}}(c)$.

In order to compare Wolff's results with other, more conventional dynamics (Glauber, heat bath) or with SW, Wolff's Monte Carlo iteration time must be rescaled. One Wolff time step has a computational cost of the cluster size $|c|$, while the conventional update is proportional to the volume of the entire lattice N . Therefore, Wolff's autocorrelation time must be multiplied by a factor $\langle |c| \rangle / N$, where

$$\langle |c| \rangle = \sum_c |c| P_{\text{Wolff}}(c) = \sum_c \frac{c^2 n_c}{N} \quad (8)$$

By the Fortuin–Kasteleyn mapping this quantity $\langle |c| \rangle$ for the random cluster model is proportional to the Ising susceptibility.

Thus, the relaxation times τ'_w measured in a Wolff dynamics simulation are rescaled according to

$$\tau_w = \frac{\langle |c| \rangle}{N} \tau'_w \quad (9)$$

which with the finite-size scaling relations $\langle |c| \rangle \sim L^{\gamma/\nu}$, $N = L^d$, and $\tau \sim L^{z_w}$ gives

$$z_w = z'_w - (d - \gamma/\nu) \quad (10)$$

This z_w is the correct parameter to compare with the dynamical exponents of Swendsen–Wang, z_{SW} .

3. ANALYSIS AND NUMERICAL RESULTS

In our computer simulations, we accumulate measurements of the internal energy, magnetization, and susceptibility at every time step. To estimate τ'_w , we compute the time autocorrelation function over samples of at least 3×10^5 steps. The autocorrelation function being considered

$$C(t) = \frac{\langle A(0) A(t) \rangle - \langle A \rangle^2}{\langle A \rangle^2 - \langle A \rangle^2} \quad (11)$$

where the average is over the sampling time and A is the energy or susceptibility. τ'_w is determined from a linear fit of $\log C(t)$ to t in the region in which the long-time single exponential decay is seen.

Table I^a

	$d=2$	$d=3$	$d=4$	Mean-field
γ	1.75	1.25	1	1
ν	1	0.629	$\frac{1}{2}$	$\frac{1}{2}$
$d-\gamma/\nu$	0.25	1.01	2	2
z'_w	0.58 ± 0.05	1.45 ± 0.10	1.95 ± 0.15	1.90 ± 0.15
z_w	0.33 ± 0.05	0.44 ± 0.10	-0.05 ± 0.15	-0.10 ± 0.15
z_{sw}	0.35	0.75	—	1

^a The first three rows give the static exponents γ , ν , and the rescaling shift. The next two rows give our measured Wolff exponents and its shifted values: $z_w = z'_w - (d - \gamma/\nu)$. The last row gives the Swendsen–Wang exponents.⁽¹⁾

Statistical fluctuations in $C(t)$ lead to errors in the calculation of τ'_w .^(12–15) To estimate those errors, we split the data sample into several groups and recomputed τ'_w for each group. Each group is much longer than τ'_w . The error is computed using standard error analysis techniques. Values of τ'_w are computed for different lattice sizes at the critical point for the infinite lattice and an estimate for z'_w is obtained by finite-size scaling,

$$\tau'_w \sim \xi^{z'_w} \sim L^{z_w} \tag{12}$$

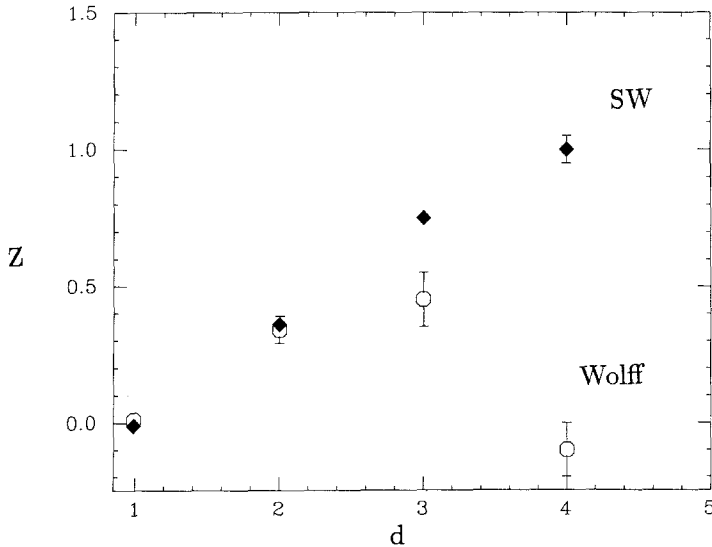


Fig. 1. Dynamical critical exponents z versus dimensionality for Swendsen–Wang and Wolff algorithms.

and z'_w is computed from a linear fit of $\log \tau'_w$ to $\log L$. Finally, the effective z_w is calculated using Eq. (10).

Other methods to estimate z_w were also used. Relaxation times computed from the area under the autocorrelation function give values of z_w which seem to agree well with our previous results. Double exponential fits⁽¹⁵⁾ also give similar results. Our results are summarized in Table I and Figs. 1 and 2.

In Fig. 3 we plot cluster distributions for Wolff and SW. As expected, the cluster distribution for Wolff is equal to the one for SW weighted by the size of the cluster:

$$P_{\text{Wolff}}(c) = \frac{|c|}{\langle |c| \rangle_{\text{SW}}} P(c) \tag{13}$$

and the two distributions must cross each other at $|c| = \langle |c| \rangle_{\text{SW}}$.

The distributions follow the finite cluster form $P(c) \sim |c|^{-\tau}$ for the random cluster model except at large values for $|c|$ when infinite (spanning) clusters contribute, producing a “bump” in the tail of the distribution.

Finally, we wish to remark on some numerical simulations designed to distinguish between the two novel aspects of Wolff’s update scheme: (1)

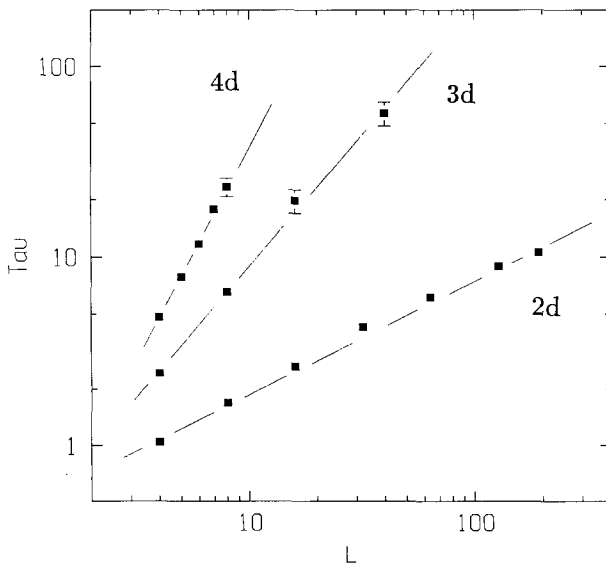


Fig. 2. Energy autocorrelation relaxation time (τ_{EE}) for the Ising Wolff dynamics in $d=2, 3, 4$.

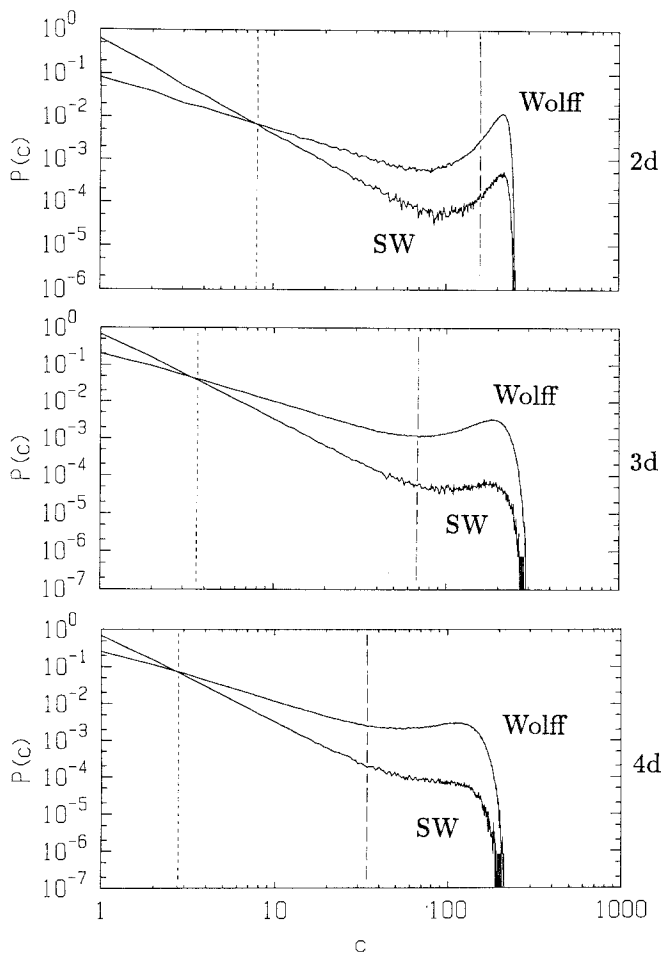


Fig. 3. Cluster probability distribution for Swendsen–Wang and Wolff dynamics for different dimensions. The cluster size is denoted by c . The system sizes are 16×16 , 10^3 , and 6^4 . Each $P(c)$ was computed with at least 10^5 clusters. The dotted line indicates the average cluster size for SW and the dashed line the average cluster size for Wolff.

flipping single clusters and (2) using a probability function $P_{\text{Wolff}}(c)$ weighted by the cluster mass. We can try update schemes with different cluster probability weights. For example, we can percolate the entire lattice, pick a single cluster at random with the probability $P(c)$ appropriate for Swendsen–Wang dynamics, and flip it with 100% probability, before repercolating the lattice. Such a scheme we refer to as a single-cluster Swendsen–Wang (scSW) algorithm. Again the rescaled autocorrelation is

defined by rescaling the measured $\tau'_{\text{scSW}} \sim L^{z'_{\text{scSW}}}$ by a factor $\langle |c| \rangle_{\text{SW}}/N$, where

$$\langle |c| \rangle_{\text{SW}} = \sum_c \frac{|c| n_c}{N_{\text{clust}}} = \frac{N}{N_{\text{clust}}} \quad (14)$$

Here $\langle |c| \rangle_{\text{SW}}$ scales as L^0 , so that the rescaled τ_{scSW} gives $z_{\text{scSW}} = z'_{\text{scSW}} - d$.

It should be emphasized that, unlike Wolff's algorithm, there may be no way of actually finding these single clusters in $O(N_{\text{clust}})$ operations. Nonetheless, choosing single clusters from various probability distributions can help in analyzing the role of cluster sizes. Our present simulations give $z_{\text{scSW}} = 0.17 \pm 0.10$ and 0.45 ± 0.15 for $d=2$ and 3, respectively. Since the intervening percolation steps should act to further decorrelate clusters, the fact that our results are consistent with $z_{\text{scSW}} \leq z_{\text{SW}}$ might have been anticipated. Further interpretations should await more accurate data across a broader range of lattices sizes and probability distributions.

4. CONCLUSIONS

The data presented in the previous section demonstrate clearly that the Wolff algorithm has smaller decorrelation times in three and four dimensions than SW, and correspondingly smaller exponents (i.e., $z_{\text{W}} < z_{\text{SW}}$).

In two dimensions, the exponents are equal ($z_{\text{W}} \simeq z_{\text{SW}}$) within error bars. Moreover, there may be finite-size effects visible in the curvature of the $d=2$ data of Fig. 2, which would imply a smaller z_{W} asymptotically.

To understand why Wolff dynamics can be faster than SW, we have measured the cluster size distribution for the SW and Wolff algorithms. It is obvious from Fig. 3 that one reason that the Wolff algorithm is more efficient than SW is that the mean size of the clusters flipped is significantly larger. Specifically, the method of choosing the Wolff clusters, i.e., choosing a cluster with a probability proportional to the number of spins in the cluster, is equivalent to multiplying the SW cluster distribution by the mean cluster size. For large clusters the size of the correlation length ξ thus results in a maximum of $\xi^{\beta/\nu}$ incipient infinite clusters per Monte Carlo time step rather than the one incipient infinite cluster per time step in SW. This restriction to one incipient infinite cluster per time step in SW is a consequence of the percolation definition of the flipped clusters.^(16,1,3)

To understand the Wolff algorithm and how it differs from SW, we must consider the dynamics of the incipient infinite clusters and how the Wolff algorithm employs them to eliminate the correlations. In the SW algorithm we have argued⁽⁹⁾ that the mechanism of domain wall diffusion,

which dominates the decorrelation mechanism in the Glauber algorithm, is retained. If the Wolff algorithm is using the incipient infinite cluster of size ξ to eliminate correlated domains, then it cannot eliminate them by domain wall diffusion. An alternate mechanism is the uniform thinning of large correlated domains in a (roughly) uniform manner.

If the incipient infinite clusters are independent, or correlated over a finite time, then one could describe the “thinning” decorrelation process in the Wolff algorithm as a directed walk in magnetization space with a step size equal to the mass of the incipient infinite cluster $\xi^{d-\beta\nu}$. Here ξ is the correlation length, β is the order parameter exponent,^(16,3) ν is the correlation length exponent,^(16,3) and d is the dimension of space. To eliminate a domain of mass ξ^d requires $\xi^d/\xi^{d-\beta\nu} = \xi^{\beta/\nu}$ steps and hence a time which scales as $\xi^{2\beta/\nu}$.

For $d \geq 4$ the incipient infinite clusters are independent. This can be seen from the following argument. In infinite dimensions incipient infinite clusters with fractal dimension 4 will not interfere with each other. Therefore, one expects no correlation between large clusters, since they do not “see each other.” This leads to a $\tau'_w \sim \xi^{2\beta/\nu}$ and a renormalized $z_w = 0$. Since z_w for $d \geq 4$ should be the same as the infinite-dimension value, we expect $z_w = 0$ in $d = 4$. This is consistent with the data (see Table I).

To support this argument, we have measured the time-dependent susceptibility–susceptibility correlation function $\langle \chi\chi \rangle$ (Fig. 5). Since the mean cluster size is equivalent to the susceptibility,^(16,3) $\langle \chi\chi \rangle$ will be proportional to the cluster–cluster correlations for clusters the size of ξ . In $d = 4$, $\langle \chi\chi \rangle$ decays exponentially, leading to the conclusion that there are only finite time correlations and that the clusters can be treated as independent.

In addition, we have plotted in Fig. 4 the size of the “flipped cluster” for each unrenormalized time step. The y axis is the size of the cluster, with the sign denoting the direction of the flip, and the x axis is the time where one time unit corresponds to one cluster flip. The brackets about some of the points in the figure represent the number of spins that a particular cluster had in common with the previous cluster flipped in the opposite direction. This plot, consistent with Fig. 5, shows almost no correlation between large clusters in $d = 4$.

We have made the same cluster size–time plots for $d = 2$ and 3. The difference is striking. Clearly there is considerably more correlation between large clusters in 2 and 3 dimensions than there is in 4. This is supported by the measurement of $\langle \chi\chi \rangle$ in $d = 2$ and 3. As can be seen from Fig. 5, the relaxation time scales with the system size. Fitting the data to exponential decay, we extract a relaxation time that diverges as $\xi^{z_{\chi\chi}}$, where $z_{\chi\chi} = z_w$ within the accuracy of the data.

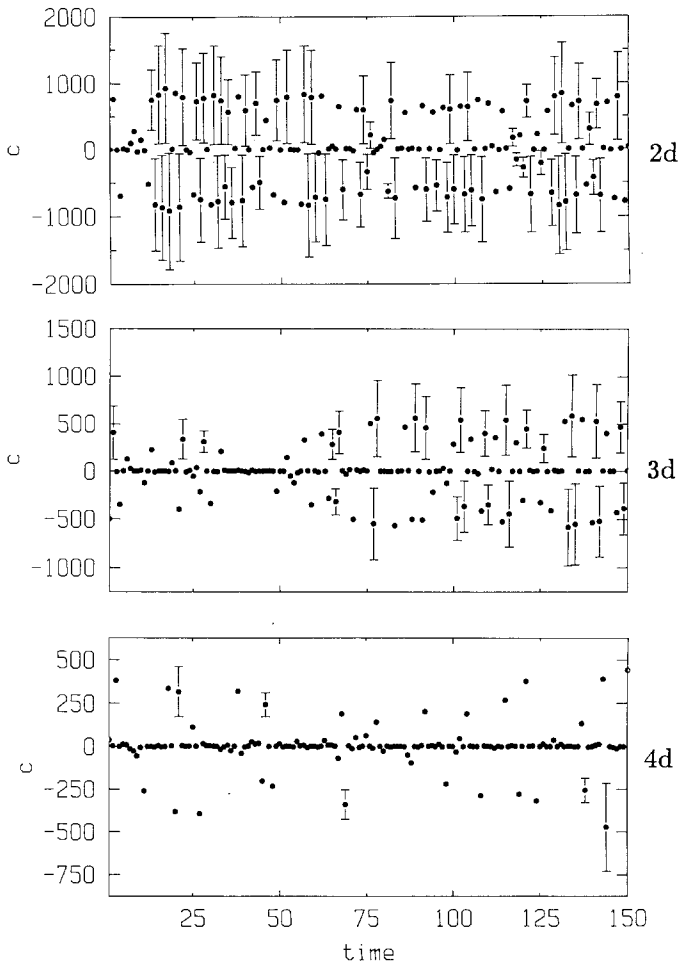


Fig. 4. Size of the flipped cluster versus time step, for Wolff dynamics in $d=2, 3, 4$. The brackets indicate the number of spins that the flipped cluster had in common with the previous cluster flipped.

This leads us to postulate the following mechanism to explain the acceleration of decorrelation in the Wolff algorithm. The dominant mechanism for decorrelation is in the clusters the size of the correlation length thinning the correlating domains in a spatially uniform manner. This is in contrast to the SW algorithm, which appears to maintain the domain wall diffusion mechanism. We can then think of the decorrelation as a directed walk in magnetization space where each step requires a time $\xi^{2/d}$.

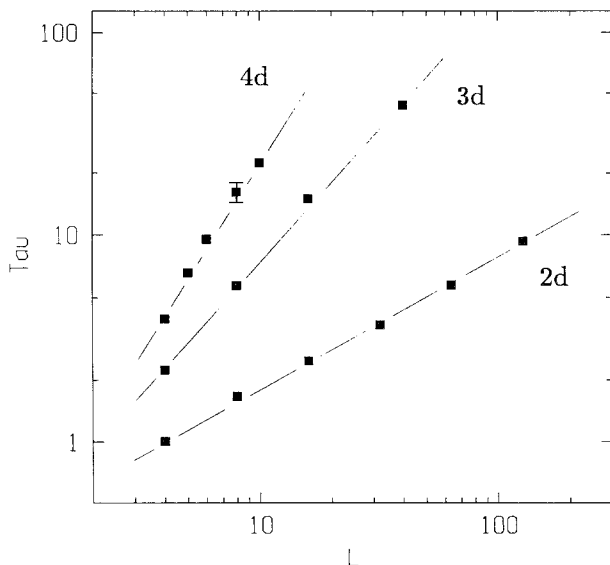


Fig. 5. Susceptibility autocorrelation relaxation time (τ_{xx}) for the Ising Wolff dynamics in $d=2, 3, 4$. The z exponents computed from these data agree well with the ones computed from Fig. 2.

With the proper renormalization of the time by the factor $d - \gamma/\nu$ this leads to the following simple formula for the Wolff exponent:

$$z_w = \frac{2\beta}{\nu} + z_{xx} - d + \frac{\gamma}{\nu} \quad (15)$$

or with hyperscaling $z_w = z_{xx}$, consistent with the measurements.

The above considerations support the conclusion that the Wolff algorithm is more efficient than the SW algorithm for obtaining static critical exponents in dimensions greater than or equal to three. Within the accuracy of our data, Wolff and SW are equivalent in $d=2$ and $z_w = z_{sw}$ trivially in $d=1$.

Finally in this section we note that the Wolff algorithm appears to have changed a basic ingredient in the way decorrelation takes place in Glauber. Specifically, rather than domain wall diffusion, which is maintained as the dominant mechanism in SW, Wolff appears to use a “thinning” process as discussed above. In this regard SW appears to be more faithful to the original local Glauber model dynamics and may be more useful in obtaining information about dynamical evolution in processes far from equilibrium such as nucleation and domain growth.

ACKNOWLEDGMENTS

We would like to acknowledge many useful conversations with A. Sokal, A. Ferrenberg, T. Ray, R. Swendsen, R. Edwards, H. Gould, A. Coniglio, N. Jan, R. Giles, and D. Stauffer. We also thank J. Leao from the Physics Computing Facility and Chip Cole from the Academic Computing Center at Boston University. This project was supported in part by a grant from the Office of Naval Research.

REFERENCES

1. R. Swendsen and J. S. Wang, *Phys. Rev. Lett.* **58**:86 (1987).
2. C. M. Fortuin and P. W. Kasteleyn, *Physica* **57**:536 (1972).
3. A. Coniglio and W. Klein, *J. Phys. A* **13**:2775 (1980).
4. F. Niedermayer, *Phys. Rev. Lett.* **61**:2026 (1988).
5. R. G. Edwards and A. D. Sokal, *Phys. Rev. D* **38**:2009 (1988); A. Sokal, Monte Carlo methods in statistical mechanics: Foundations and new algorithms, Lecture Notes.
6. R. Brower and P. Tamayo, *Phys. Rev. Lett.* **62**:1087 (1989).
7. T. Ray, P. Tamayo, and W. Klein, *Phys. Rev. A* **39**:5949 (1989).
8. P. C. Hohenberg and B. Halperin, *Rev. Mod. Phys.* **49**:435 (1977).
9. W. Klein, T. Ray, and P. Tamayo, *Phys. Rev. Lett.* **62**:163 (1989).
10. U. Wolff, *Phys. Rev. Lett.* **62**:361 (1989).
11. P. L. Leath, *Phys. Rev. B* **14**:5046 (1976).
12. K. Binder, ed., *Monte Carlo Methods in Statistical Physics* (Springer-Verlag, 1979).
13. H. Gould and J. Tobochnik, *Computer Simulation Methods* (Addison-Wesley, 1988).
14. E. Stoll, K. Binder, and T. Schneider, *Phys. Rev. B* **8**:3266 (1973).
15. S. Tang and D. P. Landau, *Phys. Rev. B* **36**:567 (1987).
16. T. Ray and W. Klein, *J. Stat. Phys.* **53**:773 (1988).