Journal of Statistical Physics, Vol. 58, Nos. 1/2, 1990

Critical Acceleration of Lattice Gauge Simulations

R. Ben-Av,[1] D. Kandel,[2] E. Katznelson,[1] P. G. Lauwers,[3] and S. Solomon[1]

Received June 28, 1989; revision received August 29, 1989

We present a stochastic cluster algorithm that drastically reduces critical slowing down for Z_2 lattice gauge theory in three dimensions. The dynamical exponent z is reduced from $z > 2$ (standard Metropolis algorithm) to $z \approx 0.73$. The Monte Carlo pseudodynamics acts on the gauge-invariant flux tubes that are known to be the relevant large-scale low-energy excitations. A comparison of our results with known results for the 3D Ising model and ϕ^4 model supports the conjecture of universality classes for stochastic cluster algorithms.

KEY WORDS: Critical slowing down; dynamical exponents; stochastic cluster algorithms; Z_2 lattice gauge theory.

1. BACKGROUND

Monte Carlo (MC) simulations are an important tool in statistical mechanics and field theory. They provide the possibility to verify analytical calculations, give hints on how to solve models, and provide the only reliable quantitative estimates for many models.

A major drawback of the Metropolis algorithm[1] and its variants (the most widely used MC simulation methods) is that they become highly inefficient near critical points. When a second-order phase transition is approached the correlation length diverges. This signals the appearance of large-scale low-energy excitations in the system which dominate the dynamics of the simulation. The Metropolis algorithm being local in nature does not handle these large-scale excitations efficiently; the autocorrelation time τ of the system (the number of Metropolis sweeps required to generate

[1] Department of Physics, Weizmann Institute of Science, Rehovot 76100, Israel.
[2] Department of Electronics, Weizmann Institute of Science, Rehovot 76100, Israel.
[3] Höchstleistungsrechenzentrum (HLRZ), c/o KFA Jülich GmbH, Postfach 1913, D-5170 Jülich, Federal Republic of Germany.

0022-4715/90/0100-0125$06.00/0

statistically independent configurations) diverges with the correlation length ξ. This phenomenon is known as critical slowing down (CSD).

The manner in which the autocorrelation time τ diverges with the correlation length ξ is expressed in terms of the dynamic critical exponent z. As one approaches the critical point, τ diverges as

$$\tau \sim \xi^z \tag{1}$$

In order to determine z one can perform simulations near the critical point to measure both τ and ξ, and then use Eq. (1) to calculate z. When using this method one should choose the linear size of the lattice L such that the relation $\xi \ll L$ is satisfied. z can also be determined by means of simulations *at* criticality. In this case τ diverges as

$$\tau \sim L^z \tag{2}$$

with the linear size of the system.

For many models (e.g., the ϕ^4 theory) the dynamic critical exponent of local algorithms, like the Metropolis algorithm, is around $z \approx 2$ (see, e.g., ref. 2). Therefore a method that has a considerably smaller exponent can save orders of magnitude in computer time. Using such a method, one will be able to perform high-precision simulations that cannot be done with standard MC techniques.

It is obvious that a global update algorithm[3–6] is needed in order to achieve a considerable reduction of the dynamic exponent z. However, trying to suggest global changes on geometrical sublattices usually leads to very low acceptance rates on one hand, and suggesting blocks based on the current configuration breaks detailed balance on the other. A new class of updating methods for Monte Carlo simulations has been developed over the past few years as an elegant and efficient way out of this dilemma.[7–19] By means of *stochastic blocking*, these algorithms manage to select clusters of variables that are updated in a homogeneous way without violating detailed balance. By an appropriate choice of the blocking criterion, the stochastic cluster methods succeed in drastically reducing CSD for several classes of models: the Ising model in two, three, and four dimensions,[11–14,19] the XY model in two dimensions, the $O(3)$ model,[17] the ϕ^4 model,[15] etc. Until now it was not clear how these powerful ideas could be applied to the important class of models with gauge symmetries. In this work we present the first successful implementation for such a model: Z_2 lattice gauge theory in three dimensions. It has been shown that, using the stochastic blocking in a multigrid framework, one can, in the case of Ising models in two and three dimensions, eliminate CSD completely ($z = 0 \pm 0.1$ for the two-dimensional Ising model).[12–14,19] An adaptation of

the stochastic cluster algorithm for Z_2 lattice gauge theory to the multigrid framework could possibly lead to a similar result for this model.

The Z_2 lattice gauge theory can be defined in the following way. Let Λ be a three-dimensional cubic lattice of size L^3. The dynamic variables of the theory can only take the values $+1$ or -1 and are located on the links of the lattice. The energy of a configuration is defined as the sum of the energies of the individual plaquettes p:

$$\mathcal{H} = \sum_p E(p) \tag{3}$$

The energy of a plaquette p is

$$-E(p) = S_1 S_2 S_3 S_4 \tag{4}$$

where S_i, $i = 1, 2, 3, 4$, are the variables which are located on the links of plaquette p. It can easily be seen that all configurations consist of sets of closed loops of frustrated plaquettes embedded in a medium of satisfied plaquettes (see Section 2.1). In fact this is a consequence of the discretized version of Gauss' law, which implies conservation of electric flux in the absence of sources. The specification of *flux tubes* (the closed loops of frustrated plaquettes) completely determines the configuration modulo gauge transformations. This property is heavily used in our work; our MC procedure consists in updating the large geometry of the flux tubes, rather than the microscopic elementary variables defined on the links.

It is a well-known fact that Z_2 lattice gauge theory is dual to the Ising model in three dimensions; their partition functions are equal. Both models have been investigated extensively by means of MC simulations, and the most accurate data were obtained for the Ising model. The critical temperature of the Ising model, and by duality that of the gauge theory, were determined to four significant digits, and good estimates for the critical exponents were obtained as well. We want to stress that this duality does not imply a configuration-to-configuration correspondence between the models. Therefore, the equivalence between the models cannot be used to translate the successful stochastic cluster algorithm for the Ising model into a stochastic cluster algorithm for Z_2 lattice gauge theory.

2. THE METHOD

2.1. Introductory Remarks

Before we embark on a detailed description of the method and a proof of its validity, it is useful to emphasize some general aspects. The method

presented here is explicitly gauge invariant, i.e., the variables that enter the arguments and are updated are the plaquette values and not the link variables. This is analogous to working with the field strength $F_{\mu,\nu}$ instead of the vector potential A_μ in the continuum theory. However, the fact that the underlying model is a gauge model imposes a condition on the plaquette variables that must be satisfied at all times—the number of frustrated plaquettes on an elementary cube must be even. This constraint is analogous to the Bianchi identity $dF=0$ in the continuum; hence we will refer to it as BI. From BI it follows that every plaquette configuration consists of closed strings of frustrated plaquettes. In order to make these considerations more transparent, let us introduce the dual lattice. Each point of the dual lattice corresponds to an elementary cube of the original one and each dual link corresponds to a plaquette of the original lattice. The value assigned to a dual link is equal to the value $-E(p)$ of the corresponding plaquette p, i.e., -1 if the plaquette is frustrated and $+1$ otherwise. In this way each configuration of the dual lattice corresponds to a single configuration of the original lattice modulo gauge transformations. The BI constraint can be expressed as a constraint on the links of the dual lattice in the following way: The number of frustrated links attached to a point of the dual lattice must be even, that is, the configurations on the dual lattice consist of closed loops of frustrated links. Configurations, on both the original and the dual lattices, that satisfy the BI constraint will be called *valid*. It is important to notice that this duality is not equivalent to the duality between the partition functions of the Ising model and Z_2 lattice gauge theory in three dimensions; as pointed out above, the latter duality cannot be expressed as a duality between configurations.

2.2. The Algorithm

The algorithm presented in this work is a *legitimate* MC method in the sense that it generates a Markov chain of configurations that is ergodic, and each configuration appears with a probability which is proportional to the Boltzmann factor $e^{-\beta\mathscr{H}}$. Given an *old* configuration C_n, we will describe the stochastic rules for choosing the next configuration in the Markov chain—the *new* configuration C_{n+1}.

The first part of our procedure consists in *freezing* or *deleting* all the terms of the Hamiltonian $\mathscr{H}$, i.e., the elementary plaquettes of the lattice. During this part, each plaquette is visited once and a stochastic decision is made to either freeze or delete it, with probabilities that depend on its energy in the old configuration C_n. The plaquette p is deleted with probability

$$P_d(p) = e^{\beta[-1+E(p)]} \tag{5}$$

and frozen with probability

$$P_f(p) = 1 - P_d(p) \tag{6}$$

where $E(p)$ stands for the energy of the plaquette p in C_n. From this choice of probabilities it follows that a plaquette that was frustrated in C_n is always deleted. On the other hand, a plaquette that was satisfied is sometimes frozen and sometimes deleted, with probabilities depending on the inverse temperature β. This part of the algorithm is analogous to putting virtual bonds in the original cluster algorithm developed for the Ising model.[11]

The second part of the algorithm consists in choosing a new valid plaquette configuration C_{n+1}. After all plaquettes have been either frozen or deleted in the first part we can partition the set of valid plaquette configurations into two subsets A and B.

A: Those valid plaquette configurations that differ from the old configuration C_n only in values of deleted plaquettes. In terms of plaquette values A is the set of valid configurations in which $-E(p) = +1$ for all frozen plaquettes.

B: Valid configurations that differ from C_n in values of one or more frozen plaquettes.

Our algorithm picks a random configuration C_{n+1} from subset A such that all configurations in A have the same probability to be chosen. In the next subsection we explain how this is accomplished in practice. This second part of the algorithm concludes one MC update of the lattice. If more updates are needed, one should repeat the whole procedure with C_{n+1} as the old configuration and so on.

The fact that our algorithm maintains detailed balance follows directly from the proof in refs. 12–14. In our case the terms in the Hamiltonian that are frozen or deleted are the plaquette energies. Our procedure is clearly ergodic, since every valid plaquette configuration can be reached in one sweep from any other valid configuration. The only thing we still have to proove is that the second part of the algorithm indeed picks a *random* configuration from the set A.

2.3. The Random Choice of a New Plaquette Configuration

In this subsection we describe how a valid plaquette configuration C_{n+1} is picked at random from A, the subset of all valid plaquette configurations that differ from C_n only in values of plaquettes that were deleted in the first part of the procedure.

We find it convenient to describe this part of the algorithm in terms of the dual problem. The links dual to the deleted plaquettes together with their endpoints, corresponding to elementary cubes of the original lattice, form a graph that we call the *graph of deletions*. To go from the configuration C_n to a new configuration C_{n+1}, both in the set A, we have to choose new values for the dual links on this graph. In general, the graph of deletions consists of several disconnected parts that can be treated separately. Each disconnected part is made up of a number of loops and some *dangling* ends; links may be part of more than one of these loops. A proper choice of C_{n+1} is obtained by assigning to each of the loops a random value (-1 or $+1$). The dual links are then assigned new values according to the following rules. Those links that are part of a dangling end are assigned a value of $+1$. The rest of the dual links are part of loops. For each of these links we calculate the product of ± 1 values of all the loops it is part of. The value of the product is assigned to the link.

In order to make the identification and manipulation of loops manageable and to turn these ideas into a practical algorithm, we were led to introduce some additional concepts. For each disconnected part of the graph of deletions we construct a *spanning tree*—a subgraph of deletions which connects all nodes of this connected part, but does not contain any closed loops. We will prove that the dual links of the graph of deletions

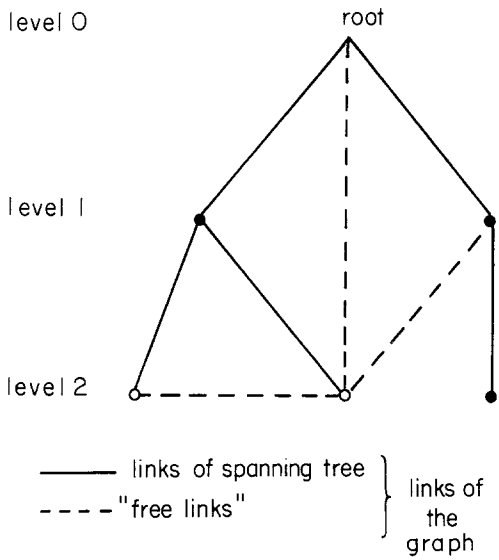

Fig. 1. A graph of deletions and its spanning tree. Links of the spanning tree are denoted by solid lines and free links by dashed lines.

that do not belong to the spanning tree may always be given arbitrary independent values without violating the BI constraint; for this reason they will be called *free links*. Moreover, we will show that, once all free links have been assigned a random value, the values of the links of the spanning tree are uniquely determined by BI.

In practice we proceed in the following way. For each disconnected part of the graph of deletions we build a spanning tree (see Fig. 1), and an arbitrary node of this tree is selected to be its *root*. The *level* of a node of the spanning tree is now defined as the number of links of the tree that make up the connection along the tree between its root and this node. As a consequence, a link of the spanning tree always connects two nodes that are one level apart and because spanning trees do not contain closed loops there is never more than one link of the tree directly connecting a particular node to nodes of the next lower level. Now the free links of this disconnected part of the graph of deletions are assigned independent random values: $+1$ or -1 (see Fig. 2). By applying the BI constraint at the nodes of the highest level, we see that at each of these nodes the link connecting this node to a node of the next lower level is forced to have a particular value (see Fig. 3). Now we can apply BI at the nodes of the next lower level, and it is clear that the links connecting these nodes to nodes of a lower level are forced by BI to take a definite value: $+1$ or -1. This procedure can be continued down to the nodes of level 1, and in this way all links of the dual lattice are assigned a value (see Fig. 4). Contradictions and ambiguities cannot arise, because applying BI at a node of the tree at level 1 or higher always forces one and only one link. By this construction BI is guaranteed to hold at all nodes of the dual lattice except possibly at the root of the spanning tree. We now show that BI holds at the root as

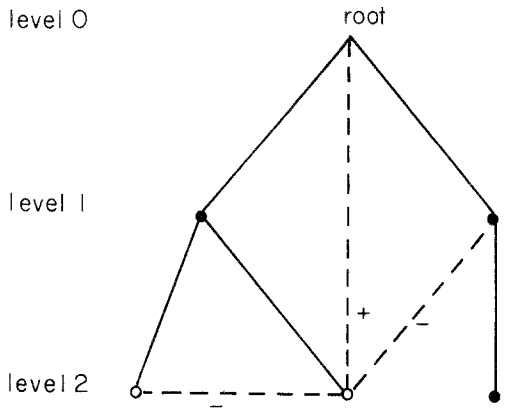

Fig. 2. Assignment of random values $+1$ or -1 to the free links.

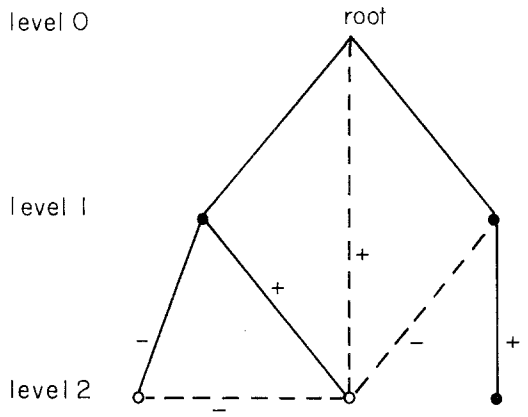

Fig. 3. Free links of the highest level are assigned the values $+1$ or -1 using the BI constraint.

well. Let us call the part of the spanning tree that connects a node to nodes of higher levels without passing through nodes of lower levels the *subtree* of this node. We can then define the *frustration number* of a node as the number of times the subtree of that node is touched by frustrated free links. The procedure we presented above forces each link of the tree to take the value $(-1)^n$, where n is the frustration number of the highest level endpoint of the link. At the root, k links belonging to the graph of deletions meet. A number l out of these k links are part of the spanning tree, while the $k-l$ remaining ones are free links, m of which are frustrated. The

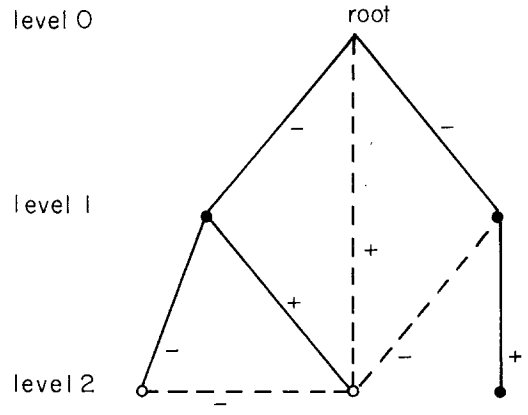

Fig. 4. A possible assignment of all the links on the graph of deletions.

product of values of all the links meeting at the root can now be written as

$$(-1)^{n_1}(-1)^{n_2}\cdots(-1)^{n_l}(-1)^{m}=(-1)^{(n_1+n_2+\cdots+n_l+m)} \tag{7}$$

where $n_1 \cdots n_l$ are the frustration numbers of the l nodes of level 1. From the fact that each frustrated free link touches the spanning tree in two nodes, we obtain

$$(-1)^{(n_1+n_2+\cdots+n_l+m)}=(-1)^{2n_f} \tag{8}$$

where n_f is the number of frustrated free links. Consequently, the product of values of all the links meeting at the root of the spanning tree is equal to $+1$. Therefore BI is satisfied at the root as well, and our method produces only valid configurations that belong to the set A. Moreover, our proof shows that each configuration in A is uniquely determined by the values of the free links, and hence is produced by our algorithm.

Since any configuration of the free links is generated by our procedure, and all these configurations are produced with the same probability, we conclude that in the second part of our algorithm we generate a random configuration from the set A.

2.4. Summary and Final Remarks

Let us summarize the major steps in a single MC update of the lattice:

1. Each plaquette is either frozen or deleted in a stochastic manner that depends only on the energy of the plaquette in the present configuration.
2. The graph of deletions is constructed on the dual lattice and the values of the rest of the links, corresponding to frozen plaquettes, are set to $+1$.
3. For each disconnected part of the graph of deletions a spanning tree is constructed and a root is selected.
4. Free links are given random values.
5. New values of the rest of the links are determined using BI repeatedly.

This concludes our detailed description of the stochastic cluster algorithm for the MC simulations of Z_2 lattice gauge theory. We proved that it is a legal updating procedure, i.e., it is ergodic and it generates configurations with the correct probability distribution. Before we discuss numerical tests of the method (see next section), two remarks are in order. First, it

is important to notice that a spanning tree can be constructed in a number of operations which is proportional to the volume of the system. Therefore, the amount of computer time needed to perform a single sweep of our MC algorithm grows linearly with the volume.

The second remark concerns generalizations of the correspondence between gauge models and tube dynamics. One can consider a D-dimensional square lattice with a Z_2 variable defined on each d-dimensional element of this lattice ($d=0 \rightarrow$ points $\rightarrow$ Ising models, $d=1 \rightarrow$ links $\rightarrow$ gauge theories, etc.). One defines the energy of each configuration as the sum of energies of all the $(d+1)$-dimensional elements of the lattice. The energy of each $(d+1)$-dimensional element is defined as a product of values of the d-dimensional elements forming its boundary. The dual construction described above creates closed, extended objects of dimensionality $D-d-1$ on the dual lattice. In particular, for $d=D-3$ one obtains the Euclidean partition function of a discretized system of strings. For $d=D-4$ one obtains membranes. The case $D=3$, $d=0$ was first observed by Polyakov.[20]

3. SIMULATIONS

In order to determine the properties of the stochastic cluster algorithm, we performed several long runs with our carefully tested program on CRAY X-MP/48 and CRAY Y-MP/832 computers. We used finite-size scaling [see Eq. (2)] to determine the dynamic critical exponent z. In order to do this, we carried out all our simulations at the critical coupling of the infinite system, $\beta_{cr}=0.7614$. This value was obtained from the duality relation

$$\beta_{Z_2\,\text{gauge}} = -\tfrac{1}{2}\ln[\tanh(\beta_{\text{Ising}})] \tag{9}$$

using the critical coupling of the Ising model $\beta_{\text{Ising}}=0.221654$.[21] The number of MC sweeps for the different lattice sizes were 400,000 (8^3 lattice), 100,000 (16^3), 40,000 (32^3), and 10,000 (64^3). From our data we calculated the time-delayed energy–energy correlation function

$$C_E(t)=\frac{\langle E(0)\,E(t)\rangle - \langle E\rangle^2}{\langle E^2\rangle - \langle E\rangle^2} \tag{10}$$

where $E(t)$ is the energy per unit volume at time step t. Errors in the autocorrelation function and autocorrelation time τ were calculated by dividing our data into independent bins. We determined autocorrelation

times by fitting the long-time behavior of $C_E(t)$ to a simple exponential decay

$$C_E(t) = Ae^{-t/\tau} \tag{11}$$

Our results for τ are presented in Fig. 5 as a function of the linear size of the lattice L. Using Eq. (2), we calculated the dynamic critical exponent and found $z = 0.73 \pm 0.06$.

In order to compare the new method with the local Metropolis algorithm, we performed runs with a fully vectorized program on four different lattice sizes: 8^3, 12^3, 16^3, and 24^3. For this algorithm we determined z not only from the energy autocorrelation function, but also from the Polyakov-loop autocorrelation function. The data are shown in Fig. 6. Our best estimate for z from both correlation functions is $z = 2.5 \pm 0.3$. This is larger than the value 2 that is naively expected for local MC algorithms and also larger than 2.03, the value obtained by Metropolis for the Ising model in three dimensions.[22] Due to critical slowing down, we needed about 7×10^6 Metropolis sweeps on the 24^3 lattice to determine τ with 10% error. It is clear that much more computer time would be required to

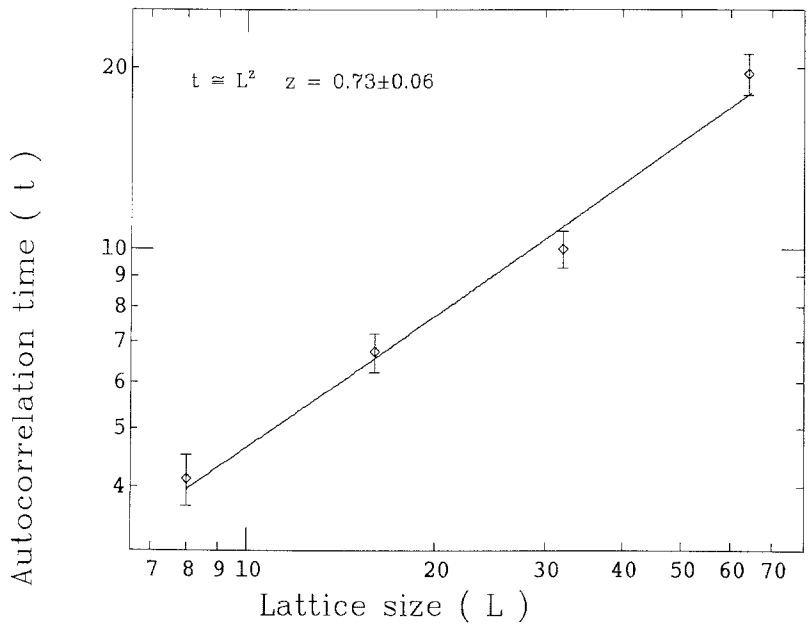

Fig. 5. Results of Monte Carlo simulation of Z_2 lattice gauge theory in three dimensions at the critical point. The stochastic cluster algorithm was used. Autocorrelation times of plaquette energy are plotted vs. lattice size.

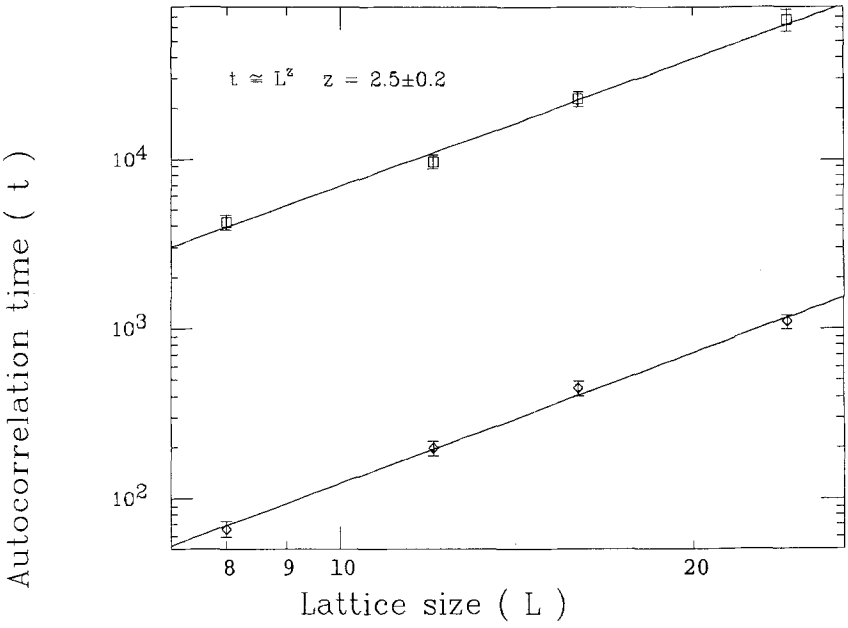

Fig. 6. Results of Monte Carlo simulation of Z_2 lattice gauge theory in three dimensions at the critical point. The local Metropolis update algorithm was used. Autocorrelation times for Polyakov loop (cubes) and for plaquette energy (diamonds) are plotted vs. lattice size.

determine z more accurately. We want to point out that since our stochastic cluster algorithm is gauge invariant, the autocorrelation function for the Polyakov loops is always zero.

Clearly, stochastic cluster algorithms are more efficient than local algorithms for large lattices, but they are much more complicated, and usually require more operations per spin update. Moreover, stochastic cluster algorithms do not vectorize well. It is important, therefore, to determine the size of the lattice at which our cluster algorithm becomes more efficient than the Metropolis method. This depends on the type of computer used. On one processor of the CRAY Y-MP, the Metropolis program we used updates 0.34×10^6 spins per second in the scalar mode and 4.2×10^6 in the vector mode during a simulation of a 32^3 lattice. The cluster algorithm, on the other hand, performs 0.125×10^6 spin updates per second in the scalar mode and 0.142×10^6 in the vector mode. The speed of the Metropolis algorithm can still be increased by careful use of techniques such as multispin coding; the fastest Metropolis program for Z_2 lattice gauge theory of which we are aware reaches a speed of 11×10^6 spin updates per second on a CDC Cyber 205 computer.[23] It is also possible

to increase the speed of the cluster algorithm considerably by refining and partially vectorizing the part of the algorithm in which the graph of deletions and the spanning trees are constructed. Even with these improvements, the number of operations per spin update will be larger in our cluster algorithm by more than an order of magnitude. This advantage of the Metropolis algorithm is not significant, since its dynamic critical exponent is large. In our simulations the cluster algorithm was more efficient than the vectorized Metropolis algorithm for $L > 10$. Extrapolating our results to $L = 100$, we find that for a simulation of Z_2 gauge theory on a 100^3 lattice, our algorithm would be at least 25 times more efficient in real computer time than the fully vectorized Metropolis algorithm.

4. DISCUSSION AND CONCLUSIONS

The results we presented in the previous section demonstrate that our stochastic cluster algorithm is superior to the traditional Metropolis algorithm as soon as the correlation length exceeds the value $\xi \approx 10$. The physical reason for the high efficiency of the cluster algorithm lies in the fact that it performs large-scale moves. It is well known that the creation and destruction of closed strings of frustrated plaquettes (closed loops of frustrated links on the dual lattice) play an important role in the pseudodynamics of the simulation of Z_2 lattice gauge theory near criticality. A very large number of Metropolis sweeps is required to generate or destroy a large, closed string of frustrated plaquettes because of the local nature of the algorithm. On the contrary, in our algorithm all frustrated plaquettes are deleted and consequently the probability to destroy such a closed string is large. Since our algorithm satisfies detailed balance the probability to create such strings is also large.

Successful stochastic cluster algorithms have also been developed for the Ising and the ϕ^4 theory in three dimensions,[11,15] both models are in the same universality class with Z_2 lattice gauge theory. It is striking that in all of these cases the dynamic critical exponent is reduced to the same value within error bars. This strongly supports the conjecture of dynamic universality in stochastic cluster algorithms. Recently the stochastic multigrid Monte Carlo method that was known to reduce z for the Ising model in two dimensions was successfully applied to this model in three dimensions as well; CSD was completely eliminated ($z = 0$ within error bars) using a W-cycle with rescaling factor $b = 2$. We therefore expect that a multigrid version of our algorithm will eliminate CSD completely for Z_2 lattice gauge theory.

The principles that guided us in developing our method seem to form

the basis for other stochastic cluster algorithms as well. First of all, physical understanding of the pseudodynamics of simulations near criticality is essential, i.e., the large-scale, low-energy excitations whose creation and destruction dominate the pseudodynamics must be identified. The first step of the stochastic cluster algorithm is a *freeze/delete* decision for all or part of the terms of the (possibly reformulated) Hamiltonian. The second step consists in the identification of the set of configurations of the system that do not violate the constraints imposed by frozen interactions. In the last step the new configuration is chosen from this set either at random, when all terms in the Hamiltonian are subjected to the freeze/delete decision, or according to the probability distribution governed by the remaining terms in the Hamiltonian. The success or failure of this method hinges fully on the choice of the terms in the Hamiltonian that are subjected to the freeze/delete decision. A successful choice is such that the constraints imposed by the frozen terms do not prevent the large-scale low-energy excitations that dominate the pseudodynamics of the simulation from being destroyed and created with a large probability.

The development of the Z_2 cluster algorithm is part of a continuing effort to apply stochastic cluster methods and multigrid ideas to a wide range of statistical models. We believe it is a step forward in the development of powerful global updating schemes for more realistic lattice gauge theories. In this context it may be useful to concentrate on Z_2 subgroups or to attempt the creation and annihilation of flux tubes. Work is in progress on the development of multigrid versions for Z_2 lattice gauge theories and the ϕ^4 model in order to reduce the dynamic critical exponent even further.

The existence of a gauge-invariant multiscale algorithm raises the hope that in the context of QCD it will shed light on the confining mechanisms which make hadrons rather than quarks the relevant degrees of freedom.

ACKNOWLEDGMENTS

We thank Achi Brandt for continuous fruitful collaboration and for teaching us about multiscale dynamics in MC simulations. During the development of the algorithm we benefited from helpful discussions with M. Marcu.

We thank the HLRZ in Jülich for a generous amount of computer time that enabled us to carry out most of the simulations on their CRAY X-MP/48 and CRAY Y-MP/832 computers. P.G.L. thanks Prof. Y. Frishman and the Einstein Center for financial support that made a three-week stay at the hospitable Weizmann Institute possible; R.B. thanks

Prof. F. Gutbrod and the HLRZ for a kind invitation and financial support for travel and a two-week stay at the HLRZ. S.S. is the incumbent of the Recu Career Development Chair.

REFERENCES

1. N. Metropolis, A. W. Rosenbluth, A. H. Teller, and E. Teller, *J. Chem. Phys.* **21**:1087 (1953).
2. P. C. Hohenberg and B. I. Halperin, *Rev. Mod. Phys.* **49**:435 (1977).
3. F. Fucito and S. Solomon, *Phys. Lett.* **134B**:235 (1984).
4. F. Fucito and S. Solomon, *Phys. Lett.* **134B**:230 (1984).
5. F. Fucito and S. Solomon, *Nucl. Phys. B* **251**:505 (1985).
6. E. S. Hood, A. Redondo, S. Solomon, and Y. Zeiri, *Phys. Rev. Lett.* **55**:634 (1985).
7. P. W. Kasteleyn and C. M. Fortuin, *J. Phys. Soc. Jpn.* (Suppl.) **26**:11 (1969).
8. M. Sweeny, *Phys. Rev. B* **27**:4445 (1983).
9. H. Meyer-Ortmanns, *Z. Phys. C* **27**:553 (1985).
10. J. Goodman and A. D. Sokal, *Phys. Rev. Lett.* **56**:1015 (1986).
11. R. H. Swendsen and J.-S. Wang, *Phys. Rev. Lett.* **58**:86 (1987).
12. A. Brandt, Talk at the Third Copper Mountain Conference on Multigrid Methods (April 1987).
13. D. Kandel, E. Domany, A. Brandt, D. Ron, and Y. Loh, *Phys. Rev. Lett.* **61**:1591 (1988).
14. D. Kandel, E. Domany, and A. Brandt, *Phys. Rev. B*, to appear.
15. R. C. Brower and Pablo Tamayo, *Phys. Rev. Lett.* **62**:1087 (1989).
16. W. Klein, T. Ray, and P. Tamayo, *Phys. Rev. Lett.* **62**:163 (1989).
17. U. Wolff, *Phys. Rev. Lett.* **62**:361 (1989); *Nucl. Phys. B* **300**:501 (1988); DESY Report 88-144 (October 1988); DESY Report 89-021 (February 1989).
18. A. Brandt, R. Ben-Av, and S. Solomon, preprint WIS-88/67/OCT-PH, *Nucl. Phys. B*, to appear.
19. R. Ben-Av and S. Solomon, preprint WIS-89/81/JAN-PH.
20. A. M. Polyakov, *Gauge Fields and Strings* (Horwood Academic Publishers, 1987).
21. G. S. Pawley, R. H. Swendsen, D. J. Wallace, and K. G. Wilson, *Phys. Rev. B* **29**:4030 (1984).
22. S. Wansleben and D. P. Landau, *J. Appl. Phys.* **61**:3968 (1987).
23. S. Wansleben, *Phys. Lett.* **148B**:461 (1984).