Random surface dynamics for Z_2 gauge theory

R. C. Brower
Department of Electrical, Computer and Systems Engineering, Boston University,
44 Cummington Street, Boston, Massachusetts 02215
and Physics Department, Boston University, 590 Commonwealth Avenue, Boston, Massachusetts 02215

Suzhou Huang
Physics Department, Boston University, 590 Commonwealth Avenue, Boston, Massachusetts 02215

(Received 19 July 1989)

A new Monte Carlo dynamics is proposed for Z_2 lattice gauge theory that reduces the dynamical exponent for critical slowing down in three dimensions to $z=0.61\pm0.05$ in comparison with a heat-bath exponent of $z=2.1\pm0.2$. The dynamics is based on plaquette percolation and a nonlocal Monte Carlo update rule analogous to the Swendsen-Wang algorithm for the Potts model. However, the acceleration mechanism for the gauge theory, unlike the spin models, is driven by the percolation and topology of extended surfaces, not the percolation of connected clusters.

It has been known for a long time that scalar field theories and Brownian motion (or random paths) are intimately connected. A less-well-understood (but presumably equally valid) connection exists between gauge-field theories and random surfaces. For instance, in lattice gauge theories, the strong-coupling expansion can be expressed as a sum over interacting random surfaces. It is widely conjectured that the leading term for renormalized continuum SU(N) gauge theory at large N is an as-yet-undiscovered relativistic string, with perturbative interactions in higher orders of $1/N$.

Here we wish to exploit the connection between gauge theories and random surfaces to find faster Monte Carlo algorithms. Our approach uses plaquette percolation to form the surfaces and a dynamics analogous to the nonlocal Monte Carlo Swendsen-Wang[1] (SW) algorithm, which has been shown to drastically reduce critical slowing down.

The SW algorithm is based on a detailed understanding of the Coniglio-Klein[2] site bond percolation clusters and the Fortuin-Kasteleyn[3] mapping of the Ising (or Potts) model into these random clusters. On the other hand, the theoretical understanding of plaquette percolation is less complete. It has been shown,[4] however, that one can identify a confining-deconfining phase transition for unrestricted random (or Bernoulli) plaquette percolation on a cubic lattice. If one considers a closed (Wilson) loop, there is a critical percolation rate p^* at which the probability of forming a continuous sheet spanning the loop makes a transition from area-to-perimeter law. Thus it is plausible that the random surface formed by sheet percolation provides a mechanism analogous to the divergent connectedness length in site bond percolation.

We apply these ideas to formulate an acceleration dynamics for the Z_2 lattice model in $d=3$ dimensions, which reduce the dynamical exponent from $z=2.1\pm0.2$ for heat bath to $z=0.61\pm0.05$. A very similar algorithm can be formulated for the Z_3 gauge theory as well.

Also we are optimistic that extensions of our Z_2 or Z_3 plaquette percolation dynamics reported here can be found for the continuous Abelian and non-Abelian gauge theories of particle physics. Our optimism is based on the recent extension of bond percolation methods to continuous field theories by Brower and Tamayo[5] (ϕ^4 theory) and by Wolff[6] [O(n) spin models] which successfully mapped (or embedded) Ising variables into the continuous field.

Plaquette Percolation. To formulate our dynamics, we begin by demonstrating the equivalence between the Z_2 gauge theory and an appropriately weighted ensemble of percolated plaquettes.

Z_2 gauge theory is defined by the probability distribution

$$P_G[s]=\frac{1}{Z_G}\exp\left[\beta\sum_P(s_{ij}s_{jk}s_{kl}s_{li}-1)\right], \qquad (1)$$

where the sum extends over all plaquettes on a simple cubic lattice. As a shorthand, we will write the product of spins over the links around a plaquette P (i.e., on the boundary, ∂P) as s_P. By rewriting the gauge distribution as a product over plaquettes

$$P_G[s]=\frac{1}{Z_G}\prod_P[(1-e^{-2\beta})\delta_{s_P,1}+e^{-2\beta}], \qquad (2)$$

we can introduce additional percolation variables $n_P=0,1$ on each plaquette to arrive at a joint distribution:

$$P_J[s,n]=\frac{1}{Z_J}\prod_P[p\delta_{s_P,1}\delta_{n_P,1}+(1-p)\delta_{n_P,0}]. \qquad (3)$$

The percolation probability is $p=1-e^{-2\beta}$. Summing out the percolation variables, we see that gauge theory distribution $P_G[s]$ is one of the marginal distribution. If, instead, we sum out the gauge spins the other marginal distribution for the percolation variables is a new kind of "random cluster" model,[7,8] which we will refer to as the random sheet model:

$$P_{RS}[n]=\frac{1}{Z_{RS}}N_G\prod_{n_p=1}p\prod_{n_p=0}(1-p)\ . \tag{4}$$

The weight factor N_G is the number of configurations consistent with the constraint that all percolated plaquettes have $s_P=1$. In the case of the q-state Potts model, the analogous weight factor is just $(q)^{N_c}$ where N_c is the number of clusters. If we change variables, $s=e^{i\pi\sigma}$, both of these degeneracy factors can be viewed as the size of the null space under the linear constraint $d\sigma=0$ mod 2. The linear form $d\sigma$ is a discrete gradient in the Ising case and a discrete curl in the gauge case. Finding these solutions will be the crucial and most difficult part of our algorithm.

Update Algorithm. Various update procedures can be imagined which preserve detailed balance for the joint model and therefore equilibrium for the gauge model. We choose the transition matrix to be the product of the conditional probabilities for percolating plaquettes at fixed spin $P_J[n|s]$ followed by the conditional probability for flipping spins at fixed plaquette percolation $P_J[s|n]$:

$$W(s',n'\leftarrow s,n)=P_J[s'|n']P_J[n'|s]\ . \tag{5}$$

Our update algorithm consists of two steps: (1) Percolate all plaquettes at probability $p=1-e^{-\beta(1+s_P)}$; (2) flip all spins randomly subject to the constraint that $s_P=1$ remains fixed on all the percolated plaquettes.

You might suppose (erroneously) that the important length scale would relate to the size of the connected clusters of percolated plaquettes. In fact, the connectedness length of these clusters diverges at a much smaller $\beta\simeq0.2$ relative to the critical point ($\beta_c=0.7614$) of the Z_2 gauge theory. At the phase transition, $\beta=\beta_c$, almost all the links (about 96%) form one very big cluster. Incidentally, this feature is discouraging in terms of finding an efficient "single cluster" update scheme in the spirit of Wolff,[6] since almost always a random link belongs to a cluster of O(N), the lattice volume.

The correct way to define the relevant length scale is to generalize the feature of the bond percolation model that accounts for long-range correlations. In the random cluster model, two spins are correlated if and only if they lie in the same cluster. Equivalently one can say that the line connecting correlated spins in homologous to zero [see Fig. 1(a)]. This is generalized to the case of a Wilson loop. Nonzero contribution to the Wilson loop on a closed curve Γ in the random sheet model (4) can be shown to be nonzero if and only if there is a percolated sheet that spans the loop and lies totally inside the cluster. Thus you say the loop must be homologous to zero [see Fig. 1(b)].

To understand why this is the case, imagine the situation where there is no "percolated sheet" for a Wilson loop. Then we can find a solution to the constraints that changes the flux through some filament on the dual lattice that winds around this loop an odd number of times. It should be clear that the essential ingredient to gauge percolation dynamics is the topological structure of the percolation space. Since the 3D Z_2 gauge theory is dual to the three-dimensional (3D) Ising model, the standard order-disorder description of 3D confinement should be helpful in gaining a more complete understanding of the problem.

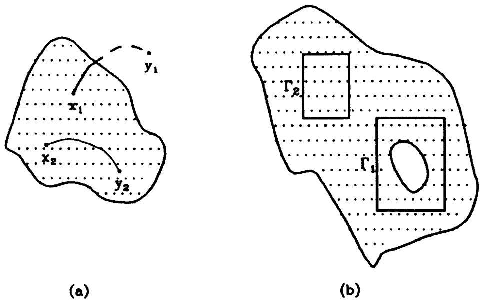

FIG. 1. The dots denote the percolated domains: (a) In the Potts model, spins at x_1 and y_1 have zero correlation, while spins at x_2 and y_2 have a correlation of one, because the connecting line lies inside the domain; (b) in the Z_2 gauge theory a Wilson loop for Γ_1 is zero, while the Wilson loop for Γ_2 is one, because there is an unbroken percolated surface spanning Γ_2.

Computational Methods and Results. As in the Swendsen-Wang algorithm, the percolation step (i) is trivial. We percolate the plaquettes, which satisfy the constraint that the link product on the plaquette is one, $s_P=1$, with probability $p=1-e^{-2\beta}$. Obviously, this process is $O(N)$, order of lattice volume N. The attempt to percolate a single cluster as in the Wolff algorithm leads also to an $O(N)$ step since the critical point is strongly into the ordered phase of cluster percolation.

In contrast with the spin models, step (ii) to solve the condition to find configurations with zero curl (or equivalently to find the null space of the incidence matrix) is more difficult. One must replace the Hoshen-Kopelman[9] (HK) cluster finding algorithm used in the spin models with a new algorithm. It should be kept in mind that, if we parametrize the efficiency of this computational step in terms of its complexity $O(N^{1+\zeta/d})$, the CPU time of our algorithm will actually scale as $t_{\text{CPU}}\sim L^{z+\zeta}$, where $L=N^{1/d}$ is the linear size of the system. As in the HK algorithm, we would again like to find an $O(N\ln^{\alpha}N)$ algorithm so the real algorithmic exponent is not shifted.

Our computational method can be described on the basis of a Gaussian elimination procedure for the incidence matrix:

$$\partial P\Longrightarrow\begin{array}{c|cccccc|}
 & l_1 & \overset{\overset{\partial * l}{\Downarrow}}{l_2} & l_3 & l_4 & \cdots & l_n \\
P_1 & 1 & 1 & 1 & 0 & \cdots & 0 \\
P_2 & 1 & 1 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
P_m & 0 & 0 & 0 & 0 & \cdots & 1
\end{array}\begin{bmatrix}\sigma_1\\ \sigma_2\\ \sigma_3\\ \sigma_4\\ \sigma_5\\ \vdots\\ \sigma_n\end{bmatrix}=0\ .$$

However, since the number of rows (percolated plaquettes) and columns (links) of the incident matrix are proportional to the lattice volume, we never actually write down the matrix to avoid the algorithm being of order volume squared. Instead the incident matrix is

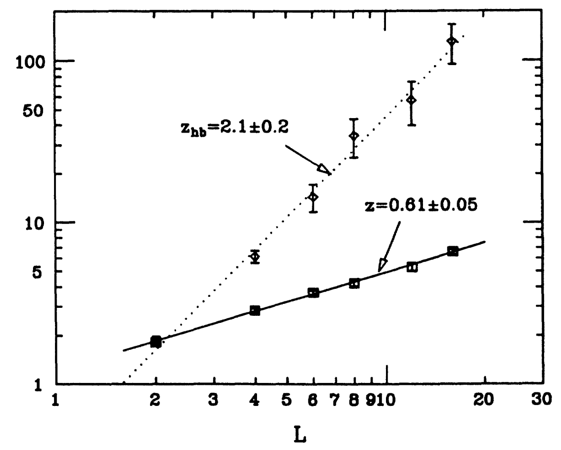

FIG. 2. A log-log plot of relaxation time τ vs linear size of the system L. The diamonds are the MC data for conventional heat bath with a fitted dynamical exponent $z_{\text{hb}}=2.1\pm0.2$. The squares are the MC data for the percolation algorithm with a fitted dynamical exponent $z=0.61\pm0.05$. The dotted line and solid line are meant to guide the eyes.

defined implicitly in terms of linked lists for the nonzero elements in the rows ∂P and columns $\partial^* l$.

To reduce the size of the linked lists, we preprocess them by eliminating entries associated with pure gauge choices. This is achieved first by gauge fixing in a maximal tree which deletes associated links and then by iteratively getting rid of "dead end" plaquettes, which have only one nongauge fixed link. To appreciate the importance of this step, let us imagine that all the plaquettes are percolated. In this extreme case, the preprocessing can cut the number of rows and number of columns by a factor of L each, leaving only those links and plaquettes associated with a two-surface of nontrivial topology, analogous to toroidal Polyakov loops in the case of two-dimensional lattice with periodic boundary condition.

Finally, we manipulate the linked lists with exclusive OR (XOR) operations that implement Gaussian elimination, mod 2. Our procedure is as follows. Consider finding the null space defined by the equation $AX=0$. It is well known[10] that the operation in Gaussian elimination on the ith column can be represented by the replacement $A\rightarrow AP_i$ and $X\rightarrow P^\dagger X$, so that the upper full diagonalization is an ordered product, $A_D=AP_1P_2\cdots P_N$. The $P_i's$ are the individual Jacobi transformation matrices. However, in our problem the matrix of null vectors is not sparse. Unlike the cluster problem of Swendsen and Wang the null vectors cannot be orthogonalized with each spin contributing to a single null vector. Thus to avoid the $O(N^2)$ problem of computing the null vectors, we do not write this matrix down.

Instead we note that our goal is to find the spin state after flipping each null vector with 50% probability. By storing the sequence of operations $P_1, P_2, \dots, P_N$, we can reverse the order, so that the sequence carries us from the null space back to the original spins basis. It is this "time"-reversed Gaussian elimination that we use to flip the spins. Also since we are working in a Z_2 system (mod 2 arithmetic) at each step all components which are flipped an even number of times can be dropped. Hence our "time"-reversed flipping procedure takes fewer computational steps than the diagonalization step itself.

TABLE I. Autocorrelation time and computational complexity of the algorithm vs linear size of the system.

L	τ_E	τ_P	Sparseness	Stack
2	1.82(04)	1.53(05)	3.79	4.16
4	2.85(07)	2.40(07)	5.91	4.88
6	3.69(09)	3.21(11)	8.76	6.53
8	4.21(15)	3.73(08)	12.6	8.85
12	5.30(26)	4.81(21)	22.8	15.1
16	6.60(24)	6.22(25)	35.6	23.2

The relaxation time is found for the energy by measuring its autocorrelations

$$C(t)=\langle E(t)E(0)\rangle-\langle E(t)\rangle\langle E(0)\rangle\propto e^{-t/\tau} \qquad (6)$$

and similarly for the Polyakov loop. The dynamical exponent $z(\tau=kL^z)$, is extracted from a log-log plot of τ vs L as in Fig. 2. Our best estimate, using L up to 16, is $z=0.61\pm0.05$ for energy correlations and $z=0.66\pm0.05$ for the Polyakov loop correlations. As a comparison, a similar analysis for the conventional heat-bath algorithm is also plotted with $z_{\text{hb}}=2.1\pm0.2$ as expected for the Glauber dynamics.[11] The Monte Carlo iterations consisted of 20-K heat-bath cycles followed by 100-K percolation cycles, except for the 16^3 lattice which involved half as many cycles.

In order to determine how the CPU time scales with the system size, we measure the memory allocation in the stack normalized by the volume, and we measure the effective "sparseness" of our matrix as the number of XOR operations involved in adding row and column lists. Table I shows the numerical results for these quantities. The "sparseness" measure of CPU time scales as $L^{d+\zeta}$ where $\zeta\simeq1.4$ for each sweep over the whole lattice, in contrast with a nonsparse matrix algorithm of L^{3D} or the conventional heat-bath algorithm scaling as proportional to L^d. Of course our data is very rough and it could be consistent with $O(N\ln^a(N))$ behavior. More likely further optimization is necessary to get this. The complexity of diagonalizing such irregular sparse matrices depends on the order with which you select the rows and columns. Probably we need to consider further (as in our preprocessing) the locality of the physical problem in some kind of block renormalization scheme to further reduce the computational complexity.

In conclusion, we have studied a new dynamics based on the percolation of random sheets (or random surfaces) at their critical point that reduces critical slowing down. The critical exponent is nearly equal to the Swendsen-Wang number on the dual lattice for the 3D Ising model.

We can generalize this approach to Z_3 gauge theory, simply by replacing mod 2 condition by mod 3, so that it opens up the possibility of generalizations via discrete embedding into the center of the SU(2) or SU(3) gauge theories pertinent to field theories of particle physics. We are at present investigating these applications particularly with respect to the finite-temperature deconfinement phase transition. Since this transition is believed to be driven by disorder in the center of the group for the Polyakov loop the physics suggest that our approach will be effective. Preliminary numerical results support this contention.

We would like to acknowledge many useful conversations with Robert Edwards, Roscoe Giles, Robert Kotiuga, William Klein, and Alan Sokal.

[1] R. Swendsen and J. S. Wang, Phys. Rev. Lett. **58**, 86 (1987).

[2] A. Coniglio and W. Klein, J. Phys. A **13**, 2775 (1980).

[3] C. M. Fortuin and P. W. Kasteleyn, Physica **57**, 536 (1972).

[4] M. Aizenman, J. T. Chayes, L. Chayes, J. Frohlich, and L. Rosso, Commun. Math. Phys. **92**, 19 (1983).

[5] R. C. Brower and P. Tamayo, Phys. Rev. Lett. **62**, 1087 (1989).

[6] U. Wolff, Phys. Rev. Lett. **62**, 361 (1989).

[7] A. Maritan and C. Omero, Nucl. Phys. **B210** [FS6], 553 (1982).

[8] M. Aizenman and J. Frohlich, Nucl. Phys. **B235** [FS11], 1 (1984).

[9] J. Hoshen and R. Kopelman, Phys. Rev. B **14**, 3438 (1976).

[10] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing* (Cambridge University Press, Cambridge, England, 1988).

[11] P. C. Hohenberg and B. Halperin, Rev. Mod. Phys. **49**, 435 (1977).