Microcanonical Cluster Monte Carlo Simulation

Michael Creutz[a]
Physics Department, Brookhaven National Laboratory, Upton, New York 11973
(Received 2 March 1992)

I propose a numerical simulation algorithm for statistical systems which combines a microcanonical transfer of energy with global changes in clusters of spins. The advantages of the cluster approach near a critical point augment the speed increases associated with multispin coding in the microcanonical approach. The method also provides a limited ability to tune the average cluster size.

PACS numbers: 05.20.−y

Monte Carlo simulation is now a major tool for the study of both critical phenomena in condensed matter physics and nonperturbative field theory in particle physics. With continual tuning over the years, current algorithms are well adapted to today's supercomputers. Nevertheless, on large systems near a critical point, severe slowing of the evolution to independent states encourages the search for yet better methods. Two intriguing and apparently unrelated ideas for such improvement are cluster algorithms and microcanonical techniques.

Swendsen and Wang [1] built on the work of Ref. [2] and proposed a cluster flipping procedure as a way to make coherent long-range changes in a single Monte Carlo updating step. They were motivated by the rather slow probing of important long-distance physics by local algorithms when the couplings of a system are near a critical point. Later Wolff [3] presented a powerful variation on the approach, where at each step a single large cluster of variables is formed and modified.

Another useful scheme is the microcanonical Monte Carlo approach [4]. Here a set of additional variables, referred to as "demons," are introduced to transfer energy around the system. During the updating, the combined energy of the system of interest and the auxiliary variables is held absolutely constant. In this the temperature is an output of the simulation, being extracted from the distribution of demon energies. The primary advantage is that the demons can take a very simple form, and for discrete systems they can be treated entirely with simple bit manipulation. With many demons stored in a few computer words, an effective parallelization is possible on conventional serial computers. In simple cases no floating-point arithmetic is ever needed, and many individual spins can be updated in parallel via bitwise operations. This gives programs for the Ising model which run about an order of magnitude faster than conventional approaches [5].

In this paper I present a combination of these ideas, a cluster algorithm where the cluster growth is determined entirely by bit manipulations involving a set of microcanonical demons. The primary gain is increased simulation speed while retaining the advantages of a global updating scheme near a critical point. In addition, discussions of detailed balance are particularly straightforward in the microcanonical language, simplifying justification of variations in the cluster algorithms. The microcanonical approach also introduces new parameters for tuning the cluster size. As with the local microcanonical method, the temperature is determined as a function of the constant total system energy. Wolff [6] has considered another hybrid of the cluster and microcanonical approach, where, rather than the energy, the total area of the cluster boundaries was held fixed. This allowed him to directly and efficiently study dynamically diluted Ising systems.

In the infinite-volume limit, the canonical and microcanonical ensembles give rise to the same physics. The method discussed here maintains a conserved energy and thus is microcanonical in nature. The energy conservation is global, and indeed locally the dynamics is essentially canonical. While in principle ergodicity can be a problem with a microcanonical approach, this is expected to be a serious issue only in extreme circumstances, such as with very low temperature or small lattices. Ergodicity can be assumed by converting the algorithm to a canonical one with an occasional refreshing of the auxiliary variables, as discussed below.

For simplicity I begin with the simple Ising model. On each site i of an arbitrary lattice lies a spin variable σ_i taking values in $\{1,-1\}$. The energy of each bond is lowest if the neighboring spins have the same value, and is increased by two units for antiparallel spins. Thus I consider the Hamiltonian

$$H_\sigma = -\sum_{\{i,j\}} \sigma_i \sigma_j , \tag{1}$$

where $\{i,j\}$ denotes the set of nearest-neighbor pairs, each pair appearing once in the sum. I am interested in the statistical mechanics of this system at inverse temperature β, and thus consider the partition function

$$Z_\sigma = \sum_{\{\sigma\}} \exp(-\beta H_\sigma) . \tag{2}$$

On a two-dimensional square lattice this model has a second-order transition at $\beta = \frac{1}{2}\ln(1+\sqrt{2}) = 0.44068\ldots$ [7].

As in Ref. [4] I augment this system with a set of auxiliary variables called "demons." In that reference the demons were associated with the lattice sites. Here, how-

ever, I place them on the system bonds. Each of these demons carries a sack of energy which it can use to "flip" or change the state of the bond it occupies. Thus for each neighboring pair of sites i and j, I associate a demon energy $D_{ij} \equiv D_{ji} \geq 0$. For the Ising case the bond energies always change in steps of two, so I am free to restrict the demon energies to non-negative even integers. It is also convenient to place a capacity limit on the demon in the form of an upper bound for its energy $D_{ij} \leq D_{\max}$. The practical advantage of this constraint lies in the storage and manipulation of the demon energies with only a few bits. For example, with two bits per demon I can consider the individual demon energies to lie in the set $\{0,2,4,6\}$, and store 32 such demons in two 32-bit computer words. As with the local microcanonical approach, the upper limit on D_{ij} still leaves the algorithm exact, although it will modify the cluster shapes.

The total energy for the coupled system is

$$H = -\sum_{\{i,j\}} \sigma_i \sigma_j + D_{ij} . \tag{3}$$

The corresponding canonical partition function is

$$Z = \sum_{\{\sigma,D\}} \exp(-\beta H) . \tag{4}$$

This immediately factorizes into contributions from the lattice and the demons. It is only through a microcanonical constraint that these variables become coupled.

The algorithm consists of three parts: cluster growing, cluster flipping, and demon shuffling. For the demon shuffling I move the demons around to new locations. This can be done in an arbitrary way because the Hamiltonian in Eq. (3) leaves the demons uncoupled. Indeed, the separate bits of the demons are also uncoupled; thus, the first bits could be shuffled separately from the second bits. (For the test simulations below, the bits were shuffled together.) Because it is so arbitrary, the shuffling needs only insignificant computer time.

For the cluster growing, I divide the demons into two sets, "contented" and "sad." A demon is contented if it possesses an amount of energy which allows a change in the state of the currently occupied bond. Thus, if the neighboring spins are parallel and the demon has two or more units of energy, then it is happy. If the demon cannot accommodate the change in the bond energy, it is sad. This might occur for parallel spins if the demon has no energy, or for antiparallel spins if the demon's energy supply is full and cannot accept more. A cluster is now defined as a complete set of sites joined by sad demons. All bonds on the exterior boundary of such a cluster carry contented demons. The contentedness of the demon on any given bond corresponds closely with the bond occupation variables used in Ref. [2].

I now come to the cluster-flipping stage. Here I can either follow the approach of Swendsen and Wang [1] or the variation of Wolff [3]. In the former case, the lattice is divided into clusters as above, and with a random probability all spins in each cluster are either flipped or not. In the Wolff approach, a single random site is chosen, and the corresponding cluster has all of its spins inverted. This picks a given cluster with a probability proportional to its volume, and thus gives larger average cluster sizes. For the remainder of this discussion I consider this single-cluster approach. When the cluster(s) of spins is (are) flipped, the appropriate changes of the demon energies on the cluster edges must also be made. Thus the energy in Eq. (3) is absolutely conserved.

The justification of the procedure is a simple application of detailed balance. After a cluster is flipped, all demons retain their contented or sad state. Thus if we were to regrow the clusters, their shapes would be unchanged. If the demons are not moved and a cluster is grown from the same point, a second application of the algorithm will return exactly to the starting state. With an ensemble in equilibrium, all states of equal energy are equally likely. Under the algorithm the states of the coupled demon-spin system break up into pairs of degenerate states. These states just flip back and forth between each other, and remain equally likely. The demon-shuffling stage then changes the cluster breakup for later steps.

As discussed in Ref. [4], the microcanonical approach has no explicit coupling parameter in the algorithm. In equilibrium, the temperature is an output and depends on the total system energy. I can easily find it from the distribution of demon energies. Here the separate demon bits are uncoupled, so I can extract the inverse temperature from the expectation of a single demon bit. For example, I have

$$\beta = -\tfrac{1}{2} \ln(-1 + 1/P) , \tag{5}$$

where P is the average fraction of the demons with their lowest bit set.

The microcanonical approach is easily modified to canonical in the same way that the local microcanonical algorithm of Ref. [4] reduces to the canonical Metropolis *et al.* [8] procedure. For this reduction, I refresh the demons after each cluster step; i.e., I replace the demon energies with a random positive number selected with Boltzmann probability $\exp(-\beta D_{ij})$. Indeed, if the demon has no upper limit on his energy, the refreshing procedure recovers exactly the algorithms of Refs. [1] and [3].

As in the local case, for large volumes and when the demon scrambling takes place over long distances, the canonical and microcanonical evolutions become essentially indistinguishable. Thus for systems involving continuous dynamical variables, where random number generation represents only a minor part of the necessary arithmetic, there is no obvious advantage of the microcanonical approach over the canonical one. If the demon scrambling takes place over a short range, the energy conservation takes on a local character, and there could be important dynamical differences between the algo-

rithms.

The primary advantage of the approach lies with discrete systems, where all necessary arithmetic can be done entirely by bitwise operations. For example, on a 32-bit machine a group of 32 adjacent spins can be stored in a single computer word, I, and 32 two-bit demons can be stored in two computer words, $D1$ and $D2$. Using shifts and logical operations, the calculations to find the sad and contented demons and to grow the cluster can be done for all 32 demons simultaneously.

The gain here is less than in the local case because much of the time is spent growing the cluster. Once the sad bonds are found, this is the same as in the canonical algorithm. The cluster growth can still be done by bit manipulation, but with a porous cluster much of the work involves irrelevant sites. I note in passing that, unlike in the local approaches, no division of the lattice into independent sets of sites, such as a checkerboard, is involved.

Wolff [3] observed that for the Ising model his average cluster size is tied to the magnetic susceptibility. As the demons have upper as well as lower bounds on their energies, this connection is less precise here. In general, the clusters tend to be small at high temperatures where it is easy to satisfy the demons' desires, while they become increasingly dense at low temperatures where only a few demons will have the required energy to excite their bonds. In Fig. 1 consider the two-dimensional model and show the average cluster size as a function of beta for demons carrying from one to four bits. Note how the average cluster size increases as the number of demon bits is reduced. This dependence is essentially invisible with three or more bits. Indeed, occupancy of the higher bits is exponentially suppressed by a Boltzmann factor.

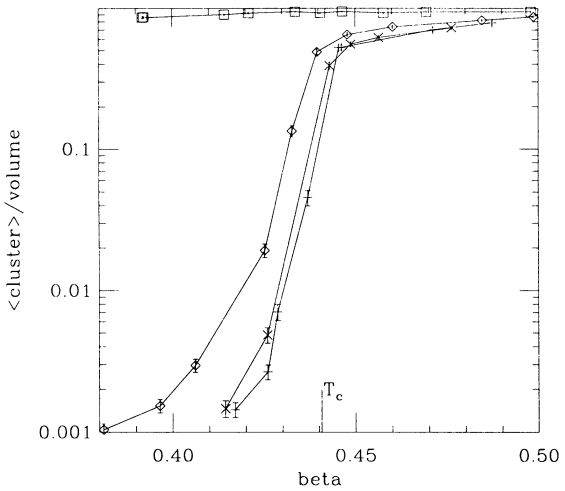

FIG. 1. The average cluster volume divided by the total system volume as a function of the inverse temperature. The results are for the two-dimensional Ising model on a 320 by 320 lattice. The squares, diamonds, crosses, and plus signs are for one- through four-bit demons, respectively.

Intuitively, the most independent configuration after a single step should be obtained when about half the spins are flipped. In the Ising case this occurs conveniently near the critical temperature. At high temperatures the clusters are quite small, and the algorithm seems to have no advantage over a conventional local approach. Conversely, at low temperatures the clusters dominate the lattice and primarily serve to flip the lattice back and forth between opposite magnetizations. Note, however, that if a hot lattice is rapidly quenched with a local algorithm to a low-temperature state where several distinct domains are frozen in, a switch to a cluster algorithm quickly generates clusters which fill single domains. In this case the approach is quite efficient at relaxing the system to its true ground state.

Figure 1 shows that with single-bit demons, for all energies the clusters tend to be quite large, and most spins flip at each step. While the algorithm thus will require more iterations to reach a truly independent configuration, it is perhaps worth noting that in this single-bit limit the algorithm is particularly simple. Contented demons, which form the cluster edge, are those where of the three bits consisting of the spins and the demon itself an odd number are set. This can be determined by a simple exclusive OR operation between the three bits.

Varying D_{max} gives some control over the average cluster size. Even for discrete models, this control is an essentially continuous parameter because not all demons need to have the same number of bits. In general, additional constraints on the demon energies work to increase the number of sad demons. As it is forbidden to exclude any unhappy demon bonds from the cluster, such constraints always tend to increase the average cluster size. Unfortunately, it is unclear how to modify the approach to obtain smaller clusters.

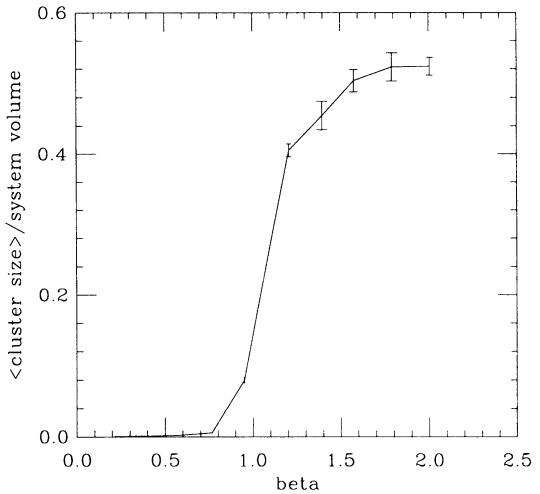

FIG. 2. The average cluster volume divided by the total volume as a function of the inverse temperature for the U(1) spin model in two dimensions on a 50 by 50 lattice.

In general, cluster algorithms are more limited in their applicability to systems where the spins are continuous variables. The usual approach is to imbed a Z_2 symmetry in the spin manifold [3]. The microcanonical idea directly adapts to such systems. In this case, however, the bit manipulation advantages of the discrete models are lost. In Fig. 2 I show the beta dependence of average cluster size for this algorithm as applied in two dimensions to the U(1) or "X-Y" model, where σ is a complex number of unit magnitude. At each stage, a fixed random direction in the group is chosen, and the group elements in the cluster reflected about it. At high temperatures the clusters are again small, but for couplings of order unity, where the primary physical interest in this model lies [9], the clusters cover a fair fraction on the system and should give an efficient decorrelation time. For this figure no upper bound was placed on the demon energies; so, the results should be nearly identical to the canonical approach.

In summary, I have presented a simulation scheme which combines features of the cluster algorithms of Refs. [1,3] with the fast microcanonical approach of Ref. [4]. This enables simulations done entirely by bit manipulation, permitting parallel operations on conventional computers. Furthermore, the introduction of the auxiliary demon variables provides a particularly simple way to justify detailed balance for cluster algorithms in general.

I thank A. Gocksch, P. Hsieh, Y. Shen, and A. Sokal for useful discussions.

[a]Electronic address: creutz@wind.phy.bnl.gov.

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

[2] P. W. Kasteleyn and C. M. Fortuin, J. Phys. Soc. Jpn. Suppl. **26**, 11 (1969); C. M. Fortuin and P. W. Kasteleyn, Physica (Utrecht) **57**, 536 (1972); C. M. Fortuin, Physica (Utrecht) **58**, 393 (1972); **59**, 545 (1972).

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

[4] M. Creutz, Phys. Rev. Lett. **50**, 1411 (1983).

[5] M. Creutz, G. Bhanot, and H. Neuberger, Nucl. Phys. **B235 [FS11]**, 417 (1984); M. Creutz, P. Mitra, and K. J. M. Moriarty, Comput. Phys. Commun. **33**, 361 (1984); J. Stat. Phys. **42**, 823 (1986); M. Creutz and K. J. M. Moriarty, Comput. Phys. Commun. **39**, 173 (1986); M. Creutz, K. J. M. Moriarty, and M. O'Brien, Comput. Phys. Commun. **42**, 191 (1986).

[6] U. Wolff, Nucl. Phys. **B300 [FS22]**, 517 (1988).

[7] L. Onsager, Phys. Rev. **65**, 117 (1944); T. D. Schultz, D. C. Mattis, and E. H. Lieb, Rev. Mod. Phys. **36**, 856 (1964).

[8] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).

[9] R. Gupta, J. DeLapp, G. G. Batrouni, G. C. Fox, C. F. Baillie, and J. Apostolakis, Phys. Rev. Lett. **61**, 1996 (1988).