JOURNAL OF COMPUTATIONAL PHYSICS 17, 10–18 (1975)

A New Algorithm for Monte Carlo Simulation of Ising Spin Systems*

A. B. BORTZ

Belfer Graduate School of Science, Yeshiva University, New York, New York 10033

M. H. KALOS

Courant Institute of Mathematical Sciences, New York University, New York, New York 10012

AND

J. L. LEBOWITZ

Belfer Graduate School of Science, Yeshiva University, New York, New York 10033

Received June 27, 1974; revised September 4, 1974

We describe a new algorithm for Monte Carlo simulation of Ising spin systems and present results of a study comparing the speed of the new technique to that of a standard technique applied to a square lattice of 6400 spins evolving via single spin flips. We find that at temperatures $T < T_c$, the critical temperature, the new technique is faster than the standard technique, being ten times faster at $T = 0.588\ T_c$. We expect that the new technique will be especially valuable in Monte Carlo simulation of the time evolution of binary alloy systems. The new algorithm is essentially a reorganization of the standard algorithm. It accounts for the a priori probability of changing spins before, rather than after, choosing the spin or spins to change.

INTRODUCTION

The Monte Carlo method has been used extensively to study spin one-half Ising systems with nearest neighbor interactions [1–8]. It has been used to study equilibrium properties [1] as well as the time evolution [2–8] of spin lattice systems having interactions described by the hamiltonian

$$\mathscr{H} = -J \sum_{i,j} S_i S_j - H \sum_i S_i ; \qquad S_i = \pm 1, \tag{1}$$

* Research supported by AFOSR Grant #73-2430A and AEC Contract AT(11-1)-3077.

where the summation indices refer to sites on a d-dimensional lattice (usually with toroidal boundary conditions) and the first summation is restricted to nearest neighbor spins. New configurations of the lattice are generated from an initial configuration by reversing single spins (single spin flip studies [1–4]), or by interchanging pairs of unlike nearest neighbor spins (spin exchange studies [5–8]). Because a spin exchange leaves the number of up and down spins unchanged, it has been used to study binary alloys, identifying spin up with atomic species A and spin down with atomics species B.

The standard Monte Carlo algorithm generates a sequence of configurations as follows. (1) Given a configuration $\mathscr{C}$, the program selects at random with uniform probability a spin (or pair of nearest neighbor spins). (2) The program computes the probability P of reversing (interchanging) the spin(s) according to a thermodynamically reasonable formula. The formula reflects the fact that if the reversal (interchange) produces configuration $\mathscr{C}'$, the equilibrium ratio of the probability of $\mathscr{C}'$ to the probability of $\mathscr{C}$ is $\exp(-\Delta E/kT)$, where ΔE is the energy of $\mathscr{C}'$ less the energy of $\mathscr{C}$ and kT is Boltzmann's constant times the absolute temperature. (3) The program chooses at random a fraction R with uniform probability over the interval [0, 1). (4) Then it performs the reversal (interchange) if $R \leqslant P$. In time evolution problems, P is usually computed as

$$P = x/(1 + x); \qquad x = \exp(-\Delta E/kT). \tag{2}$$

An attempted reversal (interchange) at any lattice site (pair) represents the same amount of time as an attempted reversal (interchange) at any other site (pair). The number of attempts can then be used as a measure of time. In equilibrium studies, P is usually computed [9] as

$$P = \begin{cases} 1, & \text{if } \Delta E \leqslant 0, \\ \exp(-\Delta E/kT), & \text{otherwise,} \end{cases} \tag{3}$$

speeding up the program by eliminating step (3) if $\Delta E \leqslant 0$ and increasing the probability of generating a new configuration at the expense of losing the time variable. Both (2) and (3) describe a Markov process whose stationary state is the equilibrium distribution with fixed magnetic field (magnetization).

When the system is in or near equilibrium or some metastable state, the rate of generating new configurations using the standard algorithm becomes quite slow, since P is then usually very small. In ferromagnetic ($J > 0$) spin exchange problems, the rate of generating new configurations is even slower, since it becomes very unlikely that a pair of nearest neighbors, selected at random, will be different. Thus the usefulness of the standard algorithm is limited by the low probability of generating new configurations.

We describe here an algorithm which generates a new configuration with every choice of spin (neighbors) without changing the behavior of the process described by (2). It does so by accounting for the a priori probability of reversal (interchange) before, rather than after, choosing the site (pair) to change. In that sense it is a reorganization of the standard algorithm. Primarily because of extra bookkeeping, the computation time to generate a new configuration using this algorithm is longer than the computation time to attempt a new configuration using the standard algorithm. However, in situations in which the standard algorithm usually rejects the test configuration, the algorithm we present here generates new configurations much faster.

The n-Fold Way

The new algorithm, hereafter called the n-fold way, is based on the fact that there is a small number n of classes of sites (neighbor pairs) classifying sites (pairs) by their probability of reversal (interchange). For example, in the single spin flip square lattice with periodic boundary conditions there are only ten classes of sites, and we can number them as shown in Table I. The standard algorithm, as noted above, chooses among all sites (nearest neighbor pairs) with equal probability, determines that the selected site (pair) is of class i by scanning its neighbors, and finally decides whether to flip (interchange) by finding P, usually from a table of

TABLE I

Classifications of Spins in the Ten-Fold Way[a]

Class	Spin	Number of spin up nearest neighbors
1	Up	4
2	Up	3
3	Up	2
4	Up	1
5	Up	0
6	Down	4
7	Down	3
8	Down	2
9	Down	1
10	Down	0

[a] Lattice: square; Method: single spin flip; Periodic Boundary Conditions.

values P_i, and comparing it with a random fraction R. The n-fold way chooses among all sites (pairs) with a probability which is weighted in such a way that the probability of choosing a given site (pair) is proportional to its probability of flipping (interchanging). Thus once a site (pair) is selected, the flipping (interchanging) can be immediately performed.

The details of the n-fold way can be understood from the discussion that follows, a description of a test of the ten-fold way of Table I and the results of that test.

The Ten-Fold Way

We tested the ten-fold way (on a lattice of 80 $\times$ 80 spins) because many of the equilibrium properties of the (infinite) square Ising lattice, including the critical temperature T_c, are well known [10]. We describe the ten-fold way program in detail. Extensions to spin exchange models are obvious, but not trivial. Extensions to other lattices evolving via single spin flips are obvious and trivial.

The program was written for a CDC 6600 computer in Fortran Extended (FTN) language. An array LOC(6400) is partitioned into ten classes with moveable partitions. The value of an element of LOC specifies a spin's location in the 80 $\times$ 80 square. Its index and the partition keep track of the classes of the spins as follows. Let n_i be the number of spins in class i and let m_i be the number of spins whose class number is less than i. Then LOC($m_i + 1$) through LOC(m_{i+1}) contains the spins of class i (i.e., the partition is the m_i's and $n_i = m_{i+1} - m_i$). A second array LOOK(6400) is required in order to find the address in LOC and the class of a spin, given its position in the lattice. The position determines the index of an element of LOOK. The value of that element is equal to the address of the spin in LOC plus 2^{20} times the class of the spin. Thus the n-fold way requires an approximate doubling of core required for program variables (or tripling if one chooses to use two arrays, one for address in LOC and one for class, instead of packing the array LOOK).

Each spin flip requires the following calculations. First we calculate the ten numbers,

$$Q_i = \sum_{j=1}^{i} n_j P_j ; \qquad i = 1, 2,..., 10, \tag{4}$$

which change after each flip and therefore explicitly depend on time. Then we choose a random number R with uniform probability in the interval $[0, Q_{10})$. By letting the class of the selected spin be defined by i such that $Q_{i-1} \leqslant R < Q_i$ ($Q_0 \equiv 0$), we choose a spin in class i with probability $n_i P_i / Q_{10}$. We then choose a random integer l_i with uniform probability in the interval $[1, m_i]$ to find LCH, the particular spin to flip. Therefore the a priori probability of a given spin being chosen as LCH is P_i/Q_{10} which is proportional to P_i, its probability of flipping. The location in the lattice of LCH is stored in LOC ($m_i + l_i$).

To flip LCH, we rearrange LOC and its partitions in a relatively simple way, changing the class of LCH by ± 5 without changing the class of any other spins. But flipping LCH also changes the classes of its four nearest neighbors by ± 1. To find the neighbors' locations in LOC and their class (without checking against the partition of LOC, a time-consuming operation), we must look in LOOK. Once we find the neighbors in LOC, we change their classes by four more simple rearrangements of LOC. Whenever we rearrange LOC, we also make the corresponding changes in LOOK to preserve proper crossreferencing between the two arrays. The rearrangement of LOC can be made considerably simpler if we use a larger LOC array. We chose the size of LOC as we did in anticipation of memory limitations in three-dimensional simulations or in spin exchange simulations, both of which require a larger pair of arrays. (To have a reasonable crystal size in three dimensions, more lattice sites are needed. In spin exchange simulations, the n-fold way requires two arrays of pairs of neighbors, and there are twice as many pairs as spins in a square lattice, three times as many in a simple cubic.)

We increment the time variable t as follows. Q_{10} is the number of spins times the average probability that an attempt will produce a flip, given configuration $\mathscr{C}$. At each flip, the time is incremented by a stochastic variable Δt whose expectation value is proportional to Q_{10}^{-1}. Thus t is proportional to the number of attempts per site. We choose

$$\Delta t = -(\tau/Q_{10}) \ln R, \tag{5}$$

where R is a random fraction. This choice reflects properly the distribution of time intervals between flips, for a reasonable physical model. In that model, discussed in Appendix B, the cumulative time t is approximately proportional to real time.

Results

To discover circumstances under which the n-fold way is superior to the standard algorithm, we wrote a program for the standard algorithm which paralleled the ten-fold way program, using the same programming economies wherever possible. Using the same starting condition, we evolved both programs to steady state. We compared central processor time to produce all the flips needed to reach steady state in the two programs. We also calculated the ratio of times to produce a flip in steady state. The results are presented in Table II.

In zero magnetic field, the two programs perform about equally well at T_c, but as temperature decreases, the ten-fold way becomes markedly faster, being ten times as fast as the standard algorithm when $T/T_c = 0.588$. In nonzero field, the ten-fold way is even better. Almost certainly, previous authors [2–6] modified the standard algorithm using programming techniques which increased its speed without changing its basic structure. We could undoubtedly do the same to our ten-fold way program. Thus Table II should be regarded as simply illustrative.

581/17/1-2

TABLE II

Comparison of Two Algorithms: Standard and Ten-fold Way

T/T_c	$4J/kT$	$2H/kT$	Initial configuration	Approach to steady state: Calculation time (sec) Standard	Approach to steady state: Calculation time (sec) Ten-fold Way	Approach to steady state: Flips	Approach to steady state: Tries	Time ratio in steady state (standard/ten-fold)
0.588	3.0	0	Random	700	92	2.3×10^5	1.1×10^7	10
0.881	2.0	0	Random	1480	782	2.0×10^6	2.3×10^7	2.5
0.881	2.0	0.2	Random	19.0	13.0	3.0×10^4	3.0×10^5	3.6
0.881	2.0	0.2	All down	72.0	46.5	1.1×10^5	1.0×10^6	3.6
0.950	1.856	0	Random	553	415	1.0×10^6	8.6×10^6	1.4
1.000	1.763	0	Random	405	417	1.0×10^6	6.2×10^6	1.0
1.000	1.763	0	All up	16.5	12.3	3.0×10^4	2.6×10^5	1.1
1.000	1.763	0.1763	Random	25	21	5.0×10^4	3.9×10^5	1.8
1.000	1.763	0.1763	All up	2.56	1.75	2.5×10^3	3.6×10^4	2.0
1.000	1.763	0.1763	All down	49.3	41.4	1.0×10^5	7.3×10^5	2.1
1.100	1.602	0	Random	13.8	21.3	5.0×10^4	2.0×10^5	0.73
1.100	1.602	0.1602	Random	16.1	19.6	4.5×10^4	2.4×10^5	1.2

It illuminates the value of the novel organization of the n-fold way algorithm. From Table II, it is clear that even allowing for differences in programming technique, there are many situations in which the ten-fold way is definitely superior to the standard algorithm in terms of central processor time.

Extending the n-Fold Way

For ferromagnetic spin exchange studies, the n-fold way would certainly be a great improvement over the standard algorithm under most conditions. Since clustering of like spins occurs quickly, the standard algorithm would quickly waste a lot of time choosing like neighbors. Furthermore, even when an unlike pair is found, the energy change can be greater than in single spin-flip simulations (e.g., in the square lattice, there are six neighbor bonds that change in a spin exchange, but only four in a spin flip). Thus the probability of interchanging can be much less than the probability of flipping.

We expect the system to tend to a state in which the classes with the smallest probabilities of interchanging have the most pairs, especially at low temperatures. This expectation is confirmed by a previous study [7]. $P_{\min}$, the smallest probability of interchanging, therefore influences $\bar{P}$, the average probability of interchanging more than any other P_i does. Since $P_{\min}$ is smaller for interchanges than for flips, interchanges are, on the average, less likely than single spin flips at the same value of J/kT.

Based on the results presented in Table II and the above comments, we conclude that the n-fold way is a valuable alternate technique to the standard Monte Carlo algorithm for generating configurations of Ising spin systems. In many studies it may reduce computation time by an order of magnitude or more. It will be most valuable in studies involving the spin-exchange mechanism with ferromagnetic interactions. Its value in single spin-flip models, although clearly demonstrated, is limited since the greatest interest in those models is near $T = T_c$ where the standard technique appears to be as efficient as the more elaborate n-fold way.

It is interesting to contrast our algorithm with one proposed by Friedberg and Cameron [11] for the study of equilibrium Ising systems. Their method is a variant of the basic idea of Metropolis *et al.* [9] which generates the random spin flips in a way particularly well suited for modern digital computers. Our method was motivated primarily as a study of the time evolution of a system but it could be applied equally well at equilibrium. If so used it represents an entirely different approach which is or could be made complementary. Friedberg and Cameron stress carrying out the elementary operations in a very rapid way. We stress the need to emphasize sites likely to undergo transitions. But there is no reason why elements of both algorithms could not be combined.

APPENDIX A: AN INTERMEDIATE ALGORITHM

Previously we reported results of a Monte Carlo simulation of the time evolution of a two-dimensional binary alloy [7]. The algorithm used there was intermediate between the standard algorithm and the seven-fold way for spin exchanges on a ferromagnetic square Ising lattice. Using a redundant crossreferencing scheme (like the LOC/LOOK arrays), we selected an up spin (A atom) which had at least one down spin (B atom) neighbor; then we chose a neighbor at random. This made the spin exchange mechanism more like the single spin flip mechanism since each site chosen had a finite chance of being changed.

But even with this improvement, the algorithm still suffered from unlikely interchanges. Data from that study indicates that at $T/T_c = 0.588$, a system with equal concentrations of A and B atoms (equivalent to zero magnetization) required, on the average, about 40 choices of a pair of neighbors for each interchange once substantial clustering was evident. (In fact, that was nearly all of the time, for starting from a random initial state, this 40:1 ratio was established after only 3×10^4 interchanges in a study which ran to 2×10^6 interchanges.) For each interchange, there were, on the average, 13.8 AB pairs found, and for each AB pair found there were on the average 2.9 like pairs found. Thus under those circumstances, the more complicated seven-fold way would have saved considerable computation time.

APPENDIX B: COMPUTATION OF STOCHASTIC TIME VARIABLE

Suppose that our lattice system is immersed in a bath at fixed temperature T. Further suppose that this bath generates attempted flips randomly in space and time such that, on the average, there is one attempted flip per lattice site in time τ. We expect τ to depend on the temperature of the bath and the nature of the system, but only weakly on the state of the system. The probability of flipping a spin on a given random attempt is Q_{10}/N, where N is the number of lattice sites. Thus the probability of having a flip during the infinitesimal time interval dt is

$$p\,dt = (Q_{10}/\tau)\,dt. \tag{6}$$

We calculate the distribution of time intervals between flips by considering the probability that no flip occurs before time Δt has elapsed since the previous flip, $P(\Delta t)$.

The probability that no flip occurs before $\Delta t + dt$ has elapsed since the previous flip, $P(\Delta t + dt)$, must be less than $P(\Delta t)$ by dp, the probability that no flip occurs

before Δt, but one occurs between Δt and $\Delta t + dt$. But dp is clearly $P(\Delta t)\, p\, dt$; thus,

$$P(\Delta t + dt) = P(\Delta t) - P(\Delta t)(Q_{10}/\tau)\, dt. \tag{7}$$

We rewrite (7) as a differential equation

$$P'(\Delta t) = (-Q_{10}/\tau)\, P(\Delta t), \tag{8}$$

which has solution

$$P(\Delta t) = \exp(-Q_{10}\, \Delta t/\tau), \tag{9}$$

where we have used the obvious boundary condition that $P(0) = 1$.

We can now calculate the stochastic time interval between flips. We choose a random fraction R uniformly over the interval (0, 1). We then set $P(\Delta t) = R$ and solve for Δt:

$$\Delta t = -(\tau/Q_{10}) \ln R. \tag{10}$$

Since we assume that τ is nearly independent of the state of the system, the cumulative time t is approximately proportional to real time.

ACKNOWLEDGMENT

We are grateful to Dr. K. Binder for valuable comments.

REFERENCES

1. L. D. FOSDICK, Monte Carlo Computations on the Ising Lattice, *in* "Methods in Computational Physics," Vol. 1, (B. Alder, Ed.) Academic Press, New York, 1963; and references therein.
2. H. MÜLLER-KRUMBHAAR AND K. BINDER, *J. Stat. Phys.* **8** (1973), 1; and references therein.
3. K. BINDER AND E. STOLL, *Phys. Rev. Letters* **30** (1973), 47.
4. E. STOLL, K. BINDER, AND T. SCHNEIDER, *Phys. Rev. B* **8** (1973), 3266.
5. P. A. FLINN, *J. Stat. Phys.* **10** (1974), 89.
6. K. BINDER, *Z. Physik*, to be published.
7. A. B. BORTZ, M. H. KALOS, J. L. LEBOWITZ, AND M. A. ZENDEJAS, *Phys. Rev, B* **10** (1974), 535.
8. A. B. BORTZ, *J. Stat. Phys.*, to be published.
9. N. C. METROPOLIS, A. W. ROSENBLUTH, M. N. ROSENBLUTH, AND E. TELLER, *J. Chem. Phys.* **21** (1953), 1087.
10. L. ONSAGER, *Phys. Rev.* **65** (1944), 117.
11. R. FRIEDBERG AND J. E. CAMERON, *J. Chem. Phys.* **52** (1970), 6049.