

Fast Algorithm for the Simulation of Ising Models

H. J. Herrmann¹

Received May 13, 1986

Using a new microcanonical algorithm efficiently vectorized on a Cray XMP, we reach a simulation speed of 1.5 nsec per update of one spin, three times faster than the best previous method known to us. Data for the nonlinear relaxation with conserved energy are presented for the two-dimensional Ising model.

KEY WORDS: Ising model; simulation algorithm; critical dynamics.

1. INTRODUCTION

Fast simulations of Ising systems are of interest for the treatment of various problems, such as spin-glasses, nucleation, or random field systems. Moreover, they serve as a benchmark for new algorithms and computers. It has hence been a challenge to find faster algorithms for this purpose.

Before 1982 multi-spin-coding techniques on general-purpose computers^(1,2) were the fastest. Then special-purpose machines^(3,4) held the record with 30 M updates/sec (MHz). Later, faster simulations (219 MHz) were performed on the array processor DAP.⁽⁵⁾ In the meantime progress has also been made on general (vector) computers, typically reaching 100 MHz with the Metropolis algorithm.⁽⁶⁾

In this paper a new algorithm is implemented that sets a new record of 670 MHz. Our simulation is microcanonical and has strict energy conservation. It was proposed originally by Pomeau and Vichniac⁽⁷⁾ as an example for a cellular automaton. Since the algorithm is deterministic, random numbers are only required for making the initial configuration. Thus, 64

¹ Service de Physique Théorique, CEN-Saclay, 91191 Gif-sur-Yvette, France.

bits can simultaneously be updated in a loop that is automatically vectorized on the Cray.

In the next section we will present the algorithm and in the last section we discuss our results for the 2D Ising model.

2. ALGORITHM

The microcanonical algorithm described in Ref. 7 is the following: divide a square lattice (or hypercubic lattice in d dimensions) into two sublattices A and B and place a variable $\sigma_i^A(\sigma_i^B)$ having a value 0 or 1 on each site i of the sublattice $A(B)$. Now, at each time step, a whole sublattice is simultaneously updated: at even (odd) time steps one updates the sublattice $A(B)$ by⁽⁷⁾

$$\sigma_i^{A(B)}(t+1) = \sigma_i^{A(B)}(t) + A \left\{ \sum_{j=nn \text{ of } i} [1 - 2\sigma_j^{B(A)}(t)] \right\} [1 - 2\sigma_i^{A(B)}(t)] \quad (1a)$$

where the Boolean function $A(x)$ is given by $A(0) = 1$ and $A(x \neq 0) = 0$. In other words:

$$\begin{aligned} &\text{IF AND ONLY IF THE SPIN } i \text{ HAS} \\ &\text{AS MANY UP AS DOWN NEIGHBORS,} \\ &\text{IT IS FLIPPED} \end{aligned} \quad (1b)$$

but only on one sublattice at a time! (See Ref. 7.)

The above model describes the dynamics of an Ising model where the energy is conserved.^(7,8) One can also show that the basic axioms required for the transition rates of a Monte Carlo simulation⁽¹⁹⁾ (e.g., that equilibrium is a fixed point of the Markov chain) are fulfilled.

We have implemented the above algorithm on a square lattice on a Cray XMP in the following way: each bit of a 64-bit word corresponds to a spin; the linear dimension of the lattice is therefore a multiple of 128. Actually, the system is stored in four one-dimensional vectors corresponding to four sublattices. The system has periodic boundaries in both directions. In order to assure the periodic boundary conditions in up-down direction, a shadow line is attached at both the top and the bottom of the system.

The initial configuration is chosen at a given energy, e.g., by *randomly* setting a certain number of spins on one sublattice to be one and all other spins to be zero. The update of Eq. (1) is then implemented by using the Boolean function

$$f(\sigma_1, \sigma_2, \sigma_3, \sigma_4) = [(\sigma_1 \oplus \sigma_2) \wedge (\sigma_3 \oplus \sigma_4)] \vee [(\sigma_1 \oplus \sigma_3) \wedge (\sigma_2 \oplus \sigma_4)] \quad (2)$$

where \wedge , \vee , and \oplus are the logical functions “and,” “or,” and “exclusive or.” If one wants to update a certain spin at site i , one chooses as $\sigma_1, \dots, \sigma_4$ its four nearest neighbor spins (which certainly are on different sublattices) and changes the spin at site i if $f(\sigma_1, \dots, \sigma_4) = 1$; otherwise one keeps the spin as it was.

In our Fortran program the 64 spins of one word can consequently be updated *simultaneously* using only the bitwise logical functions AND, OR, and XOR. Spins in even (odd) lines are in vectors IA and IB (IC and ID) and spins in even (odd) columns are in vectors IB and IC (IA and ID). For instance, in the first line, consecutive spins in odd columns are placed in the first bit of ID(1), ..., ID(M), then in the second bit, and so on ($M = \text{size}/128$). For the updating of each of the four sublattice-vectors IA, IB, IC, and ID two separate loops are needed (that both automatically vectorize on the CRAY), one for exceptional words where one of the neighbors is not a bit in the same position as the bit of the site to be updated and one for all other words. In the first type of loop an additional circular left shift operation is needed. As an example, we give the instructions for the update of ID in a system $2N \times 2N$:

```

M = N/64
MM = M - 1
J = 1
DO 1 I = 1, N
  I1 = IA(J)
  I2 = IA(J + M)
  I3 = IC(J)
  I4 = SHIFT(IC(J + MM), 63)
  ID(J) = XOR(ID(J), OR(AND(XOR(I1, I2), XOR(I3, I4)),
    AND(XOR(I1, I3), XOR(I1, I4))))
1  J = J + M
  IF(M.EQ.1) GOTO 2
  DO 2 K = 2, M
    J = K
    DO 3 I = 1, N
      I1 = IA(J)
      I2 = IA(J + M)
      I3 = IC(J)
      I4 = IC(J - 1)
      ID(J) = XOR(ID(J), OR(AND(XOR(I1, I2), XOR(I3, I4)),
        AND(XOR(I1, I3), XOR(I2, I4))))
    3  J = J + M
  2  CONTINUE

```

The magnetization IM can be calculated by using the bit count function POPCNT of the 114 Cray Fortran Compiler:

```

NV = N*M
IM = 0
DO 4 I = 1, NV
IM = IM + POPCNT(IA(I)) + POPCNT(IB(I))
      + POPCNT(IC(I)) + POPCNT(ID(I))
4 CONTINUE

```

A complete listing of the program is available from the author on request.

By using the above program one obtains a speed of 1.48 nsec per update of one spin and 0.53 nsec per spin for the calculation of the magnetization for systems of length 1280. If the length is only 128, the speed decreases by about 20%. If vectorization is suppressed, the process is about five times slower.

3. RESULTS AND CONCLUSION

In order to test the algorithm, we calculated the magnetization of the Ising model on a square lattice close to the critical energy $E_c = -\sqrt{2}$ as shown in Fig. 1. The results are compared to the exact curve of an infinite system (see, e.g., Ref. 10). The temperature can be obtained from an exact relation to the energy.⁽¹⁰⁾ The critical point is well reproduced by the data (taking into account the rounding effects due to finite sizes).

In the ordered phase the system can lock into limit cycles (since it is completely deterministic) which correspond to metastable states. In these cases the magnetization does not reach its equilibrium value. The average value over many samples, however, corresponds to the expected exact value. The existence of these metastabilities, which is reminiscent of a spin-glass, might be explained as follows: at low temperatures there are separated clusters of spin one and the energy is then not only globally, but also locally conserved. For example, the cluster

$$\begin{array}{cccccc}
 0000 & 0000 & 0000 & 0000 & 0000 & 0000 \\
 0100 & 0110 & 0010 & 0010 & 0110 & 0100 \\
 0010 & \rightarrow & 0110 & \rightarrow & 0100 & \rightarrow & 0100 & \rightarrow & 0110 & \rightarrow & 0010 \\
 0000 & 0000 & 0000 & 0000 & 0000 & 0000
 \end{array} \quad (3)$$

has a limit cycle of five. These effects are, however, not important close to the critical temperature, where most simulations are made.

Relaxation times close to the critical temperature T_c are longer for our

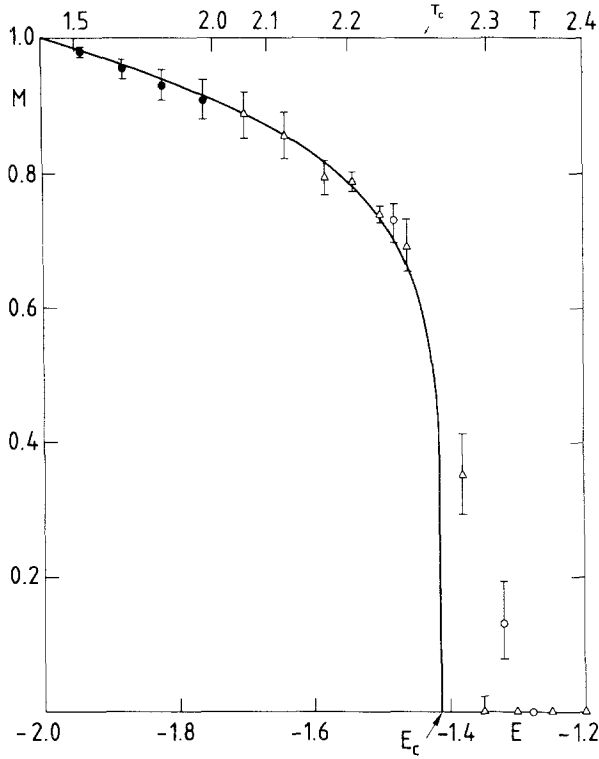


Fig. 1. Magnetization against energy and temperature for the Ising model on a square lattice. (—) The exact result. The length of the simulated systems is (○) 128, (△) 256, and (●) 1280. The statistical error bars are made over up to ten samples.

microcanonical algorithm than for Metropolis Monte Carlo [about 6000 updates per spin at $(T - T_c)/T_c = 0.01$ in order to reach equilibrium].

The study of the relaxation close to T_c is an interesting subject as such. Define the nonlinear relaxation time τ of the magnetization $M(t)$ for $T > T_c$ as

$$\tau = \int_0^{\infty} M(t) dt \quad (4)$$

Then one expects close to T_c critical slowing down⁽¹¹⁾:

$$\tau \propto (T - T_c)^{-z} \quad (5)$$

where z is called the nonlinear dynamical critical exponent. Equivalently one defines the linear dynamical exponent z_l for time correlations in

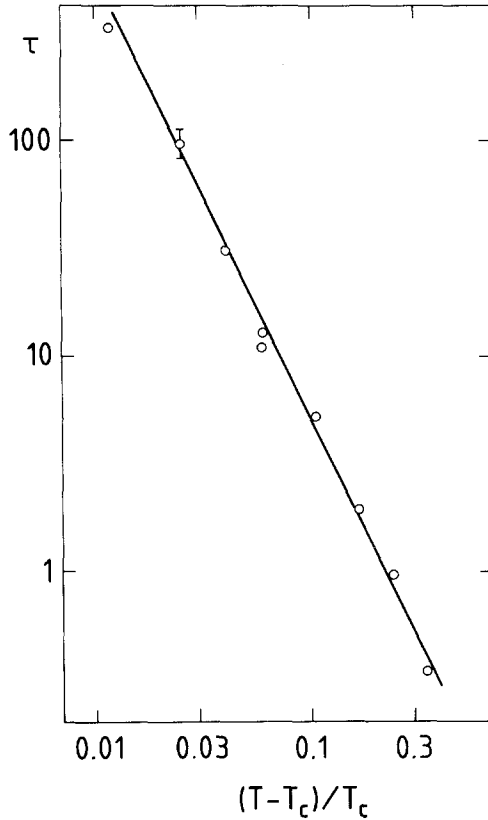


Fig. 2. Log-log plot of the nonlinear relaxation time against $(T - T_c)/T_c$.

equilibrium. For the case of energy conservation one expects for the linear dynamical exponent $z_l = 2 - \alpha/\nu$.⁽¹¹⁾ On the other hand, a relation between linear and nonlinear exponents has been proposed in Ref. 12: $z = z_l - \beta/\nu$. So one might expect $z = 2 - (\alpha + \beta)/\nu$. Figure 2 is a log-log plot of τ against $(T - T_c)/T_c$. The slope gives $z = 2.1 \pm 0.2$. This value is not inconsistent with the value $z = 1.875$ predicted by the theory.

In conclusion, we have described an algorithm that simulates microcanonically the Ising model and is about three times faster than the fastest simulation that exists presently. It is easily implementable within Fortran on a general-purpose computer. We have studied some microcanonical effects, such as the longer relaxation times and dynamical scaling with conserved energy. The algorithm can be easily applied to simulations of $+/-$ Ising spin-glasses and a little less efficiently to Potts

models. Compared to other microcanonical simulations, e.g., that of Ref. 13, our algorithm is at least ten times faster and has exact energy conservation, not approximate energy conservation.

ACKNOWLEDGMENT

I thank H. O. Carmesin, R. Lacaze, Y. Pomeau, D. Stauffer, and G. Vichniac for helpful and interesting discussions.

REFERENCES

1. R. Zorn, H. J. Herrmann, and C. Rebbi, *Comp. Phys. Comm.* **23**:337 (1981); C. Kalle and V. Winkelmann, *J. Stat. Phys.* **28**:639 (1982).
2. S. Wansleben, J. G. Zabolitzky, and C. Kalle, *J. Stat. Phys.* **37**:271 (1984).
3. R. B. Pearson, J. L. Richardson, and D. Toussaint, *J. Comput. Phys.* **51**:241 (1983).
4. J. H. Condon and A. T. Ogielski, *Rev. Sci. Instrum.* **56**:1691 (1985).
5. S. F. Reddaway, D. M. Scott, and K. A. Smith, *Comp. Phys. Comm.* **37**:351 (1985).
6. G. Bhanot, D. Duke, and R. Salvador, *J. Stat. Phys.*, in press; J. G. Zabolitzky, private communication (1986); H. Schumacher, Staatsexamensarbeit Köln (1985).
7. Y. Pomeau, *J. Phys. A* **17**:L415 (1984); G. Vichniac, *Physica D* **10**:96 (1984).
8. E. Goles and G. Y. Vichniac, preprint.
9. K. Binder, in *Phase Transitions and Critical Phenomena*, Vol. 5b, C. Domb and M. S. Green, eds. (Academic Press, New York, 1976), p. 1.
10. C. Domb, *Ad. Phys.* **9**:149 (1960).
11. B. I. Halperin and P. C. Hohenberg, *Rev. Mod. Phys.* **49**:435 (1977).
12. Z. Racz, *Phys. Rev. B* **13**:263 (1976); M. E. Fisher and Z. Racz, *Phys. Rev. B* **13**:5039 (1976).
13. M. Creutz, *Phys. Rev. Lett.* **50**:1411 (1983); M. Creutz, P. Mitra, and K. J.M. Moriarty, *Comp. Phys. Comm.* **33**:361 (1984).