A FAST ALGORITHM FOR INVESTIGATIONS ON THE THREE-DIMENSIONAL ISING MODEL

Michael CREUTZ

Department of Physics, Brookhaven National Laboratory, Upton, NY 11973, USA

P. MITRA

Institute for Computational Studies, Department of Mathematics, Statistics and Computing Science, Dalhousie University, Halifax, Nova Scotia B3H 4H8, Canada

and

K.J.M. MORIARTY *

Institute for Computational Studies at CSU, P.O. Box 1852, Fort Collins, CO 80522, USA

Received 10 April 1984; in revised form 8 May 1984

PROGRAM SUMMARY

Title of program: ISING

Catalogue number: ACCP

Program available from: CPC Program Library, Queen's University of Belfast, N. Ireland (see application form in this issue)

Computer: CDC CYBER 170-730 (dual processor); *Installation*: Dalhousie University Computer Center

Operating system: CDC Nos 2.1

Programming language used: FORTRAN-77 and CDC COMPASS

High speed storage required: 25 Kwords

Number of bits in a word: 60

Peripherals used: terminal, line printer

Number of lines in combined program and test deck: 550

Keywords: Ising model, phase transitions, critical exponents, correlation functions, magnetization, microcanonical methods

Nature of the physical problem
We wish to study the critical temperature and critical exponents of the three-dimensional Ising model.

Method of solution
A microcanonical method with demons [1] is used for the simulation of the three-dimensional Ising model. The updating procedure is essentially deterministic and uses only integer arithmetic. Correlations can be measured.

Restriction on the complexity of the program
The only restriction on the use of the program is the time needed to acquire reasonable statistics.

Typical running time
The execution time increases with the number of spin updates required. The program will carry out 860000 spin updates per second on the CDC CYBER 170-730 or 24000000 spin updates per second, including measuring the demon energy and lattice magnetization, on the CDC 7600. The test run output with a reduced lattice size took 4.8 s on the CDC CYBER 170–730.

Reference

[1] M. Creutz, Phys. Rev. Lett. 50 (1983) 1411.
G. Bhanot, M. Creutz and H. Neuberger, Nucl. Phys. B235 [FS11] (1984) 417.

* Permanent addresses: Institute for Computational Studies, Department of Mathematics, Statistics and Computing Science, Dalhousie University, Halifax, Nova Scotia B3H 4H8, Canada, and Department of Mathematics, Royal Holloway College, Englefield Green, Surrey TW20 0EX, UK.

0010-4655/84/$03.00 © Elsevier Science Publishers B.V.
(North-Holland Physics Publishing Division)

LONG WRITE-UP

1. Introduction

Numerical simulation of statistical systems has usually been carried out by stochastic methods, especially the "Metropolis" Monte Carlo method [1]. Of late, however, it has been realized that deterministic methods can be used instead [2,3]. The essential idea is the one underlying ergodic theory: the complexity inherent in a large system is expected to generate, in effect, random numbers without explicit use of pseudorandom number generators. However, instead of obtaining a sequence of states by integrating the equations of motion along a phase space trajectory, we use a modified microcanonical method [3]. An array of "demons" is let loose on the system. They interact sequentially with the microscopic units of the system and try to change their states. In general, a random choice of the final state is involved, but in cases like the Ising model where each spin has only two possible states, there is no randomness. The attempted changes are permitted if and only if energy conservation can be obeyed. The demons carry energy, and it is the sum of the energies of the interacting demon and the system that has to be conserved.

An algorithm for the treatment of the two-dimensional Ising model along these lines has been presented earlier [4]. Here we take up the three-dimensional Ising model.

2. Outline of the theory

The system consists of a set of spins located at the sites of a three-dimensional lattice. Each spin s_i can take the values ±1. The energy of a configuration is given by

$$E = \sum_{\text{bonds}(ij)} (1 - s_i s_j), \tag{1}$$

s_i and s_j denoting the spins on the two sides of the bond(ij). Physical quantities of interest are the bond average $\langle s_i s_j \rangle$ (= 1 − energy per bond) and more general correlation functions $\langle s_i s_{K(i)} \rangle$, where $K(i)$ denotes a site a specified distance away from the site i. Connected correlation functions are defined by subtracting the factorizing part: $\langle s_i s_{K(i)} \rangle - \langle s_i \rangle^2$. $\langle s_i \rangle$ measures the alignment of spins, i.e. the magnetization.

Since each spin has six nearest neighbors, the possible changes in the energy of the system consequent upon the flipping of a spin are 0, ±4, ±8, ±12. We allow each demon four energy states, with the values 0, 4, 8 or 12. For convenience, we remove a factor of 4 from the demon energy and consider $E_\text{d} = 0$, 1, 2 or 3. After a number of passages through the lattice, the demons become thermalized and their energy distribution becomes Boltzmannian. If the statistical temperature is β^{-1}, the average energy of the demon should be

$$\begin{aligned}4\langle E_\text{d}\rangle = {} & [4\exp(-4\beta) + 8\exp(-8\beta) \\ & + 12\exp(-12\beta)] \\ & \times [\exp(-\beta) + \exp(-4\beta) \\ & + \exp(-8\beta) + \exp(-12\beta)]^{-1},\end{aligned} \tag{2}$$

where the factor 4 was mentioned above. By measuring the demon energy and inverting eq. (2), one can determine the temperature.

In practice one has to work on a finite lattice, so that boundary conditions have to be specified for the definition of "nearest neighbours" at the boundaries. We use periodic boundary conditions in two directions and a shifted periodic condition in the third:

$$\begin{aligned} s(x + I_1, y, z) &= s(x, y + 1, z), \\ s(x, y + I_2, z) &= s(x, y, z), \\ s(x, y, z + I_3) &= s(x, y, z). \end{aligned} \tag{3}$$

Here the sites have been identified by Cartesian coordinates. I_1, I_2 and I_3 are the periodicities (sizes) in the three directions.

3. Code description

Since each spin can take only two values, efficient coding is achieved by using only one bit of a computer word to store its value. On a computer with 60-bit words, 60 spins can be stored in a single word. Our program envisages a lattice with $I_3 = 120$. The spins at odd and even sites on a single z-row are stored in separate words, which form two consecutive members of the list of words. The ordering of these words is as follows. The pair of words from one z-row are followed by the pair of words coming from the z-row obtained by moving one unit in the x-direction. The procedure is continued until the z–x plane is covered, then one moves one unit in the y-direction and starts from the lowest value of x. Because of the shifted boundary conditions, eq. (3), the determination of nearest neighbours is quite simple. It should be noted that the 60 spins in any single word do not interact among themselves. This is essential if the updating procedure is to work on the 60 bits at the same time. Each of the 60 spins is made to interact with a demon. One has therefore 60 demons. The variable E_d, which denotes one-fourth of the energy of a demon, takes the binary values 00, 01, 10 or 11. One word stores the first bit for the 60 demons, another the second.

The updating procedure, as explained above, is deterministic and uses only integer arithmetic. It is implemented in the subroutine MONTE written in COMPASS, the CDC assembler language. The demons hop along the list of spin-words, trying to flip the spins. The neighbours of each spin have to be located, and the proposed change in energy calculated. The demon energy then determines whether the change is to be permitted. As stated above, interactions of 60 spins with the 60 demons are handled simultaneously, and the 60 yes–no decisions are stored in one word. The appropriate spins are then flipped. For improving the equilibration, the bits in the words storing the demon energies are permuted. A fixed cyclic permutation is carried out after every iteration. In addition to this, a semi-random permutation is carried out in the main program between iterations. Because of this feature our simulation is not completely deterministic.

The energy of the system, as given by eq. (1), is measured by the subprogram ENERGY. It is approximately fixed by the parameter E used in initializing the lattice and the demons. The MONTE subroutine, which implements the simu-

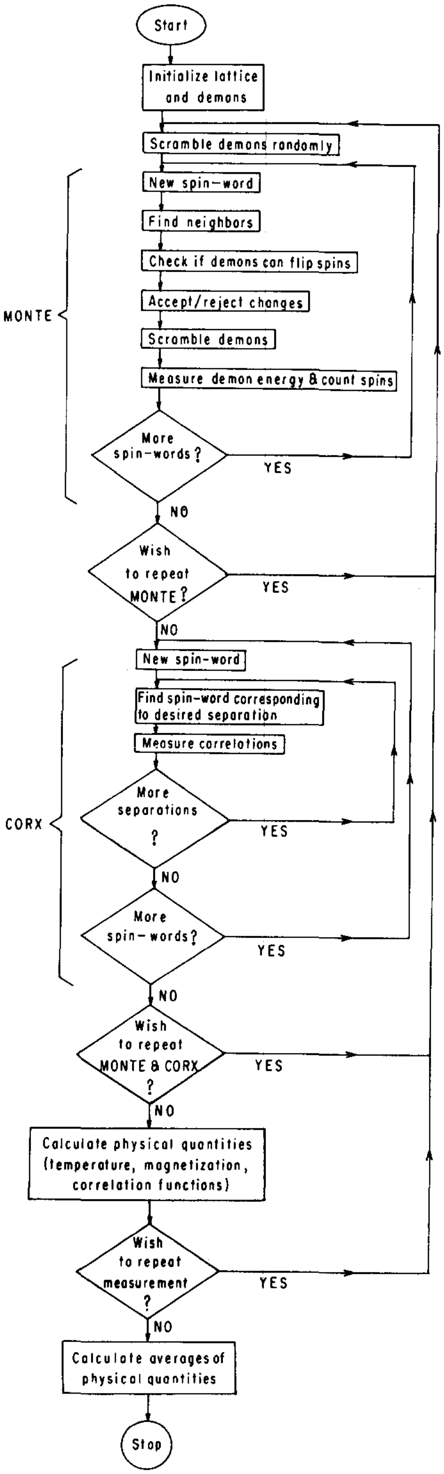
Start
Initialize lattice and demons
Scramble demons randomly
New spin-word
Find neighbors
Check if demons can flip spins
Accept/reject changes
Scramble demons
Measure demon energy & count spins
More spin-words ?
YES
NO
Wish to repeat MONTE ?
YES
NO
New spin-word
Find spin-word corresponding to desired separation
Measure correlations
More separations ?
YES
NO
More spin-words?
YES
NO
Wish to repeat MONTE & CORX ?
YES
NO
Calculate physical quantities (temperature, magnetization, correlation functions)
Wish to repeat measurement ?
YES
NO
Calculate averages of physical quantities
Stop
MONTE
CORX

Fig. 1. Flow chart for the Ising model program.

lation, also measures the demon energy and the magnetization. A subprogram BETA is used to compute the temperature from the demon energy. Correlation functions for spins separated in x and y directions are measured by a COMPASS subroutine CORX. Another subroutine, CORZ, can be used for the z-direction.

The flow chart of the program is shown in fig. 1.

4. Conclusions

As indicated above, our program assumes $l_3 = 120$. l_1 and l_2 are arbitrary but not divisible by 11 for technical reasons in subroutine MONTE. A $128 \times 128 \times 120$ lattice is the intended size, though for trial runs we used an $8 \times 8 \times 120$ version. The algorithm described in this paper was implemented for this size and sample results are shown in our test run output. If correlation functions are not measured our program handles about 8.6×10^5 spins per second on the CDC CYBER 170-730. If the CORX subroutine is called once for every 10 sweeps, the effective rate goes down to 8.3×10^5 spins per second. These rates are a factor of 30–40 lower than those achieved in machines specially built for the Ising model [5]. Considering the flexibility of our program as contrasted with the machine-dependence of those approaches, we tend to regard this as satisfactory. In fact, on a faster machine, e.g. the CDC 7600, speeds comparable to the Santa Barbara Ising model machine (25 Mspin updates per second with no measurements) are achieved using the program described in the text.

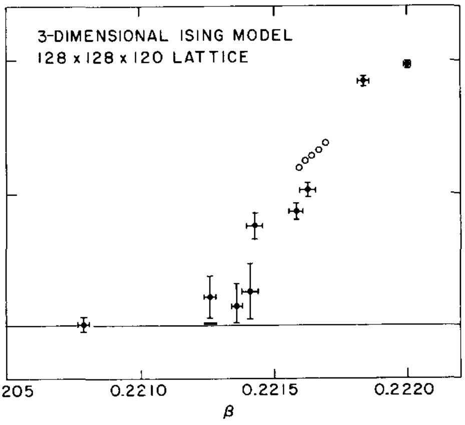

Fig. 2. The magnetization in the three-dimensional Ising model ($128 \times 128 \times 120$ lattice) near the critical inverse temperature. The open circles represent the results of the Santa Barbara Ising model machine [5] using a 64^3 lattice.

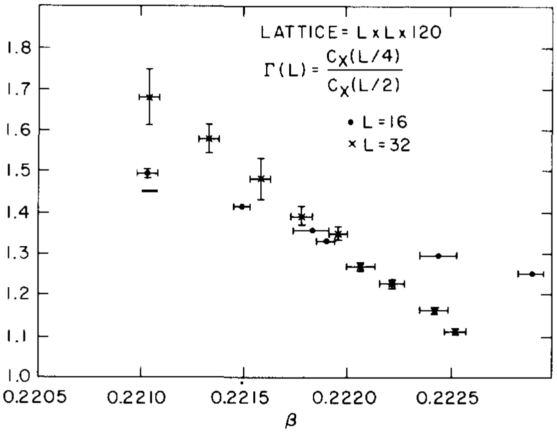

Fig. 3. The ratio of correlation functions for separations $L/2$ and $L/4$ for an $L \times L \times 120$ lattice.

We are now in the process of migrating this code to a vector processor, the 2 Mword 2 pipe CDC CYBER 205 at Colorado State University. A recent attempt [6] at implementing the Metropolis algorithm for the 3-dimensional Ising model on the University of Bochum CDC CYBER 205 resulted in a code capable of 10.2 Mspin updates per second. By the use of the microcanonical method [3,4] described in this paper, we expect to achieve absolute speeds of hundreds of million spin updates per second. This work [7] will be reported shortly. A recent calculation [8] using the ICL DAP parallel processor achieved a speed of 6Mspin updates per second.

For the interest of the reader, we include, in fig. 2, some preliminary data on the magnetization of the three-dimensional Ising model on a $128 \times 128 \times 120$ lattice. These results were obtained on a CDC 7600. Also shown in fig. 2 are some results of the Santa Barbara Ising model machine [5] on a 64^3 lattice. By speeding up our program considerably by working on the CDC CYBER 205, we hope to reduce our error bars, in fig. 2, to the size of a data point. In fact, we wish to concentrate most of our efforts on improving the preliminary data shown in fig. 3 on the ratios of correlations of spins C_x near the critical inverse temperature. The separations considered are $L/2$ and $L/4$ for an $L \times L \times 120$ lattice. The data shown in figs. 2 and 3 are the result of 10000 iterations per data point with the first 1000 iterations thrown away and the averages taken over the remaining 9000 iterations.

Acknowledgements

One of the authors (P.M.) wishes to thank D.D. Betts and the Natural Sciences and Engineering Research Council of Canada for financial support (NSERC grant number: A4889). Another author (K.J.M.M.) would like to thank Mr. R.T. French and the Idaho National Engineering Laboratory for the award of a grant (Fellowship number: MN-064-83) which made his visit to Fort Collins, Colorado possible. We would like to thank Dalhousie University Computer Center for the granting of time on their CDC CYBER 170–730 where this calculation was undertaken. This research was also carried out in part under the auspices of the US Department of Energy under contract No. DE-AC02-76CH00016.

References

[1] Monte Carlo Methods in Statistical Physics, ed. K. Binder (Springer-Verlag, Berlin, 1979).

[2] D.J.E. Callaway and A. Rahman, Phys. Rev. Lett. 49 (1982) 613; Phys. Rev. D28 (1983) 1506.

[3] M. Creutz, Phys. Rev. Lett. 50 (1983) 411.

[4] G. Bhanot, M. Creutz and H. Neuberger, Nucl. Phys. B235 [FS] (1984) 417.

[5] R.B. Pearson, J.L. Richardson and D. Toussaint, J. Comput. Phys. 51 (1983) 241.
M.N. Barber, R.B. Pearson, D. Toussaint and J.L. Richardson, Santa Barbara report NSF-ITP-83-144 (1983).

[6] S. Wansleben and J.G. Zabolitzky, Monte Carlo Simulation of Ising Model by Multi-Spin Coding on a Vector Computer, University of Cologne report (November 1983).

[7] D. Barkai, M. Creutz, P. Mitra and K.J.M. Moriarty, to be published.

[8] D.J. Wallace, University of Edinburgh preprint no. 83/260 (1983).
G.S. Pawley, R.H. Swendsen, D.J. Wallace and K.G. Wilson, Phys. Rev. B29 (1984) 4030.

TEST RUN OUTPUT

```
LATTICE SIXE  8 X 8 X 120

ELATT= .6472222222222 EDEM= .007638888888889
ETOT= .6548611111111

AVERAGE OVER 10 ITERATIONS,EACH WITH 10 SWEEPS
RUNNING AT .8888888888889 MFLIPS
BETA= .2210931883207 SISJ= .3512187771267
MAGNETIZATION= -.023046875

AVERAGE OVER 10 ITERATIONS,EACH WITH 10 SWEEPS
RUNNING AT .8347826086957 MFLIPS
BETA= .2227851128925 SISJ= .3511692843967
MAGNETIZATION= .0007682291666669
CORR(IX/4)= .1964583333333
COOR(IX/2)= .1359375
CORR(IX/2,IY/2)= .1196875

AVERAGE OVER 10 ITERATIONS,EACH WITH 10 SWEEPS
RUNNING AT .8411829134721 MFLIPS
BETA= .2224260846221 SISJ= .3511797553168
MAGNETIZATION= -.05866406249999
CORR(IX/4)= .1933854166667
CORR(IX/2)= .1321354166667
CORR(IX/2,IY/2)= .1185416666667

AVERAGE OVER 10 ITERATIONS,EACH WITH 10 SWEEPS
RUNNING AT .8448844884489 MFLIPS
BETA= .2211537065034 SISJ= .3512170003255
MAGNETIZATION= .1541848958333
CORR(IX/4)= .1904166666667
CORR(IX/2)= .1365104166667
CORR(IX/2,IY/2)= .13046875

AVERAGE OVER 10 ITERATIONS,EACH WITH 10 SWEEPS
RUNNING AT .8258064516128 MFLIPS
BETA= .221008213459 SISJ= .3512212727865
MAGNETIZATION= .2303828125
CORR(IX/4)= .1938020833333
CORR(IX/2)= .143125
CORR(IX/2,IY/2)= .1150520833333

*** AVERAGES AFTER DISCARDING FIRST BATCH ***
AV. BETA= .2218432793693 +/- .0004471724275268
AV. MAG.= .08166796875 +/- .06684101555469
AV. CORR(IX/4)= .193515625 +/- .001237002010018
AV. CORR(IX/2)= .1369270833333 +/- .002282672533389
AV. CORR(IX/2,IY/2)= .1209375 +/- .003326476780617
CORR(IX/4)/CORR(IX/2)= 1.414428370159 +/- .02480551229889
CORR(IX/2)/CORR(IX/2,IY/2)= 1.135188574795 +/- .04098916185012
```