Dynamical properties of a new fast algorithm for the simulation of the Ising model

1988 J. Phys. A: Math. Gen. 21 L315

(http://iopscience.iop.org/0305-4470/21/5/009)

J. Phys. A: Math. Gen. 21 (1988) L315-L320. Printed in the UK

LETTER TO THE EDITOR

Dynamical properties of a new fast algorithm for the simulation of the Ising model

Raúl Toral†

Physics Department, University of Edinburgh, King's Buildings, Edinburgh EH9 3JZ, UK

Received 3 November 1987

Abstract. A new algorithm recently proposed by Pomeau and Herrmann for the microcanonical simulation of the Ising model is very efficiently implemented on a DAP computer. Its dynamical properties are studied and compared with those of the standard Metropolis algorithm.

Numerical simulations are now, more than ever before, a valuable tool for studying the properties of mathematical models in physics (for reviews see Binder (1979, 1984)). As computers have become more and more powerful, the kind of problems being solved with their help have become more and more difficult and time consuming, demanding in turn more powerful algorithms. In many aspects of today's numerical studies of phase transitions we are interested in critical properties. In particular, one of the most important methods for obtaining critical exponents in lattice models is to use the theory of finite-size scaling (see Barber 1983). This method involves computing magnitudes at, or very near to, the critical point of the infinite system (indeed the only one for which a critical point exists) for systems with different finite sizes and extrapolating the results to the infinite system. Due to the very large nature of the phase space available even for very small systems, Monte Carlo techniques are used to obtain approximate values. This is a formidable task. It is well known that when a critical point is approached there is a dramatic increase in fluctuations (implying that more measurements are needed in order to get a given accuracy) and a critical slowing down (implying that more updates are needed in order to get statistically independent configurations). Very fast algorithms to do the simulations are required to overcome these problems (Goodman and Sokal (1986), Swendsen and Wang (1987); see Sokal (1988) for a recent review). The simulations are usually carried in the canonical ensemble where the Metropolis algorithm is used. Of course, it is not compulsory to use the canonical ensemble. The equivalence of the different collectivities in the thermodynamical limit makes it possible to extract useful information about the equilibrium properties from other ensembles (Creutz 1983).

A deterministic algorithm has been proposed by Pomeau (1984) as an example of a cellular automaton. It has been recently claimed by Herrmann (1986) that the algorithm is relevant in the simulation of the microcanonical ensemble and could offer an alternative to the more conventional Metropolis algorithm with a spectacular gain

† Present address: Department of Physics, Temple University, Barton Hall, Philadelphia, Pennyslvania 19122, USA.

0305-4470/88/050315+06$02.50

in speed. The algorithm goes as follows. Consider the Ising model with nearest-neighbour interactions in a regular lattice in d dimensions, then divide the lattice in two sublattices such that all the neighbours of the spins in one of the sublattices belong to the other sublattice. Update all the spins in one of the sublattices at once. A particular spin is flipped if and only if it has the same number of up and down neighbours. This is equivalent to saying that only flips that conserve the energy are accepted. Since this dynamics is deterministic, no random numbers are needed for the update (random numbers are only needed for the initialisation of the system). Moreover, it can be easily implemented by the use of only logical operations. Let σ_i denote a logical variable taking the values 0, 1 representing the spin variable S_i (taking the values $+1$, -1) in site i of a square lattice. If σ_{i_1}, σ_{i_2}, σ_{i_3} and σ_{i_4} are the four neighbours of σ_i, the above rule for updating σ_i can be explicitly implemented by using

$$\sigma_i' = \sigma_i \oplus \{[(\sigma_{i_1} \oplus \sigma_{i_2}) \wedge (\sigma_{i_3} \oplus \sigma_{i_4})] \vee [(\sigma_{i_1} \oplus \sigma_{i_3}) \wedge (\sigma_{i_2} \oplus \sigma_{i_4})]\} \tag{1}$$

(where $\wedge$, $\vee$ and $\oplus$ are the logical functions 'and', 'or' and 'exclusive or', and σ_i' is the new value of the spin variable at site i). This expression can be easily implemented using the (almost) standard Fortran bitwise logical operations, and has been timed as 670 million spin updates per second on a CRAY XMP (Herrmann 1986), which is about three times faster than any previous canonical simulation (Reddaway *et al* (1985), obtained using a single instruction multiple data DAP computer). If DAP computers have proved so successful in the simulation of the Metropolis algorithm, it is then natural to program this new microcanonical algorithm on the DAP. However, simply implementing this program in DAPFORTRAN (a parallel extension of FORTRAN) is not very efficient because the compiler has to translate the $\vee$ operation into the machine code set of instructions (known as APAL) which does not possess it and even the $\wedge$ and $\oplus$ operations require some additional overheads in APAL. The most efficient use of the DAP is obtained by programming the algorithm using three bits sums which is the primary instruction in APAL. It is easy to write (1) using only three bits sums. If L_1, L_2, L_3 are three logical type variables, let us denote by $Q(L_1, L_2, L_3)$ and $C(L_1, L_2, L_3)$ respectively the least significant bit and the most significant bit of the sum $L_1 + L_2 + L_3$. An algorithm replacing equation (1) is

$$\sigma_i' = Q[\sigma_i, Q(\sigma_{i_1}, \sigma_{i_2}, \sigma_{i_3}), C(C(\sigma_{i_1}, \sigma_{i_2}, \sigma_{i_3}), Q(\sigma_{i_1}, \sigma_{i_2}, \sigma_{i_3}), \overline{\sigma_{i_4}})] \tag{2}$$

($\bar{\sigma}$ denotes logical 'not'). This can be programmed using only three machine code instructions (plus the corresponding overheads). Another advantage of using the DAP is that there is a very efficient and natural mapping of a square system onto the DAP memory. The DAP memory consists of 'planes' of 64×64 bits such that every bit is connected to its four neighbours. To simulate, say, a 256×256 system we use 16 of these logical planes arranged in such a way that the number of operations necessary to get the neighbours of a given spin is minimised. Periodic boundary conditions are obtained 'for free' in the hardware. Equation (2) can then be completely programmed using about seven one-cycle machine code instructions; the exact number depends on the size of the system. The bigger the system, the smaller the number of instructions, up to a limit of a system size of 1024×1024 beyond which (and due to the specific way of addressing memory in the APAL set of instructions) one loses efficiency. This is to be compared with the 22 FORTRAN statements necessary to implement (1) (see Herrmann (1986) for a listing of the code). The maximum speed obtained on the DAP is then 2.2×10^9 spin updates per second for the 1024×1024 system and 1.6×10^9 spin updates per second for systems 128×128 and smaller.

Let me discuss now some of the dynamical properties of this new algorithm. The first thing is clear is that it is not ergodic (this is clearly admitted in the papers by Pomeau and Herrmann). There are certain configurations that repeat themselves after a limited small number of updates (see Hermann *et al* 1987). The point is whether for simulations of large enough systems, with sufficiently random initial configurations and near the critical point, the non-ergodicity is irrelevant, i.e. the part of the phase space that is not sampled by the algorithm is irrelevant. To study these and other dynamical properties, I have monitored the magnetisation

$$M(t) = \sum_{i=1}^{N} S_i(t). \tag{3}$$

Figures 1 and 2 show for the critical point the time evolution of the magnetisation for different sizes of the system for both this microcanonical and the standard parallel implementation of the Metropolis algorithm for the canonical simulation. The choice of the initial configuration for the microcanonical simulation deserves a special comment. In the microcanonical ensemble, the energy, defined by the standard relation

$$E = -J \sum_{\mathrm{NN}} S_i S_j \tag{4}$$

is a constant. I have then generated the initial configuration as follows: once a particular value of the energy, E, has been chosen I generate a completely random configuration with a density ρ such that the expected value of the energy $\langle E \rangle = JN(8\rho(1-\rho)-2)$ (N is the number of spins) is equal to E. A final tuning is obtained by randomly flipping spins until the energy has the desired value E. The qualitative system behaviour, however, seems to depend on the choice of the initial configuration as is clear from figures 1(b, c) and 1(e, f).

Another important thing to notice is that even for system sizes as large as 32×32 there are periodicities due to the non-ergodical nature of the algorithm. Only for system sizes beyond 64×64 are there no evident signs of periodicity and it would seem then that the algorithm can be safely used to compute magnitudes in the system. This appears to be true. However, it is obvious from figures 1 and 2 that even though the magnetisation fluctuations have a similar appearance in both algorithms for systems of 128×128 spins, the timescales are very different and the microcanonical algorithm seems to need an order of magnitude more updates than the canonical algorithm in order to get the same qualitative results. The conclusion is that the microcanonical algorithm has a much larger relaxation time than the canonical algorithm.

This statement, which is qualitatively observed in figures 1 and 2, can be put in a more quantitative basis by studying the behaviour of the spin–spin correlation function:

$$\Gamma(t) = \left\langle \sum_{i=1}^{N} S_i(0) S_i(t) \right\rangle. \tag{5}$$

Figures 3 and 4 show the evolution of this magnitude for the microcanonical simulation and system sizes 64×64 and 128×128 respectively. It is remarkable that the timescales for the decay of this function are comparable for both systems, indicating that the non-ergodicity of the algorithm is still causing trouble in the 64×64 system. The 'decorrelation time', τ (or time necessary for the system to forget previous configurations), can be obtained by fitting $\Gamma(t)$ to an exponential decay

$$\Gamma(t) = \Gamma_0 \, \mathrm{e}^{-t/\tau}. \tag{6}$$

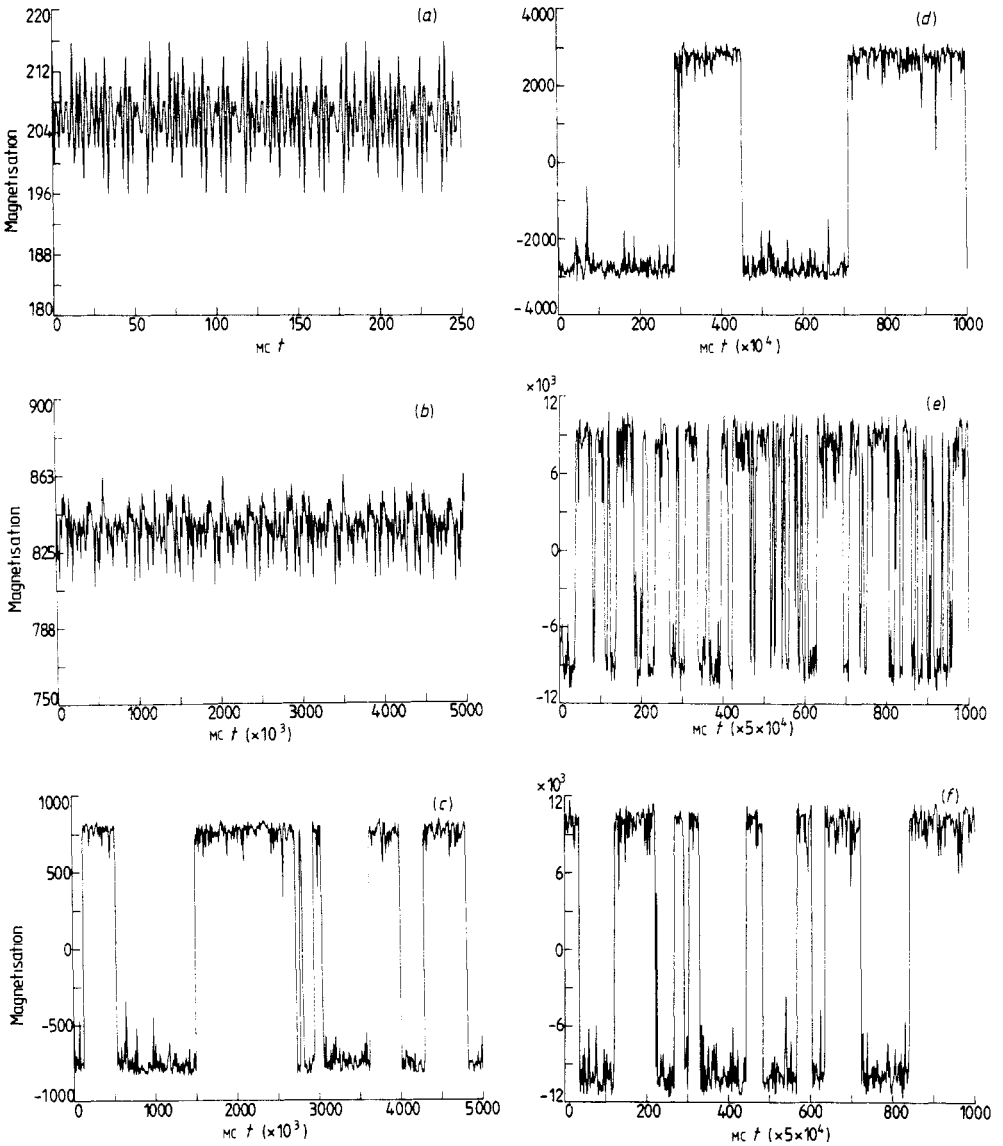

Figure 1. Time evolution of the magnetisation in the microcanonical algorithm described in the text for different values of the system size: (*a*) $L=16$; (*b*), (*c*) $L=32$; (*d*) $L=64$; (*e*), (*f*) $L=128$. (*b*) and (*c*) differ only in the seed used in the random number to generate the initial configuration. This is also the case for (*e*) and (*f*).

A least-squares fit to the data in figure 4 yields $\tau_{\text{microcanonical}} = 3.3 \times 10^5$ while a similar analysis for the standard Metropolis algorithm for the simulation of the canonical ensemble yields $\tau_{\text{canonical}} = 2.6 \times 10^4$, both for a 128×128 system, confirming again the fact that the microcanonical simulation is an order of magnitude slower when generating an independent configuration.

In conclusion, it appears that this new algorithm for the simulation of the Ising model has to be taken with great care. There are clear problems with ergodicity for systems near the critical point even for system sizes as large as 64×64. For larger

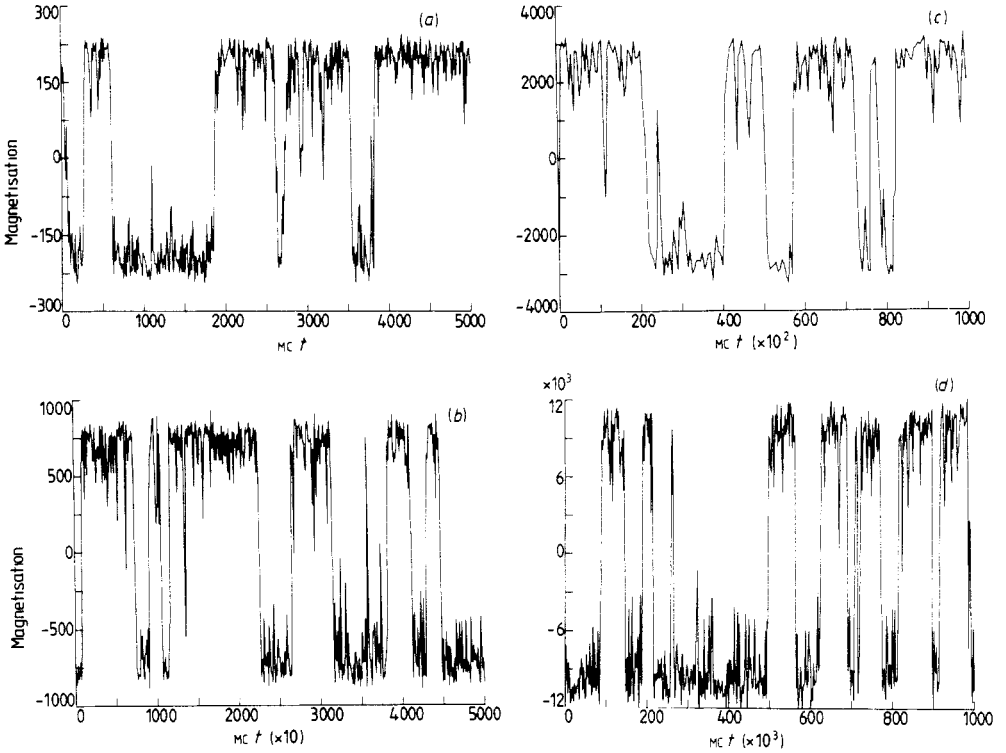

Figure 2. Time evolution of the magnetisation in the standard parallel version of the Metropolis algorithm for different system sizes: (*a*) $L=16$; (*b*) $L=32$; (*c*) $L=64$; (*d*) $L=128$.

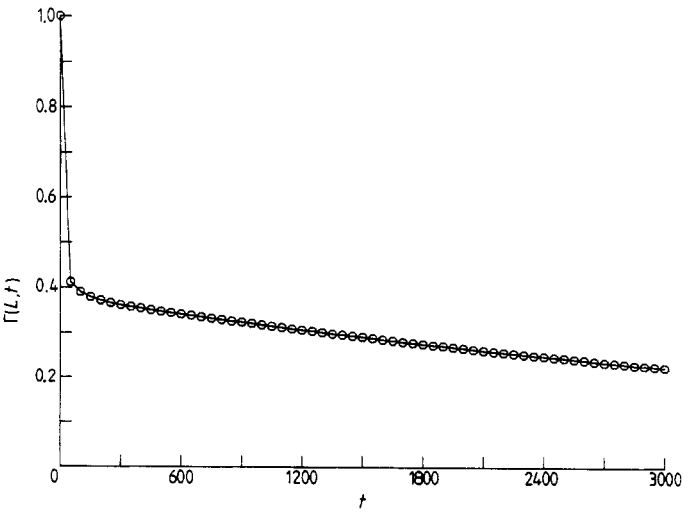

Figure 3. Time evolution of the spin-spin correlation function for the microcanonical algorithm for a square system of 64×64 spins. The error bars arise from the statistics of 2000 independent runs with different initial conditions.

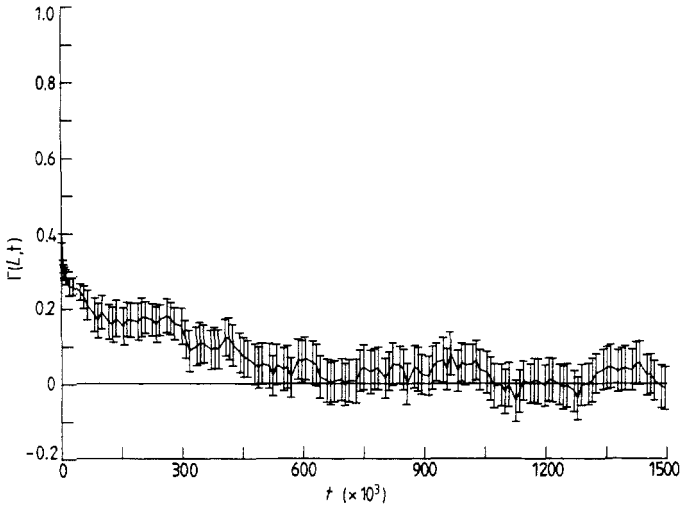

Figure 4. As figure 3 but for the 128×128 system, averaged over 500 independent runs.

systems, where we have not found any anomalous results due to non-ergodicity, the number of configurations necessary to generate an independent one is an order of magnitude larger than necessary in the conventional Metropolis algorithm. Of course, when considered as a cellular automaton model, the algorithm is interesting on its own.

I wish to thank Gyan Bhanot for interesting discussions and S Reddaway for a critical reading of an early version of the manuscript. Support by the 'Programa Cientifico de la OTAN en España' is also acknowledged.

References

Barber M 1983 *Phase Transitions and Critical Phenomena* vol 8, ed C Domb and J L Lebowitz (New York: Academic) p 145

Binder K (ed) 1979 *Monte Carlo Methods in Statistical Physics* (Berlin: Springer)

—— 1984 *Applications of the Monte Carlo Method in Statistical Physics* (Berlin: Springer)

Creutz M 1983 *Phys. Rev. Lett.* **50** 1411

Goodman J and Sokal A 1986 *Phys. Rev. Lett.* **56** 1015

Herrmann H 1986 *J. Stat. Phys.* **45** 145

Herrmann H, Carmesin H and Stauffer D 1987 *J. Phys. A: Math. Gen.* **20** 4939

Pomeau Y 1984 *J. Phys. A: Math. Gen.* **17** L415

Reddaway S, Scott D and Smith K 1985 *Comput. Phys. Commun.* **37** 351

Sokal A 1988 *Proc. 8th Int. Congr. of Mathematical Physics, Marseille* (Singapore: World Scientific) to be published

Swendsen R and Wang J 1987 *Phys. Rev. Lett.* **58** 86