Cluster algorithm special purpose computer for the three-dimensional Ising model

A. L. Talapov[1] and L. N. Shchur

Landau Institute for Theoretical Physics, Russian Academy of Sciences, GSP-1 117940, Moscow, Russia

H. W. J. Blöte

Faculty of Applied Physics, Delft University of Technology, 2628 CJ Delft, The Netherlands

(Submitted 26 June 1995)
Pis'ma Zh. Éksp. Teor. Fiz. **62**, No. 2 157–164 (25 July 1995)

We describe the ideas employed in the first special-purpose Wolff cluster algorithm computer able to simulate three-dimensional (3D) Ising models. The computer is fast enough to generate accurate data for lattices containing more than 16 million spins, at and near the Ising critical point. We have used this computer to obtain test results for the 2D and 3D Ising models at criticality. These are in an excellent agreement with exact results and with independent simulations in software.

During the last decades, the Ising model has acquired a reputation as a breeding ground for the development of new approaches to the physics of phase transitions.

The exact solution of the 2D model, first given by Onsager,[1] has been simplified by a number of authors. Now it is clear that the 2D Ising model is just a system of free fermions. Nevertheless, even in the 2D case intriguing problems still exist. For example, we mention the influence of impurities on critical behavior.[2]

In the 3D case no exact solution is available. Theoretical attempts to solve the model have gone on for decades, and many interesting methods have been developed along the way, but ultimate success has not yet been achieved.

In the absence of an exact theory, approximate results for the 3D Ising universality class have been obtained by series expansion[3,4] and numerical renormalization-group[5,6] methods. The accuracies of these analyses exceed those of typical experiments on systems believed to be in the Ising universality class.

Reliable data for the 3D Ising model can be obtained by computer simulations. Considerable efforts have taken place in this direction (see, e.g., Ref. 7–9, and the references cited therein), but many questions still remain.

Due to the random nature of the Monte Carlo approach, the statistical errors decrease in inverse proportion to the square root of the simulation time. Therefore, one needs a lot of computer time to obtain accurate results. This time becomes especially long in the most interesting region: the phase transition, where the fluctuations are huge, and the relaxation time tends to infinity.

Fortunately, critical slowing down can be suppressed by using a *cluster* algorithm. It is clear that near the critical point, where the correlation length tends to infinity, it would be preferable to flip whole clusters of spins instead of independent spins. Swendsen and Wang[10] discovered the proper way to build and flip such clusters. Indeed, simulations confirm that the critical slowing down reduces drastically when cluster-flip algorithms are used.

The Swendsen–Wang method was modified by Wolff.[11] The Wolff algorithm is even more efficient. Moreover, it is so simple that it can relatively easily be implemented in hardware.

The first cluster-algorithm special-purpose computer (SPC)[12,13] was specifically built to investigate the 2D Ising model with bond randomness.[14,15] It performs very efficiently, in particular near the critical temperature T_c. The cost of the electronic components of this SPC was less than that of a typical personal computer, whereas its speed compares favorably with that of a large mainframe computer.

Inspired by the success of the 2D SPC, we decided to follow the same approach for the 3D Ising model. The new 3D SPC has in common with the 2D SPC the property that it implements the Wolff algorithm in hardware. However, in other respects it is very different. The main changes include new very efficient organization of the spin memory, a new structure of the random number generator (due to A. Compagner[16]), and a much more efficient storage method for the cluster spin addresses. These modifications have permitted an increase of the number of lattice spins from 2^{20} in the 2D SPC to 2^{24} in the new 3D SPC.

The speed of both cluster SPC's is completely determined by the memory cycle times and thus does not depend on the speed of any microprocessors. Both the 2D and the 3D SPC spend three spin-memory cycle times t_{cy} per cluster spin. In the 3D SPC we used 4-megabit dynamic memories with $t_{\mathrm{cy}}=125$ ns; thus it requires only 375 ns per cluster spin. Our results, shown below in Table II demonstrate that this speed enables the generation of accurate data, even in the most difficult case of the largest 3D lattice, containing 16,777,216 spins, at the critical point.

The 3D SPC structure, shown in Fig. 1, is largely dictated by the Wolff algorithm used. The Wolff cluster formation process is determined by the following procedure.

First, a random site of the lattice is selected. This determines the first node: the central spin on this site and its 6 nearest neighbors. By definition, this central spin is the first member of the cluster. Then, its neighbors are 'processed,' i.e., it is decided if they are also included in the cluster. Each neighbor is processed independently.

To be included in the Wolff cluster, the neighbor is required to be parallel to the first spin (before it is flipped). But this condition is not sufficient. An additional requirement is that the bond connecting both neighbors is 'active'; bonds between interacting neighbors are active with a probability P defined by

$$P=1-e^{-2\beta J}, \tag{1}$$

where $\beta=1/T$, T is the temperature, and J is the coupling constant, which determines the interaction energy $E=-JSS'$ of two neighbor spins S and S'. When a spin is determined to be a cluster member, it is flipped immediately. In addition, its address is written in an

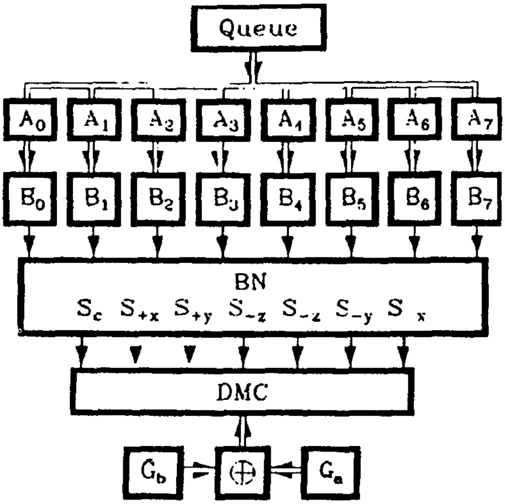

FIG. 1. Block diagram of the SPC.

auxiliary memory. This auxiliary memory contains a list of addresses of spins whose neighbors are waiting to be processed. Let us explain why this is necessary.

After processing all nearest neighbors of the first cluster spin, a new node cycle is started. That means that another cluster spin is chosen as the new central spin, its neighbors are processed, and for those that become members of the cluster, the addresses are written in the auxiliary memory. Then, another node cycle is started, and so on.

At the start of a new node cycle, the auxiliary memory supplies the new central spin address. However, there are different ways to choose this address.

In the first cluster SPC[12] the *stack* strategy was implemented. Reading and writing takes place on top of the stack. Thus an address read from the stack is always the last one that has been written. This strategy is simple, and requires only one stack pointer. But the required number of memory locations in the stack may exceed half the number of spins in the system.

An alternative is offered by the *queue* strategy, which turns out to be much more economical in terms of memory requirements. In this case the reading operations of the spin addresses take place in the same order as the writing operations. One might say that writing takes place on top of the queue, while reading takes place from the bottom. Thus, in the queue memory, we need both a read pointer and a write pointer, which are increased by 1 after each read or write operation, respectively.

At the start of the cluster, both pointers are equal to each other. During the cluster-building process, the write pointer is always larger than the read pointer. The cluster is finished when the read pointer becomes equal to the write pointer—all cluster spins have acted as the central spin of a node.

Preliminary simulations by Heringa[17] have shown the largest difference $q_{\max}$ between the write and read pointers satisfies.

$$q_{\max} \lesssim \begin{cases} 3L & \text{in 2D,} \\ 2.5L^2 & \text{in 3D,} \end{cases} \tag{2}$$

where L is the linear lattice size. Since the queue address is treated as a cyclic number, $q_{\max}$ is interpreted as the minimum size that the queue memory should have. This is a factor of the order of L smaller than the necessary stack size.

The drawback of the queue strategy is the need to use two pointers. This would slow down a simulation on a general purpose computer. But the speed of the SPC is not affected by the use of two pointers. So we implemented the queue in hardware in the case of the 3D SPC.

Unfortunately, no rigorous theory supporting the queue size limits given in (2) is known to us. These limits can easily be justified at zero temperature, when a growing cluster has a simple shape with flat faces, and only the spins on its surface contribute to the queue size. As the temperature increases, the cluster surface becomes more rough, so for a given cluster size the surface increases. On the other hand, the mean cluster size decreases with increasing temperature. So the restrictions (2), while not proved, seem reasonable.

To be on the safe side, we chose the queue memory size to be several times larger than required by (2) for the largest lattice size. Moreover, we included a special register in hardware, which stores the largest queue size occurring during the simulation. This completely rules out the possibility of unnoticed queue overflow, and permits the determination of more-precise queue size limits at different temperatures and for different lattice sizes and shapes.

Next, we discuss the structure of the memory used to store the spin values. What is the most effective way to organize such a memory?

In view of the speed of the cluster growth process, it is desirable to process the nearest neighbors of the central spin simultaneously instead of sequentially. To this purpose, the lattice is divided into a number of sublattices called *blocks*. This is done in such a way that each of the 7 spins in a node configuration belongs to a different block. Simultaneous access to all blocks enables the processing of all 6 neighbor spins in parallel.

Since the number of blocks cannot be less than the number of spins in a node configuration, the number of blocks must be at least 7. However, the sublattice decomposition must be compatible with the boundary conditions, which are periodic, and with the lattice sizes, which are restricted to powers of 2. The number of blocks must be equal to 8.

The actual block number b assigned to each spin is selected by the following formula

$$b=(x+2y+3z)\mod 8, \tag{3}$$

where the integers x, y, and z are the usual orthogonal lattice coordinates of the spins. It is easy to check that the definition (3) ensures that all spins in each node have different block numbers.

Simultaneous access to all 8 blocks yields 8 spins. Only 7 of these are part of the node configuration. Therefore we included some additional hardware in the SPC to select the central spin, its left neighbor, its right neighbor, etc. This hardware circuitry is designated as the block-spin-to-neighbor-spin (BN) distributor in the SPC block diagram (Fig. 1).

Each block of spins is stored in a physically separate memory (B_n in Fig. 1). Each spin can be a member of 7 different node configurations: it can act as a central spin, or any of 6 neighbors. But for each node configuration, the queue supplies only one address, that of the central spin. This address requires minor changes, dependent on the memory block number, to obtain the proper node spins from the block memories. Therefore, each block memory has a separate address-changing circuit (A_n in Fig. 1).

The values of the node spins are used in the decision-making circuit (DMC) to decide which of the neighbor spins are included in the cluster and thus must be flipped. If such spins are found, the DMC gives the necessary signals to flip the spins in the block memories, and to write their addresses to the queue. When the flips and writes are finished, it asks the queue for a new mode address.

The decisions taken in the DMC involve the probability of an active bond (1). Thus, the spin is included in the cluster if it has the right direction *and* if expression (1) exceeds a (pseudo) random number.

For this purpose, the SPC includes a random number generator (RNG). The importance of using an RNG of sufficient quality has amply been demonstrated and emphasized.[16,18–21]

To provide good random numbers, we use an RNG consisting of two separate shift-register type hardware random number generators—G_a and G_b. The final random numbers, which are to be compared with expression (1), are produced by combining the output of G_a and G_b by means of the bitwise 'exclusive OR' ($\oplus$) operation. Both G_a and G_b produce 32-bit random numbers $X(n)$ according to the following algorithm:

$$X(n)=X(n-n_1)\oplus X(n-n_2), \tag{2}$$

where n_1 and n_2 are integers in the range between 0 and 2^{14}. These integers are programmable by the host computer which controls the SPC. For properly chosen integers $n_2>n_1$, the period of the pseudo-random sequence $X(n)$ is

$$2^{n_2}-1. \tag{5}$$

Suitable choices for n_1 and n_2 appear in Ref. 22 and references cited therein. In particular, the data shown below were obtained with

$$n_1^{\mathbf{a}}=1393 \quad n_2^{\mathbf{a}}=4423 \quad n_1^{\mathbf{b}}=471 \quad n_2^{\mathbf{b}}=9689, \tag{6}$$

where the indices **a** and **b** distinguish between G_a and G_b. Putting (5) and (6) together, we see that the periods of the random number sequences are huge.

TABLE I. Comparison between SPC data for the 2D Ising model and exact results at T_c.

L	N	E	E_{ex}	C	C_{ex}
16	$5\cdot 10^9$	1.453070(6)	1.453065	1.4988(2)	1.4987
32	$5\cdot 10^8$	1.433642(13)	1.433658	1.8459(10)	1.8468
64	10^8	1.423936(17)	1.423938	2.1935(17)	2.1922
128	$5\cdot 10^7$	1.419071(17)	1.419076	2.5306(57)	2.5363
256	$7.5\cdot 10^6$	1.416622(24)	1.416645	2.865(24)	2.8798

The performance of shift-register RNGs has been tested extensively in combination with the Wolff algorithm.[23] On this basis, we expect that systematic effects will be negligible in comparison with any realistic statistical errors, both for simulations in two and in three dimensions using the 3D SPC.

As final tests of the 3D SPC, we performed simulations of 2D and 3D Ising models by the 3D SPC.

Simulations of the 2D model is possible because the nearest-neighbor couplings in any of the x, y and z directions can be independently set to zero. Table I shows the results obtained by the 3D SPC at the 2D critical point $\beta_c=0.44068679$ for 5 different lattice sizes $L\equiv L_x=L_z$. In the simulations of the 2D Ising model we have kept $J_x=J_z=1$, $J_y=0$, and $L_y=16$.

Here N is the total number of clusters flipped during the simulation. Data were taken at intervals of 50 Wolff clusters.

The mean number of spins in a cluster at T_c is about 1/4 of all lattice spins. The SPC generates about 7 random numbers per cluster spin. So the SPC uses a total of about $30\times N\times L^2$ random numbers, which is of the order of 10^{13} for all the cases in Table I.

The specific heat C was calculated according to the formula $C=\beta^2(\langle E^2\rangle-\langle E\rangle^2)$. Exact values of the energy per spin E_{ex} and of the specific heat C_{ex} were calculated according to Ref. 24. The agreement between the Monte Carlo data and the exact results confirms the proper operation of the 3D SPC and the good quality of the pseudo-random numbers generated by the SPC.

For the 3D model, the exact values of the thermodynamic properties, and even that of the critical point, are unknown. However, from previous work (see, e.g., Refs. 7 and 9) it is clear that β_c should be close to $\beta_3=0.221653$. In Table II we display data obtained

TABLE II. 3D Ising results taken from β_3.

L	N	E	M	C	$\langle S(0)S(L/2)\rangle$
16	10^9	1.034618(12)	0.263655(18)	1.80072(29)	0.0835743(95)
32	10^8	1.007101(35)	0.184671(62)	2.2380(11)	0.041072(22)
64	$2.5\cdot 10^7$	0.996708(60)	0.12880(18)	2.7080(75)	0.020022(46)
128	10^7	0.992854(44)	0.08976(25)	3.256(26)	0.009741(44)
256	10^7	0.991367(19)	0.06184(20)	3.772(42)	0.004651(28)

TABLE III. Comparison between 3D SPC and GP results.

L	type	N	E	C	$\langle S(0)S(L/2)\rangle$	Q
16	SPC	10^9	1.034618(12)	1.80072(29)	0.0835743(95)	0.633806(45)
	GP	$1.5\cdot 10^9$	1.034616(14)	1.80101(19)	0.0835709(95)	0.633752(40)
32	SPC	10^8	1.007101(35)	2.2380(11)	0.041072(22)	0.62863(18)
	GP	$1.2\cdot 10^9$	1.007129(10)	2.2370(4)	0.041077(7)	0.62877(6)

at β_3 using lattices with different sizes $L \equiv L_x = L_y = L_z$.

We see that the 3D SPC is not only able to calculate the energy E, specific heat C, magnetization M, and magnetic susceptibility, but also the spin–spin correlation function $\langle S(0)S(L/2)\rangle$.

Since the mean cluster size at β_3 is roughly proportional to L^2, one has to flip of the order of L clusters to obtain an independent spin configuration. This sets the time scale for the data sampling procedure. Data were taken at intervals of 10 Wolff clusters for $L=16$, 32; 50 clusters for $L=64$; 100 clusters for $L=128$; and 200 clusters for $L=256$.

The largest queue size that occurred in the simulations at β_3 did not exceed $1.5L^2$. Thus the 3D condition (2) is indeed satisfied for the very large lattices and the long simulations used here.

Furthermore, independent Monte Carlo data of sufficient accuracy, obtained by simulations on a general-purpose computer (GP), are available for system sizes $L=16$ and 32[25] and allow another sensitive test. These data, shown in Table III, apply to the energy, the specified heat, the spin–spin correlation over half the system size, and the dimensionless ratio $Q = \langle M^2\rangle^2 / \langle M^4\rangle$.

The data from both sources appear to be in a satisfactory agreement.

We conclude that the first 3D cluster SPC has passed a number of critical tests. It will serve as a power tool to study the thermodynamics of the Ising model, in particular the critical properties. Furthermore it will give an opportunity to investigate shift-register based algorithms for random number generation, which is extremely important in order to avoid biases in Monte Carlo simulations in a general context.

We are grateful to W. Selke, A. F. Bakker, A. Compagner, J. R. Heringa, and A. Hoogland for their cooperation and for sharing their insights. Furthermore, we are indebted to H. J. M. van Grol for his valuable suggestion to use a queue instead of a stack memory.

This work was partially supported by Grants 07-13-210 of NWO, the Dutch Organization of Scientific Research, INTAS-93-0211, M0Q000 of the International Science Foundation (ISF), and 93-02-2018 of the Russian Fund for Fundamental Research (RFFR).

[1]e-mail: talapov@itp.ac.ru

[1] L. Onsager, Phys. Rev. **65**, 117 (1944).

[2] W. Selke, L. N. Shchur, and A. L. Talapov, in *Annual Reviews of Computational Physics*, ed. by D. Stauffer, World Scientific, Singapore (1994), p. 17.

[3] A. J. Liu and M. E. Fisher, Physica A **156**, 35 (1989).

[4] B. G. Nickel and J. J. Rehr, J. Stat. Phys. **61**, 1 (1990).

[5] G. A. Baker, B. G. Nickel, M. S. Green, and D. I. Meiron, Phys. Rev. B. **17**, 1365 (1978).

[6] J. C. Le Guillou and J. Zinn-Justice, Phys. Rev. B **21**, 3976 (1980).

[7] A. M. Ferrenberg and D.P. Landau, Phys. Rev. B **44**, 5081 (1991).

[8] Vl. S. Dotsenko, P. Windey, G. Harris *et al.*, Phys. Rev. Lett. **71**, 811 (1993); hep-th/9504076.

[9] H. W. J. Blöte and G. Kamieniarz, Physica A **196**, 1 (1993).

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

[11] U. Wolff, Phys. Rev. Lett. **60**, 1461 (1988); Nucl. Phys. B **300**, 501 (1988).

[12] A. L. Talapov, L. N. Shchur, V. B. Andreichenko, and Vl. S. Dotsenko, Mod. Phys. Lett. B **6**, 1111 (1992).

[13] A. L. Talapov, V. B. Andreichenko, Vl. S. Dotsenko, and L. N. Shchur, Int. J. Mod. Phys. C **4**, 787 (1993).

[14] A. L. Talapov and L. N. Shchur, Europhys. Lett. **27**, 193 (1994).

[15] A. L. Talapov and L. N. Shchur, J. Phys.: Condens. Matter **6**, 8295 (1994).

[16] A. Compagner, Am. J. Phys. **59**, 700 (1991); J. Stat. Phys. **63**, 883 (1991).

[17] J. R. Heringa, private communication.

[18] A. M. Ferrenberg, D. P. Landau, and Y. J. Wong, Phys. Rev. Lett. **69**, 3382 (1992).

[19] W. Selke, A. L. Talapov, and L. N. Shchur, JETP Lett. **58**, 665 (1993).

[20] I. Vattulainen, T. Ala-Nissila, and K. Kankaala, Phys. Rev. Lett. **73**, 2513 (1994).

[21] P. D. Coddington, Int. J. Mod. Phys. C **5**, 547 (1994).

[22] J. R. Heringa, H. W. J. Blöte, and A. Compagner, Int. J. Mod. Phys. C **3**, 561 (1992).

[23] L. N. Shchur and H. W. J. Blöte, to be published (1995).

[24] A. E. Ferdinand and M. E. Fisher, Phys. Rev. **185**, 832 (1969).

[25] H. W. J. Blöte, J. R. Heringa and E. Luijten, to be published (1995).

Published in English in the original Russian journal. Edited by Steve Torstveit.