Journal of Optoelectronics and Advanced Materials Vol. 5, No. 4, December 2003, p. 971 - 976

MONTE CARLO METHODS IN SEQUENTIAL AND PARALLEL COMPUTING OF 2D AND 3D ISING MODEL

M. Diaconu*, R. Puscasu, A. Stancu

"Al. I. Cuza" University, 11 Bd. Carol I, Iasi, 6600, Romania

Because of its complexity, the 3D Ising model has not been given an exact analytic solution so far, as well as the 2D Ising in non zero external field conditions. In real materials the phase transition creates a discontinuity. We analysed the Ising model that presents similar discontinuities. We use Monte Carlo methods with a single spin change or a spin cluster change to calculate macroscopic quantities, such as specific heat and magnetic susceptibility. We studied the differences between these methods. Local MC algorithms (such as Metropolis) perform poorly for large lattices because they update only one spin at a time, so it takes many iterations to get a statistically independent configuration. More recent spin cluster algorithms use clever ways of finding clusters of sites that can be updated at once. The single cluster method is probably the best sequential cluster algorithm. We also used the entropic sampling method to simulate the density of states. This method takes into account all possible configurations, not only the most probable. The entropic method also gives good results in the 3D case. We studied the usefulness of distributed computing for Ising model. We established a parallelization strategy to explore Metropolis Monte Carlo simulation and Swendsen-Wang Monte Carlo simulation of this spin model using the data parallel languages on different platform. After building a computer cluster we made a Monte Carlo estimation of 2D and 3D Ising thermodynamic properties and compare the results with the sequential computing. In the same time we made quantitative analysis such as speed up and efficiency for different sets of combined parameters (e.g. lattice size, parallel algorithms, chosen model).

(Received July 4, 2003; accepted August 21, 2003)

Keywords: Ising model, Monte Carlo method, Spin cluster, Magnetic susceptibility

1. Introduction

Monte Carlo methods provide approximate solutions to a variety of mathematical problems by performing statistical sampling experiments on a computer.

Numerical simulations are now, more than ever before, a valuable tool for studying the properties of mathematical models in physics [1, 2]. As computers become more and more powerful, the kind of problems being solved with their help become more and more difficult and time consuming, demanding in turn more powerful algorithms.

We used Visual C^{++} and Fortran to simulate nearest neighbors interactions in a 2D or a 3D Ising lattice [3], in zero magnetic field. The system can be described by a set of properties that are temperature-dependent. These are energy, magnetization, heat capacity and magnetic susceptibility. At a specific temperature called the critical temperature, the model shows characteristics of a phase transition. The phase transition in the Ising model is a transition between an ordered and a non-ordered state. In real materials we also have transitions of this kind.

We tried different algorithms for different systems to see the differences in establishing the critical point and, also, we tried to reduce the time necessary for running the program by developing some parallel computing algorithms.

* Corresponding author: diaconumary@yahoo.com

We worked with Ising model, which is one of the simplest for phase transitions in statistical physics. The 1D Ising model presents one of the simplest interacting systems in the absence of an applied field. Because the 1D case does not present a phase transition, the interest in studying such a system is very small. Onsager showed in 1944 that the 2D system presents a phase transition.

The spins are considered to be aranged in a lattice, usualy square or triangular in 2D, in fixed positions and the interactions are only between first order neighbors.

There are many methods for deriving the partition function of the 2D Ising model and some scientists tried to generalize some of these methods for the 3D case. We also tried to apply some methods with good results in 2D case for a 3D case.

2. The algorithms

For the beginning, we are presenting some types of algorithms that we used in our paper.

We start with a single cluster algorithm, called Wolff algorithm [4]. This is a realistic cluster algorithm for lattice models. The basic idea is to look for clusters of similarly oriented spins and then change their orientation all at once.

The algorithm is as follows:

- we choose a seed spin at random from the whole lattice;
- we look at each of the neighbors of that spin. If they are pointing in the same direction as the seed spin we add them to the cluster with the probability:

$$P_{add}=1-\exp(-2J/k_BT) \tag{1}$$

- for each spin that was added in the last step we examine each of its neighbors to find the ones that are pointing in the same direction and if they are add each of them to the same probability P_{add}. If some neighbors are already cluster members, they do not need to be added again. For spins that were considered for addition but rejected before, they get another chance to be added to the cluster at this step;
- the previous step will be repeated as many times as necessary until there are no spins left in the cluster whose neighbors have not been considered for inclusion in the cluster;
- change the cluster orientation.

Another cluster algorithm is Swendsen-Wang algorithm [5]. The difference between Wolff cluster algorithm is that this algorithm is using more spin clusters at the same time. That is why it is called multi-cluster algorithm. It inspects all nodes and if $s_i=s_j$ a bond between sites is created with a probability P_{add}. After this clusters are constructed using sites wich are connected by bonds. Next step is to give to each cluster a random value, +1 or –1.

We also used the so called entropic sampling method, introduced by Lee [6]. This method gives the state density function, g{m,s} and from this we can obtain the partition function of the system and also all macroscopic quantities, e.g. specific heat or susceptibility [7].

The interaction energy is considered only between the nearest neighbors, 4 in 2D case and 6 in 3D case.

The equation for energy is the sum over all nearest neighbor pairs, with spins equaling either +1 or -1, and multiplied by a factor J, which describes the strength of the interaction between spins.

The total energy of the system is given by the formula:

$$H=-\frac{1}{2}\sum_{i.j}^{N}S_iS_jJ_{ij}-h\sum_{i=1}^{N}s_i \tag{2}$$

where J_{ij} is the exchange energy between the i and j spins, h is the external applied field and S = +1 or -1 as spin state.

$$J_{ij} = \begin{cases} J \quad if \quad the \quad spins \quad are \quad first \quad order \quad neighbors \\ 0 \quad otherwise \end{cases} \tag{3}$$

The coupling constant J represents a measure for the exchange interaction. J is positive for ferromagnetic coupling and negative for antiferromagnetic coupling.

We consider an Ising system and we say that the macrostate of the spins system is a series of values of the couple {m,s} represented by a point in the plane of all possible values of m and s. This plane is the phase space of our problem. The degenerescence g(m,s) of this macrostate we shall also call it state density like m and s are varying continuously. Is this the two variable function that we want to obtain using Monte Carlo simulations. Once obtained the state density g(m,s), the calculations for obtaining the partition function is very easy**.**

We considered different dimensions for our system and we studyed the influence of the dimensionality on the system properties.

We obtained a file which contains on the first column the values for m ($m = \sum_i S_i$), on the second column the values for s ($s = \sum_{i,j} S_i S_j$) and on the third column the values for g, the energy density function.

We used the folowing formulas:

$$Z = \sum g(E_i) e^{-\beta E_i} \text{ is the partition function of the system} \tag{4}$$

$$\langle E \rangle = \frac{\sum g(E_i) E_i e^{-\beta E_i}}{Z} = -\frac{1}{Z}\frac{\partial Z}{\partial \beta} \text{ represents the mean energy} \tag{5}$$

$$\langle E^2 \rangle = \frac{\sum g(E_i) E_i^2 e^{-\beta E_i}}{Z} = \frac{1}{Z}\frac{\partial^2 Z}{\partial \beta^2} \text{ is the mean square energy} \tag{6}$$

The heat capacity will be calculated using the formulas:

$$C = \frac{\partial \langle E \rangle}{\partial T} = \frac{\partial \beta}{\partial T}\frac{\partial E}{\partial \beta} = -\frac{1}{kT^2}\frac{\partial \langle E \rangle}{\partial \beta} \tag{7}$$

$$\frac{\partial \langle E \rangle}{\partial \beta} = -\frac{\partial}{\partial \beta}\left(\frac{1}{Z}\frac{\partial Z}{\partial \beta}\right) = -\frac{1}{Z}\frac{\partial^2 Z}{\partial \beta^2} + \left(\frac{\partial Z}{\partial \beta}\right)^2 \frac{1}{Z^2} = \left(\frac{1}{Z}\frac{\partial Z}{\partial \beta}\right)^2 - \frac{1}{Z}\frac{\partial^2 Z}{\partial \beta^2} = \langle E \rangle^2 - \langle E^2 \rangle \tag{8}$$

We obtain:

$$C = \frac{\langle E^2 \rangle - \langle E \rangle^2}{KT^2} \tag{9}$$

For susceptibility we used:

$$\chi = \frac{\langle M^2 \rangle - \langle M \rangle^2}{kT} \tag{10}$$

where we have:

$$\langle M \rangle = \frac{\sum_i m_i e^{-\beta E_i}}{Z} \text{ and } \langle M^2 \rangle = \frac{\sum_i m_i^2 e^{-\beta E_i}}{Z} \tag{11}$$

For parallel computing [8], we present the following as an outline of the steps required:

a. Lattice partitioning

This means a scattered strip (column-cyclic) partitioning of the lattice, so that the portion of the lattice allocated to each processor has L=(WP) partitions of width W, i.e. L=P columns and L2=P sites in total.

b. Selection of an initial random site

One of our processors selects the initial random site and broadcasts the information to the other processors.

c. Local expansion of the cluster

Each processor maintains a queue that stores those sites in the local lattice that are in the current generation, and also maintains a communication bufer to collect all communication requests. Each local site in the current generation is fetched from the queue, and used as a parent site to expand the cluster. Bonds are made with the appropriate probability between the parent site and all its neighboring sites, unless the neighboring site is already an element of the cluster. At this time, if the parent site is on the boundary (the left or right edge) of a partition, communication with the left or right processor may be needed. Such communication requests are saved into the communication buffer for collective communication.

All local connected sites are added to the cluster and the queue for the next generation of sites, and the spin values at these sites are updated.

d. Collective communication

Any communication requests are collected from the communication bufer and sent to the destination processors.

e. Remote expansion of the cluster

Each site in the received bufer is checked to see whether it is already an element of the cluster. If not, the site is added to the local cluster list and the queue for the next generation, and its spin value is updated.

f. Check for termination

The final step checks whether there are any sites in the new generation. This can be done easily by using a global reduction function such as logical AND. If every processor has the empty queue, the algorithm halts. If not, steps c, d and e are repeated. The parallel code to implement this message-passing program was written in C^{++}. This algorithm should be easily implementable in High Performance Fortran (HPF) using a column-cyclic data distribution.

3. Results

Figs. 1-5 show the results. Fig. 1 shows the energy density versus m while Fig. 2 illustrates the dependence of s function on m. The variations of the magnetization, magnetic susceptibility and heat capacity with the temperature are presented in Figs 3-5.

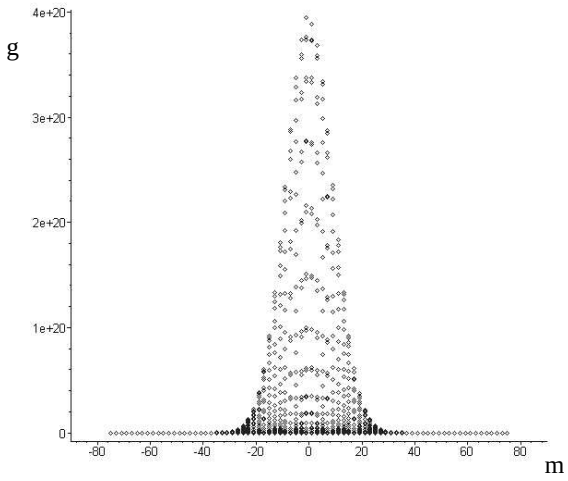

Fig. 1. The energy density (g) versus m for a 5×5×3 system obtained with the entropic sampling method.

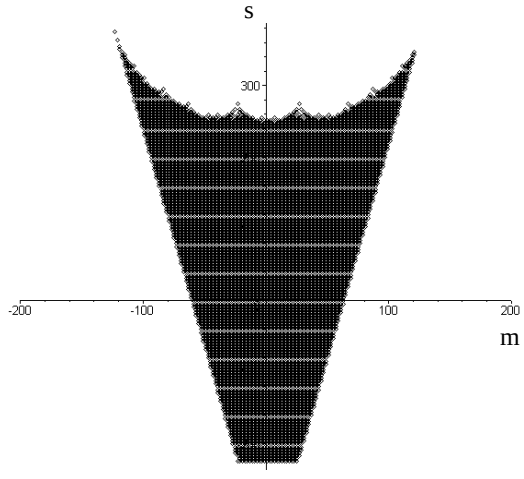

Fig. 2. The graph of s function versus m for a 5×5×5 system for the entropic sampling method.

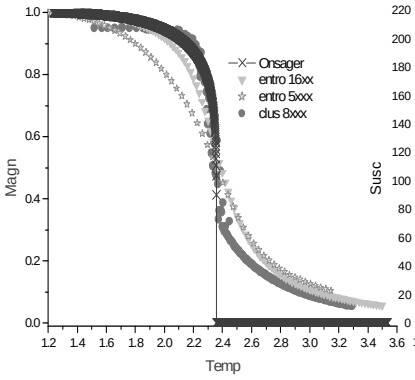

Fig. 3. Magnetization versus temperature for different systems, and obtained with different methods.

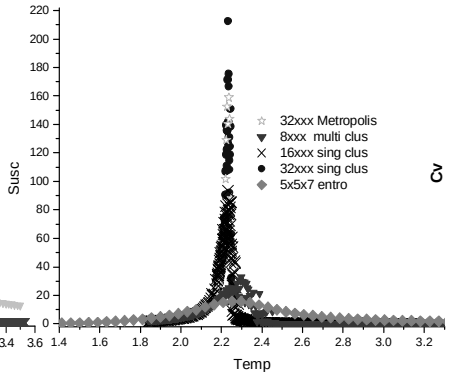

Fig. 4. Magnetic susceptibility versus temperature for different systems and methods.

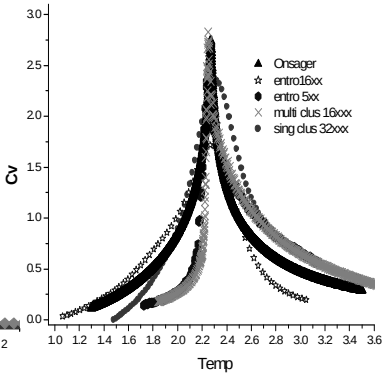

Fig. 5. Heat capacity versus temperature for different systems and methods.

4. Discussion

The magnetization, is the sum of all individual spins. It depends on how many spins are pointed in the same direction, and is comparable to the amount of order in the system. As the temperature increases, and therefore more randomness is introduced to the system, the magnetization decreases. Once again, the steepest slope of the graph is found near the critical point. Over a broad temperature scale, the graph of magnetization would range between exactly 1 at low temperatures and close to 0 at high temperatures. We compared our results with Onsager's data and observed that the critical point is less distinct for small lattice sizes.

Because the properties related to the magnetization tend to converge faster than those based on energy, they are more commonly studied and used as indicators of critical phenomena in the model.

For magnetic susceptibility, the first derivative of the magnetization, there is no exact solution, so we compared our results with those obtained using the well-known Metropolis method [10]. At critical point we see a discontinuity in the graph.

Heat capacity is essentially the first derivative of the energy. Specific heat is comparable to the amount of energy needed to raise the temperature by one unit. At approximately the critical point, a spike in the specific heat is found, which is essentially a discontinuity in the graph. The singular point appears at T=Tc; Onsager's prediction of singular point is found in the simulation result.

5. Conclusions

The theory behind the Ising model assumes that the lattice size is infinite. By decreasing the size of the lattice, it becomes easier to manage and understand, but the key features of the model are lost. In significantly smaller lattice sizes, the peaks in specific heat and magnetic susceptibility become more rounded, and the critical point is much less distinct.

Different ways of handling the edges of the lattice are called boundary conditions. We looked at two types of boundary conditions: free and periodic.

With free boundary conditions, the edges are surrounded by empty space. With periodic boundary conditions, to each spin is given its full six neighbors in 3D case. A spin at the edge wraps around to a spin on the other edge, and a spin on the corner wraps to other corners. Periodic boundary conditions better represent the infinite system that the model is based on.

It is possible that parallel programming techniques could also be used to improve the performance of the algorithms. On a massively parallel machine, parallelism could be extracted by a combination of the strip partitioning algorithm and the independent (or job-level) parallelism of

running independent Monte Carlo simulations with different random number streams on different groups of processors. For example, one might run 16 independent simulations, each of which use 16 processors. In this scenario, the main advantage of using the parallel algorithm is that it avoids the memory limitations of a single processor, and allows the use of largerlattice sizes.

Most serial Monte Carlo codes are readily adaptable to a parallel environment. Strengths are still for multi-dimensional problems and complex geometries. Care must be taken to assure reproducible results and must assure that the calculations on different processors are independent.

Optimising the performance of parallel programs [11] is significantly more difficult than optimising the performance of normal serial programs. Despite the efforts made during program design, most parallel performance tuning still relies on a ``measure-modify approach" because of the difficulties in foreseeing the effect on performance of factors such as input data, the number of available processors and the characteristics of the communication network that connect them. Performance modelling provides an alternative to this time-consuming development process by bringing the focus of performance optimisation for parallel programs back from the tuning phase to the design phase. This is possible because of the predictive capabilities of performance models, which empower programmers to make better decisions during design.

We also have to notice that cluster methods can be used for amorphous materials, because in both cases we have a short range arrangement corresponding to about 10-100 particles.

In conclusion, we have a method that produces good results and has been reasonably optimized. Tests still remain to be run for larger lattices.

References

[1] K. Binder, D. W. Heermann, Monte Carlo Simulation in Statistical Physics, (Springer-Verlag, Berlin, 1988.
[2] M. E. J. Newman, G. T. Barkema, Monte Carlo Methods in Statistical Physics, Oxford, 1999.
[3] David Chandler, Introduction to Modern Statistical Mechanics, Oxford, 1987.
[4] U. Wolff, “Collective Monte Carlo for spin systems”, Physical Review Letters **62**, 361(1989).
[5] R. H. Swendsen, J. S. Wang, Phys. Rev. Lett. **58**, 86(1987).
[6] J. Lee, Phys. Rev. Lett. 71, 211 (1993), **71**, 2353(E) (1993).
[7] I. Shteto, J. Linares, F. Varret, Phys. Rev. E **56**, 5128 (1997).
[8] G. Fox, M. Johnson, G. Lyzenga, S. Otto, J. Salmon, D. Walker, Solving Problems on Concurrent Processors, Vol. 1, Prentice-Hall, Englewood Cliffs, 1988.
[9] Colin J.Thompson, Mathematical Statistical Mechanics, Princeton, 1972.
[10] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, Chem. Phys. **21**, 1087 (1953).
[11] Raul Toral, Dynamical properties of a new fast algorithm for the simulation of the Ising model. J. Phys. A: Math. Gen. 21 (1988) L315-L320.