Optimization of the Exhaustive Enumeration Algorithm in the Ising Model

Mikhail A. Padalko[1], Petr D. Andriushchenko[1,2], Konstantin V. Nefedev[1,2]

[1]Far Eastern Federal University, Vladivostok, Russia, kukurbitofedor@gmail.com
[2]Institute of Applied Mathematics, Russian Academy of Science, Vladivostok, Russia, pitandmind@gmail.com,
nefedev.kv@dvfu.ru

Abstract

An optimized method for the precise calculation of the Ising model is presented. The algorithm makes it possible to calculate two-dimensional 10x10 lattices for periodic boundary conditions on ordinary personal computers. The method is applicable to various types of lattices. The proposed optimized method of precise calculation makes it possible to test the effectiveness of Monte Carlo methods.

1 Introduction

The Ising model is a set of spins, each of which can take one of two the states: +1 or -1. The interaction energy of spins with each other and with an external magnetic field is calculated by the formula

$$E = -J_{ij} \sum_{<i,j>} s_i s_j - H \sum s_i \tag{1}$$

where J_{ij} – exchange interaction constant (positive for ferromagnet, negative for antiferromagnet), sum symbol with index $<i, j>$ denotes the summation over all neighboring spins, s_i and s_j – spin values, H - projection of the external magnetic field strength [1]. Each spin configuration of spins has a certain energy and magnetization. The statistical sum of a spin system that is in thermal equilibrium with the environment at temperature T is calculated by formula

$$Z(T,H) = \sum_{E,M} g(E,M) \cdot \exp(-E/T - M \cdot H/T) \tag{2}$$

where $g(E, M)$ – is a multiplicity of the degeneracy of states in energy and magnetization. [2]. If one knows the partition function, it is possible to calculate all basic thermodynamic characteristics, such as internal energy, free energy, magnetic susceptibility, heat capacity, etc. [2] In the Ising model $g(E, M)$ is a discrete function of variables E and M [3]. For finite lattices $g(E, M)$ can be represented in matrix form. For most systems, the only way to find precise mean of $g(E, M)$ is an exhaustive enumeration method. The difficulty of this approach is that it is necessary to go through all the system configurations, the number of which is equal to 2^N, where N – is number of spins in the system. Calculation time using the proposed method grows exponentially with increasing number of spins. It was verified that the exhaustive enumeration method allows calculating $g(E, M)$ for the Ising model on the 6x6 square lattice (2^{36} configurations) in a time of the order of several hours using the CPU of ordinary personal computers. For example, on a quad-core AMD Phenom (tm) II X4 970 processor with a maximum 3.6 GHz frequency with paralleling on 4 cores, the calculation time is 1 h 32 min. The calculation for the Ising model on the 7x7 square lattice will take about one and a half years.

We propose an optimized method of calculation, which allows us to significantly reduce the computation time of the discrete function $g(E, M)$. Using this method, the computation time of the 6x6 2D Ising is 0.875 sec without parallelization on the same computing power. The method makes available calculations of 10x10 lattices for a time approximately equal to 1.5 h.

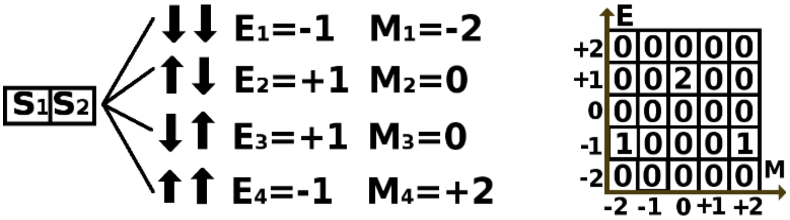
Knowing the partition function allows us to test the effectiveness of various probabilistic algorithms: Metropolis [4], Svendsen-Wang [5], Wolf [6], Wang-Landau [7], and others.

In: Sergey I. Smagin, Alexander A. Zatsarinnyy (eds.): V International Conference Information Technologies and High-Performance Computing (ITHPC-2019), Khabarovsk, Russia, 16-19 Sep, 2019, published at http://ceur-ws.org

2 Algorithm Description

The idea of the method is to choose a small subsystem, calculate all of its possible states and successively expand it to the original system. In the case of rectangular lattices, it is advisable to choose one any spin. At each step, the spin of the original system will be added to the subsystem. The process needs to be continued until the original system will be obtained.

Let introduce the concept of opened and closed spins. We call a subsystem spin opened if the number of its neighbors in a subsystem is less than the number of its neighbors in the original system, i.e. spin borders on a spin that is not yet included in the considered subsystem. Otherwise we will call it closed. In the process of calculating, it's convenient to represent the function *g (E, M)* as a table or matrix, along the horizontal and vertical axis of which all possible values of the magnetization and energy are plotted correspondingly. The value of each element of such matrix is equal to the number of configurations with the corresponding values of energy and magnetization. If a configuration with certain values of energy and magnetization does not exist, then the value of the matrix element will be zero (Fig. 1).



E \ M	-2	-1	0	+1	+2
+2	0	0	0	0	0
+1	0	0	2	0	0
0	0	0	0	0	0
-1	1	0	0	0	1
-2	0	0	0	0	0

Figure 1: The process of building the degeneration matrix in energy and magnetization for a system of two spins: s_1 and s_2.

The algorithm is applicable only for cases when the exchange interaction constant is the same for all pairs of interacting spins. Any lattice architecture and any dimensions may be used.

The algorithm consists of 9 steps:

1) Choose a small subsystem (for example 1 spin).

2) Form the list of subsystem open spins: $L=\{s_1, s_2, ..., s_n\}$, here n is quantity of opened spins and $s_1, s_2, ..., s_n$ are their numbers.

3) Using the method of exhaustive enumeration, we calculate the degeneration matrices in energy and magnetization for each configuration of opened spins.

4) Expand the subsystem by including the spin of the original system to it.

5) Make a copy of all existing degeneration matrices. So one obtains 2 identical sets of matrices.

6) Let the value of the added spin be -1. For each configuration of spins from the list *L*, we calculate the changes in energy ΔE and magnetization ΔM. Then we shift all elements in the corresponding matrices of the 1st set by the value of ΔE vertically and ΔM horizontally. ΔE is equal to the interaction energy of the added spin with spins from the list *L*, $\Delta M = -1$.

7) Let the value of the added spin be +1. For each configuration of spins from the list *L*, we calculate the changes in energy ΔE and magnetization ΔM. Then we shift all elements in the corresponding matrices of the 2nd set by the value of ΔE vertically and ΔM horizontally. ΔE is equal to the interaction energy of the added spin with spins from the list *L*, $\Delta M = +1$.

8) Update the list *L* of opened spins, taking into account added spin.

9) If the extended subsystem is equal to the original system, then the degeneracy matrix *g(E, M)* of the original system is obtained by algebraic addition of all degeneration matrices. The calculation is complete. Otherwise, in both sets, we perform the algebraic addition of the degeneration matrices corresponding to the identical spin configurations of the list *L* updated in step 8. Go back to step 4.

3 Algorithm for the Ising Model on the 2x2 Square Lattice

Consider ferromagnetic the 2D Ising model on the 2x2 square lattice with free edge boundary conditions. In this example, it is possible to trace in details all the steps of the algorithm described above. The task is to obtain the degeneration matrix in energy and magnetization. For convenience, we number the spins, like in Fig. 2.

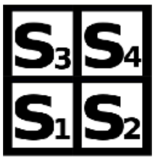


Figure 2: Spin numeration in the Ising model on the 2x2 square lattice.

Let us choose spin s_1 as the initial subsystem. The scheme of expanding the initial subsystem to the original system is shown in Fig. 3. Notations for the subsystems A, B, C is also given there. The original system is denoted as D. In the figure, the opened spins of each of the subsystems are shaded.

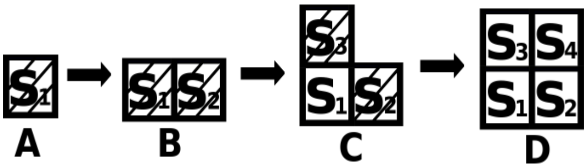


Figure 3: The scheme of expanding the initial subsystem A to the original system D. Hatched cell means that the spin of the subsystem is opened.

Figure 4 shows the first 9 steps of the proposed algorithm. On this picture one can see how an exhaustive enumeration of the configurations of subsystem A is carried out. In step 5, we copy the set of degeneration matrices obtained in step 3. Thus, we will obtain 2 identical sets of matrices. In step 6 and 7, the calculation of the changes in the magnetization ΔM and energy ΔE for each configuration of the spins of the list L and the added spin is shown. In step 9 we can see that only one matrix corresponds to each configuration of the spins from the updated list L. Therefore, according to the algorithm, the addition is not necessary. The set of degeneracy matrices obtained in step 9 fully describes the system B. Since B ≠ D, back to step 4.

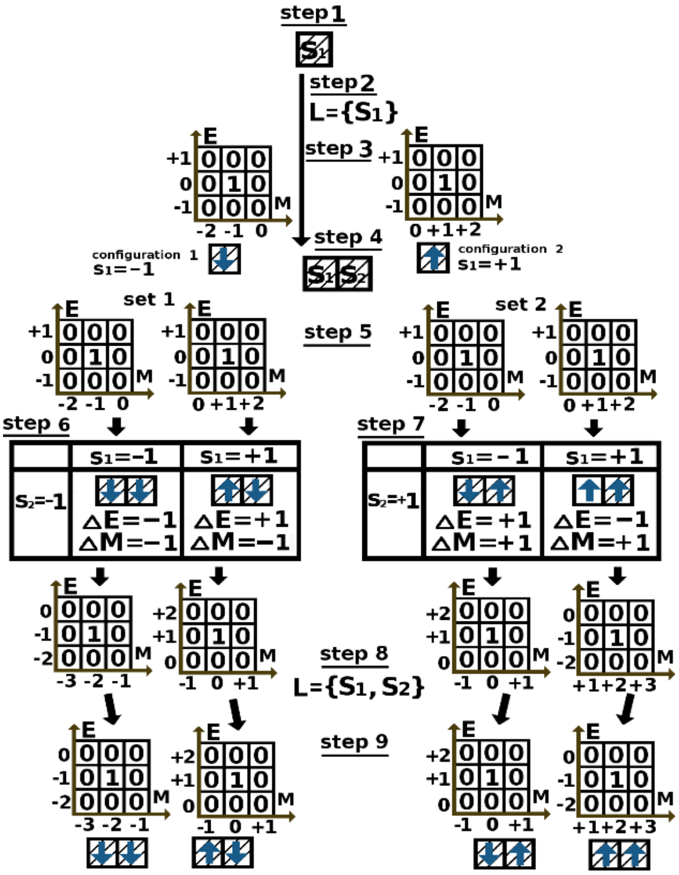


Figure 4: Visualization of the first 9 steps of the algorithm. On the scheme we can see: calculation ΔE and ΔM, shift of matrix elements at the moment of adding a spin.

At the time of the building system C in step 4, the list L contains 2 spins: s_1, s_2. Figure 5 shows the further steps of the algorithm. Add spin s_3 to the subsystem B. Copy 4 matrices of degenerations in the second set. Next we calculate the changes in the energy ΔE and magnetization ΔM for each configuration of spins from the list $L = \{s_1, s_2\}$ for the cases $s_3 = -1$ and $s_3 = +1$. Shift the matrix elements by the values of ΔE and ΔM vertically and horizontally, respectively. Next, update the list of opened spins L. Spin s_1 becomes closed for subsystem C, since the number of its neighbors in B is 2, as in the original system. Thus, s_2 and s_3 will be included in L. Next, we carry out the algebraic addition of matrices corresponding to the same configurations of the spins of the list L. Since system D is still not built, go back to step 4.

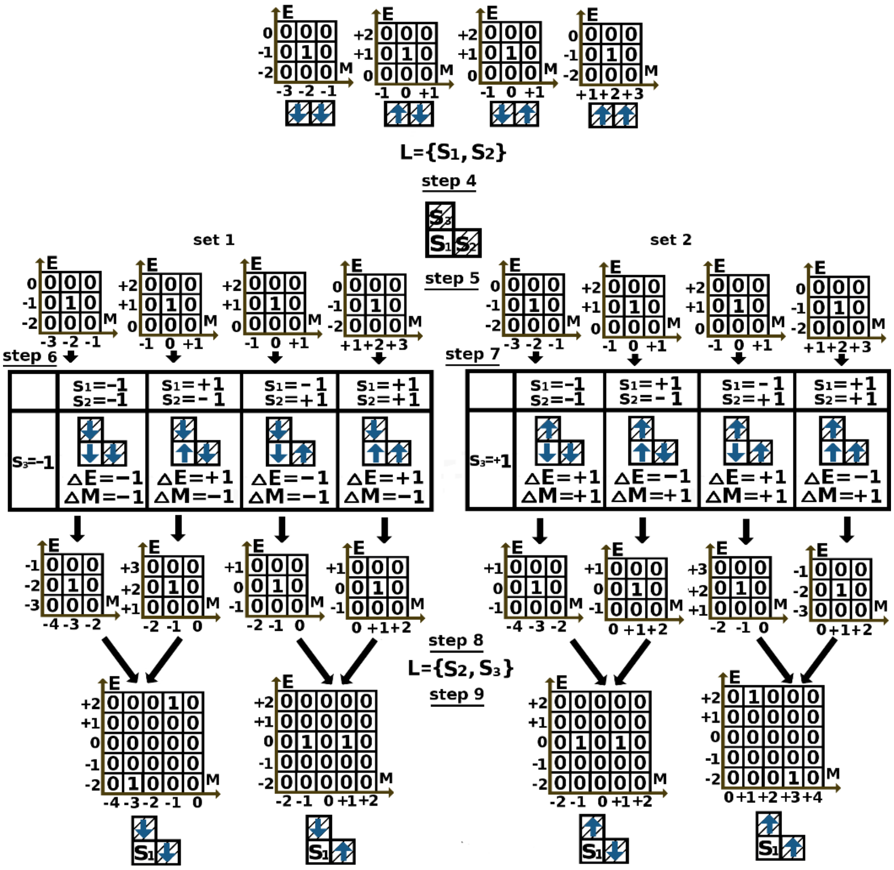


Figure 5: Visualization of the algorithm for subsystem C (steps 4-9).

Steps 4-7 will be similar to the previous ones, but after performing step 8, the list L will be empty, because all spins are closed. Because we have reached the subsystem D at step 9 it is necessary to add the resulting 8 degeneracy matrices algebraically. The resulting matrix one can see in Fig. 6.

$$
\begin{array}{r|ccccccccc}
E & & & & & & & & & \\
+4 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\
+3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
+2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
+1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 4 & 0 & 4 & 0 & 4 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hline
M & -4 & -3 & -2 & -1 & 0 & +1 & +2 & +3 & +4
\end{array}
$$

Figure 6: The resulting degeneration matrix in energy and magnetization for system D.

4 Algorithm Performance Analysis

Table 1 shows the performance of the exhaustive enumeration method on the example of the Ising model on the KxK square lattice with periodic boundary conditions with parallelization on 4 threads on an AMD Phenom (tm) II X4 970 processor.

Table 1: Performance of the exhaustive enumeration method for computing the KxK lattice with periodic boundary conditions on an AMD Phenom (tm) II X4 970 processor (with paralleling to 4 cores).

calculation number i	system size	calculation time t	real ratio t_i to t_{i-1}	theoretical ratio t_i to t_{i-1}
1	4x4	4.7 ms		
2	5x5	2400 ms	511	512
3	6x6	5580000 ms	2291	2048

Because the number of possible configurations of spins in the KxK square lattice is $2^{K \cdot K}$, the calculation will depend on K as

$$t\left(K\right) = C_1 \; 2^{K \cdot K}, \tag{3}$$

where C_1 is a constant defined by the implementation of the algorithm and the features of the computer architecture. Accordingly, with an increase in the linear size of the lattice from K to K + 1, the calculation time increases as

$$t\left(K+1\right) / t\left(K\right) = 2^{2K+1}. \tag{4}$$

In the last column of Table 1 one can see the time ratios calculated by the formula (4). They are approximately consistent with the real values in column 4. Using the formula (4), one can calculate the approximate time of the calculation of the 7x7 spin lattice: it will be longer than 6x6 213 times and will be 529 days with the same computing power.

For comparison, Table 2 shows the performance of the exhaustive enumeration method on the example of the Ising model on the KxK square lattice with periodic boundary conditions.

Table 2: Performance of the proposed optimized algorithm for computing the KxK lattice with periodic boundary conditions on an AMD Phenom (tm) II X4 970 processor (without paralleling to 4 cores).

calculation number i	system size	calculation time t	real ratio t_i to t_{i-1}	theoretical ratio t_i to t_{i-1}
1	4x4	8 ms	-	-
2	5x5	77 ms	9,6	15,01
3	6x6	875 ms	11,36	11,84
4	7x7	8902 ms	10,17	9,77
5	8x8	82145 ms	9,22	9,10

The initial subsystem was the lower left spin of t. The expansion to the original system was carried out row by row, from bottom to top. Each row was completed sequentially from left to right. In accordance with the algorithm, after each addition of the spin, degeneracy matrices were recalculated.

On can estimate the speed of the algorithm. In total, K^2 steps should be carried out until the KxK system is get. Using the expansion scheme as described above (row by row, from bottom to top), the number of opened spins at each step is approximately equal to $2 \cdot K$. The upper and lower boundary spins will be opened. In accordance with the algorithm, when one adding the next spin, all possible $2^{2 \cdot K}$ configurations of opened spins should be handled. Each configuration of opened spins corresponds to a matrix of degeneracy of states in energy and magnetization. To change the matrix of degenerations it is necessary to carry out a double cycle in rows and columns. The number of matrix elements is equal to the product of all possible values of energy $(4K^2 + 1)$ and magnetization $(2K^2 + 1)$. Thus, the calculation time will be proportional to the number of lattice spins, the number of possible configurations of boundaries and the number of elements of the degeneration matrix:

$$t_{op}(K) = C_2 \cdot K^2 \cdot 2^{2K} \cdot (2K^2 + 1)(4K^2 + 1), \tag{5}$$

where C_2 is a constant defined by the implementation of the algorithm and the features of the computer architecture. So, it is now possible to calculate the time ratio with an increase the system linear size:

$$t_{op}(K+1) / t_{op}(K) = 4 \cdot \frac{(K+1)^2 \cdot (2(K+1)^2 + 1) \cdot (4(K+1)^2 + 1)}{K^2 (2K^2 + 1)(4K^2 + 1)}. \tag{6}$$

In the last column of Table 2 we can see the time ratios calculated by the formula (6). In general, they are also consistent with the real values in column 4. It can be noted that with an increase in the linear size of the lattice, the ratio of times decreases, with the exception of small systems. The discrepancy between the theoretical and real values for the small size lattices can be explained by the features of the processor architecture. From formula (6), one can conclude that the calculation time for the 9x9 spin lattice will increase by about 8.09 times compared to 8x8 and will be about 11 minutes.

5 Conclusion

The optimized algorithm for an exhaustive enumeration is presented. It was tested on a square spin lattice with periodic boundary conditions. The calculation time of the not optimized exhaustive enumeration method increases with the size of the lattice mainly as $2^{L \cdot L}$, as time for the optimized version of the algorithm time grows mostly as $2^{2 \cdot L}$.

The proposed algorithm can be easily changed under various boundary conditions and any lattice architecture.

There are many papers devoted to various probabilistic methods for calculating the Ising models [8, 9], but the question of the quality of the results obtained with their help remains opened [10]. The method of exhaustive enumeration allows to check the accuracy of probabilistic approaches.

In the future, the proposed algorithm can be improved if we reduce the used memory to store degeneration matrices and use symmetry to spin permutations. It is expected that an improved parallelized algorithm will make it possible to calculate spin lattices up to sizes 15x15 using the computational power of the Computing Center of the Far Eastern Branch of the Russian Academy of Sciences [11].

References

1. Van Vleck, J. H.: The Theory of Electric and Magnetic Susceptibilities, Oxford University Press,. 384 p. (1932)

2. Landau, L.D., Lifshitz, E.M.: Course of Theoretical Physics. Volume 5. Statistical physics, 3 edition. — Butterworth-Heinemann, 544 p. (1975)

3. Baxter, R. J.: Exactly solved models in statistical mechanics, London: Academic Press Inc. [Harcourt Brace Jovanovich Publishers] , 498 p. (1989)

4. Metropolis, N.A., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.H.: Equation of state calculation by fast computing machines // J. Chem. Phys. – V. 21. – P. 1087-1092. (1953)

5. Swendsen, R.H, Wang, J.S.: Nonuniversal critical dynamics in Monte-Carlo simulations // Phys. Rev. Lett. – -V.-58, No. 2. – P. 86-88. (1987)

6. Wolff, U.: Collective Monte-Carlo Updating fir Spin Systems // Phys. Rev. Lett. -V. 2, No. 4. - P. 361-364. (1989)

7. Wang, F., Landau, D. P.: Efficient, Multiple-Range Random Walk Algorithm to Calculate the Density of States. Phys. Rev. Lett. American Physical Society. 86 (10): 2050—2053. (2001)

8. Newman, M.E.J., Barkema, G.T.: Monte Carlo Methods in Statistical Physics, Clarendon Press, 136 p. (1999)

9. Landau, D. P., Binder, K.: A guide to Monte Carlo simulations in statistical physics, 4th ed. Cambridge University Press, 530 p. (2015)

10. Barash, L.Yu., Fadeeva, M.A., Shchur, L.N.: Control of accuracy in the Wang-Landau algorithm // Phys. Rev. E 96, 043307. (2017)

11. Computer Center of the Far Eastern Branch of the Russian Academy of Sciences, URL: http://www.ccfebras.ru/