

2D Ising Model Representation and Analysis Using Monte Carlo and Metropolis Algorithm With Less Memory Consumption

Thavappiragasam Mathialakan

February 19, 2015

Abstract

The 2D Ising model obviously deals with mass data to study the behavior of magnetization and energy related with temperature. This analysis should be worked out in best time complexity with less memory consumption. In this paper, we introduce a less memory usage implementation technique Boolean 1D-array lattice and shows the improvement in time complexity. It also analysis this studies using our C++ coding and Matlab graphs.

1 Introduction

The electron's spin and the magnetic moment associated with are taking key role in the theory of magnetism. Ferromagnetism arises when a collection of such spins conspire so that all of their magnetic moments point in the same direction, yielding a total moment. This systems generally lose their magnetism at high temperature. The magnetic properties obviously depend on temperature. Ising model is a very useful simulation that can be used to analyze how temperature effects the magnetization.

1.1 Ising Model

This model was proposed by Lenz (1920) to study the phase transition of ferromagnets at the Curie temperature. The one dimensional case was completely developed by his pupil (1925) and the two dimensional case was done by Onsager. This is a quantum model of a collection of spins that represent magnetic moments. Two dimensional lattice is represented in a 2D array that consists a set of spins. A spin in a lattice is in either up or down that points $+z$ or $-z$. The spins are considered to be interact only between the neighbors in simplest Ising model. The equation 1 express the energy of the system that is equal to

sum of the all pairs of nearest neighbour spins $\langle i, j \rangle$.

$$E = -J \sum_{\langle i, j \rangle} S_i S_j \quad (1)$$

where J is the coupling constant and $S_i = \pm 1$ are the two spin states allowed in each lattice site. The energy between a single pair gives $-J$ if two neighbouring spins in the same direction and $+J$ if they are antiparallel. The magnetization can be calculated by taking the sum of all spins (equation 2).

$$M = \sum_i S_i \quad (2)$$

The average magnetization formed by all spins is called as magnetization density. The magnetization density for a square lattice of size $N = L^2$ is given by,

$$\frac{1}{L^2} \sum_i S_i \quad (3)$$

1.2 Monte Carlo

The Ising model is simulated using Monte Carlo (MC) that is stochastic, non-deterministic and rely heavily on the use of random or pseudo-random numbers. MC simulations are very powerful simulations and versatile numerical tools used to study large systems. In the Ising model, the Metropolis algorithm analysis a huge number of states/configurations. For example, There are 2^N considerable states in $N = L^2$ spin lattice because each spin can be either of two states. In order to avoid this inconvenience, some of the samples has been chosen from them using the MC. The expectation value of an observable A can be given as,

$$\langle A \rangle = \frac{\sum_{\text{cfg}} A_{\text{cfg}} e^{-\beta E_{\text{cfg}}}}{\sum_{\text{cfg}} e^{-\beta E_{\text{cfg}}}} \quad (4)$$

where A_{cfg} is the value of A for the state or configuration **cfg**. So, given a system that has a discrete number of states, we could, using a computer, calculate A for each state and weight these values by their Boltzman factors to find the average A .

One (bad) way of using random numbers would be to randomly pick a lot of states, measure A for each of them, and weight these values of A by their Boltzman factors. We might get close to the right answer if we sampled a lot of states, but we would spend a lot of time calculating A for states that contribute very little to the final result (an Ising lattice at very high temperature is unlikely to be in the state with all spins pointing in one direction). Instead of sampling (measuring parameters like A for) a lot of states and then weighting them by their Boltzman factors, it makes more sense to choose states based on their Boltzman factors and to then weight them equally. This is known as the Metropolis algorithm, which is an importance sampling technique.

1.3 Metropolis Algorithm

This algorithm chooses a random spin and flip it from +1 to -1 or vice versa. Due to this individual spin's flip, the energy is gained or lost, and the state is going to change from one to another. This state changing is accepted based on the energy difference of the new state related to the current state. In this way, MC simulation technique is used to fulfill the algorithm, and Boltzman factor k_B is taken as 1.

Major Steps of the Algorithm

1. Set the temperature T
2. Initialize lattice
3. Do this for a given number of iterations
 - (a) Choose random spin in the lattice and flip it to opposite.
 - (b) Calculate the energy difference between the trial state and the present state, δE .
 - (c) If $\delta E \leq 0$ the trial state is accepted. Otherwise, generate a uniform random number δR in $[0, 1]$, and new state is accepted if $\delta R \leq e^{-\beta \delta E}$
 - (d) Calculate the average energy and magnetization

2 Approaches to reduce memory consumption

The lattice spin model represents set of spins in a molecular structure. A spin in a model be in one of the two states, may be up or down. This model uses a data structure 2D integer array of 1s and -1s to represent up and down spins respectively. Suppose a lattice has N spins, it has to use 32N bits of memory to keep this record. It can be reduced to N bits if we choose Boolean array. How this Boolean representation is possible? A spin is in only two possible states, so they can be represented using Boolean bi-value, true or false. The up-spin and down-spin is given by true (1) and false (0) respectively instead of 1 and -1.

This way, further of less memory usage, helps to reduce processing time using bitwise logical operations instead of time consuming arithmetic operation multiplexer (In worst case, It takes $\theta(n^2)$ to multiply two n-digit numbers). During the energy calculation, each neighbour of a randomly selected flipped-spin is taken to multiply. A spin can has two to four neighbours or connections, but every spin has four neighbours in periodic boundary conditions, where it has link between spins in row 1, n and spins in column 1, n.

The table 1 shows the energy of a spin pair $\langle S_i, S_j \rangle$ for all four possible combinations. The energy is doubled in a pair by the effect given to each other. This energy calculation requires two multiplication operations, $mul(mul(S_i, S_j), 2)$. In our Boolean lattice, spin pair is operated by bitwise XOR and energy is chosen to -2 or 2 based on the truth value (table 2). It reduces the time complexity significantly. When we are thinking in low level

Integer Representation		
S_i	S_j	Energy
-1	-1	2
-1	1	-2
1	-1	-2
1	1	2

Table 1: The energy of a spin pair $\langle S_i, S_j \rangle$ in integer representation

Boolean Representation			
S_i	S_j	XOR	Energy
0	0	0	2
0	1	1	-2
1	0	1	-2
1	1	0	2

Table 2: The energy of a spin pair $\langle S_i, S_j \rangle$ in Boolean representation

architecture or if we have plan to design a specific electronic controller to solve this task, deal with bitwise operation, it will support easy and very low budget project because such operations can be performed using less logic gates.

How to calculate the magnetization in bit/Boolean array? Suppose the lattice of size $N = L^2$ has k up-spins, the magnetization will be calculated by the equation 5. In our representation, up-spins are 1 and down-spins are 0, sum of the bit array gives the number of up-spins, so the magnetization is given by the equation 6.

$$M = 2k - L^2 \quad (5)$$

$$M = 2 \sum_i S_i - L^2 \quad (6)$$

3 Implementation

The high-level programming language C is the most preferable language to deal with mass data, and C++ is an advanced version of C. So, we chose C to implement this ising model using Metropolis algorithm as described above. Since 1D array offers better memory locality and less allocation and deallocation overhead, It is faster way than 2D for dense matrices. As the reason given above, the data structure 1D Boolean array is chosen to keep spins in lattice.

4 Results and Discussion

In our testing case, a lattice of size $N = 10^2$ is taken, and initially it is configured with all up; a 1D Boolean array of size 100 is set to true. We applied both boundary conditions (BC): 1) Every spin in all four edges of lattice has no round robin connections. 2) Periodic BC. When using the first BC, initial value of average energy and magnetization is given by $-2L(L + 1)$ and 1.0 ($= L/L$) respectively. The energy will be $2N$ and the same magnetization for second BC. The average energy and magnetization are calculated for every 0.1 temperature increases in range $[0, 5]$. This task is processed for 10 000 iterations. The figure 1 shows the changes on magnetization with temperature. It decreases when temperature increases, and we found the critical temperature, T_c is 2.3 approximately. The energy is a continuous function of temperature, and increases as a function of temperature (Figure 2).

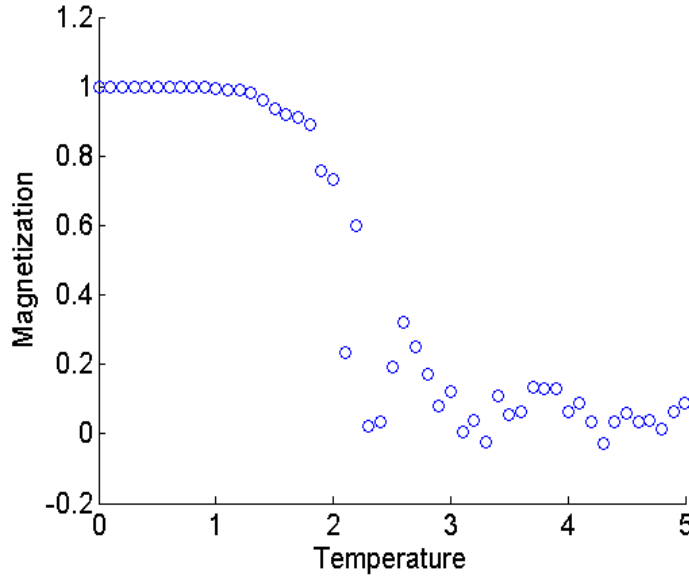


Figure 1: The change of Magnetization with increasing temperature when $J = 1$ and $k_B = 1$.

Finally, for the fixed temperature 2.3, the configuration is changed through 1000 iterations, and magnetization is measured and graphed as shown in the Figure 3. This falls on the linear line $y = -0.0003x + 0.9573$. And also the Figure 4 shows the energy related to configurations. The linearity of this relationship is given by the equation $y = 0.0005x - 1.6353$.

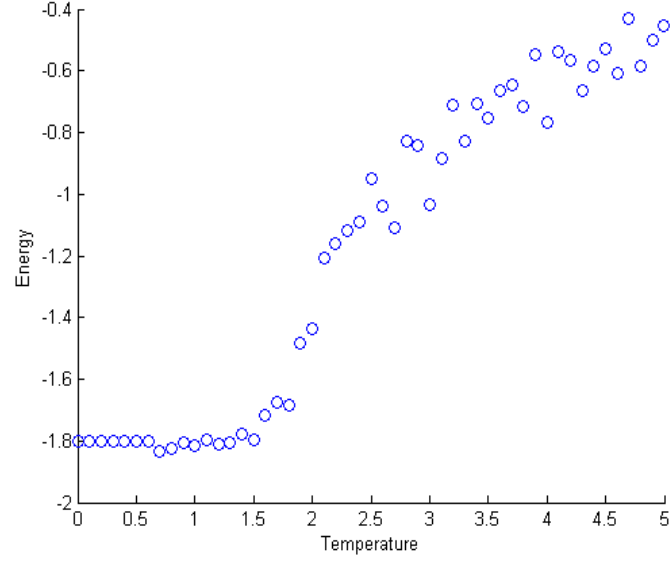


Figure 2: The change of Energy with increasing temperature when $J = 1$ and $k_B = 1$.

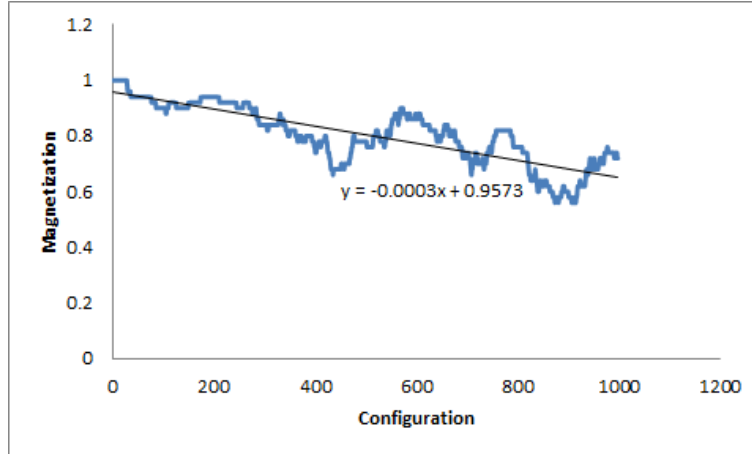


Figure 3: The change of Magnetization related with configurations at temperature 2.3

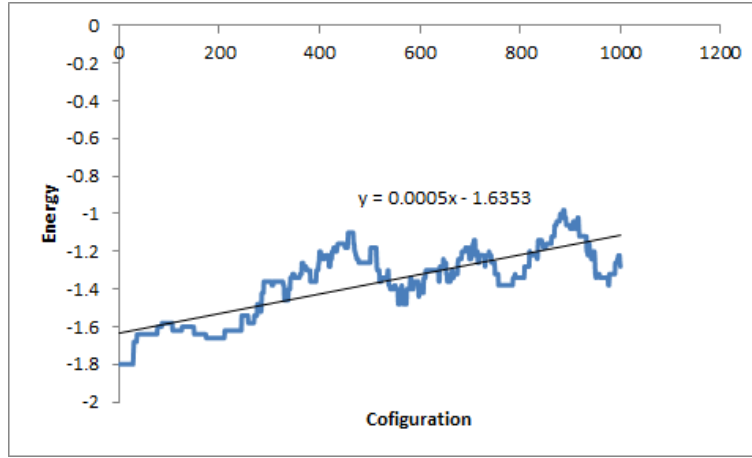


Figure 4: The change of Energy related with configurations at temperature 2.3.

5 Conclusion

We tested our approach successfully using minimum memory and time. The magnetization decreases with increasing temperature, but in contrast, average energy increases with it. Further, even though those measurements fluctuating for the configuration changes, It can be brought into a linear trend.

A Periodic Boundary Condition

The figures 5 and 6 clearly show the distinguishes between periodic boundary condition and without it.

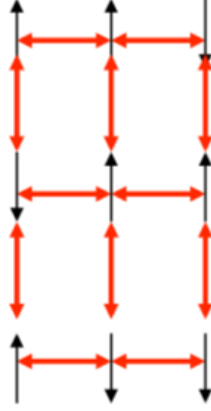


Figure 5: The links in 3x3 lattice without PBC.

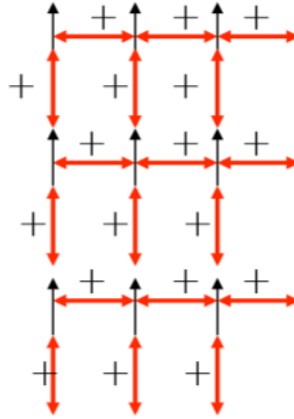


Figure 6: The links in 3x3 lattice with PBC.

We also tested our implementation of ising model for PBC, the magnetization and energy changes related with temperature are shown in figure 7 and 8 respectively.

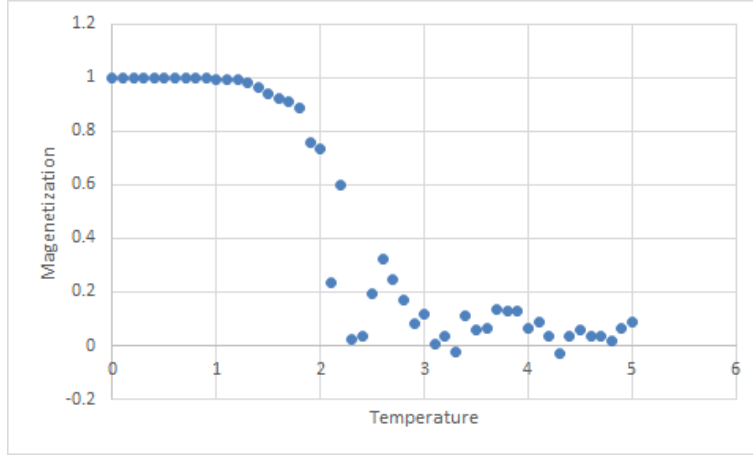


Figure 7: The change of Magnetization with increasing temperature when $J = 1$ and $k_B = 1$ using PBC

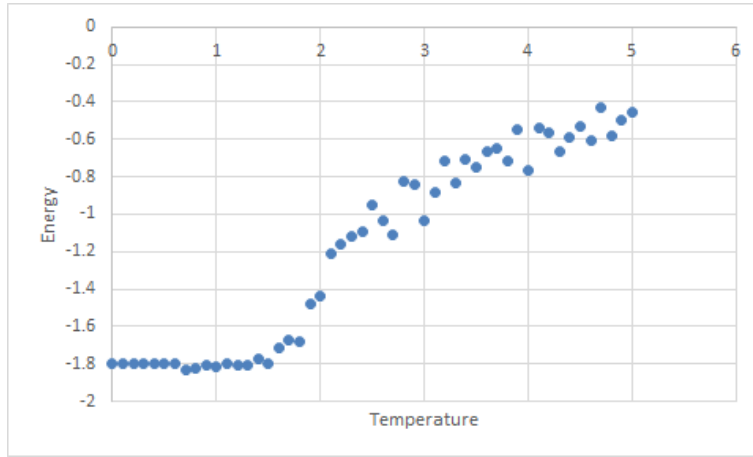


Figure 8: The change of Energy with increasing temperature when $J = 1$ and $k_B = 1$ using PBC