Monte Carlo Method on 2D Ising Model

20165030 Kim, Seongsu

March 11, 2022

Abstract

The paper include the contents of the basic implementation of Monte Carlo methods for 2D Ising model and the physical and mathematical backgrounds for statistical mechanics. The paper is based on the [NB99], [Yeo92], [LB09] and [BH21]. The programs for simulating algorithms are mainly written in C++ and some of them python. The data analysis of algorithms are produced by matplotlib in python.

1 Monte Carlo Methods for Statistical Physics

1.1 What is Monte Carlo methods?

The Monte Carlo method is one of the well-known numerical calculation methods that use simulation of randomness. For most problems from statistical physics, the degree of freedom of the system is too enormous so that we usually can't calculate or simulate every possibility of the system. To work out the limitation of computational power and time, we adequately choose the sample from the distribution of the physical states and calculate and study the physical property of system.

1.2 The principle of equilibrium thermal Monte Carlo simulation

1.2.1 The expectation and estimator

In statistics, the expectation is defined as the integral of multiple of quantity and probability distribution. For the probability distribution $p(x)$, the expectation of quantity $Q(x)$ from $p(x)$ is

$$\langle Q \rangle = \int Q(x)p(x)dx \approx \frac{1}{N} \sum_{x \sim p(x)} Q(x). \tag{1}$$

where $x \sim p(x)$ means x that follows $p(x)$ and N is number of samples. So if we know the distribution of x, we can discretely choose the sample following $p(x)$ and find the estimator of $\langle Q \rangle$. Also, we can change above the expectation to sample it from the another probability distribution $q(x)$:

$$\langle Q \rangle = \int Q(x)\frac{p(x)}{q(x)}q(x)dx \approx \frac{1}{N} \sum_{x \sim q(x)} Q(x)\frac{p(x)}{q(x)}. \tag{2}$$

Thus, if we want to choose the $Q(x)$ from uniform distribution, we choose N samples from uniform and estimator can be calculated by averaging the $Q(x)$ from samples. But this is not quite good approach since the finite number of N samples may be hard to show the property of the original distribution.

For example, the distribution p_x in the Ising model follows the Boltzmann distribution which exponentially small for most of the region of x. Therefore, if we choose the finite number of x uniformly, then it is almost selected from the exponentially small region and contribute the small number of estimator. In principle, if we can choose the sufficiently large number from the sample, then estimator fits with real expectation value. But it is hard to compute in that way by the limitation of time.

The good strategy for the Monte Carlo method is finding methods of sampling that follows the real probability distribution p_x, so that result can account the property of the system from sufficiently small number of samples which takes reasonable computation time.

1.2.2 The strategy for applying Monte Carlo methods to statistical mechanics

To simulate the statistical physical phenomena by Monte Carlo method, there should be some assumptions to make it calculable. If we use the rejection method to select the samples from Boltzmann distribution, it takes enormous times to pick N samples since almost x will be rejected.

Thus, we need a new method to generate the distribution - we use the transition of the states like the physical states of the system varies by time passing. So, if we knows every transition probability from state μ to the other states ν at time t, we can estimate which state will be next. To make it possible to compute the model, we can assume transition probability is time-independent and dependent only to current states. And also introducing the concept of the acceptance ratio, we choose the next state and determine to transit to next state or not by the transition probability.

1.2.3 Markov process

One of the mathematical assumption is Markov process. The Markov process assumes that transition probability to current state μ to ν, $P(\mu \to \nu)$, is independent to the time and preceding steps of states but only dependent to current state and the probable state.

And the sum of transition probability from μ to every possible state ν also satisfy the constraint

$$\sum_{\nu} P(\mu \to \nu) = 1. \tag{3}$$

The succeeding process of the Markov process is called Markov chains. To study the equilibrium states of the system, the probability distribution of each states p_μ should follow the Boltzmann distribution after some steps of Markov chains.

1.2.4 Ergodicity

The philosophy of ergodicity is simple - we can access every possible states by Markov process in principle. To account the physical phenomena successfully, although we can't easily access to the less probable states by transition process, we must be able to access to every possible states.

1.2.5 Detailed balance

To study the equilibrium states of the system, we need to argue that what is equilibrium. The equilibrium is the system such that system seems to stay remain same. So the amounts of net changes of the system is 0 although there are flows of inward and outward.

In the view of equilibrium of states is the net amounts of the change of probability distribution is 0. Let the weight of finding state μ is $_\mu$, and the derivation of w_μ by the time is

$$\frac{dw_\mu}{dt} = \sum_{\nu} \left[w_\nu P(\nu \to \mu) - w_\mu P(\mu \to \nu) \right], \tag{4}$$

and this is called master equation. The equation implies that amount of changes of weight w_μ by time is sum of inward and outward flow of states.

In equilibrium, the master equation should be 0 for every states when w_μ reach p_μ (which implies the weight of each states w_μ reaches to target distribution p_μ) :

$$\sum_{\nu} \left[p_\nu P(\nu \to \mu) - p_\mu P(\mu \to \nu) \right] = 0, \tag{5}$$

or

$$p_\mu = \sum_{\nu} \left[p_\nu P(\nu \to \mu) \right]. \tag{6}$$

This condition is called balance condtion.

How we confirm that w_μ goes to p_μ is also important topic. Firstly, the approximate representation of weight by time development is

$$w_\mu(t+1) \approx w_\mu(t) + \frac{dw_\mu}{dt} = \sum_{\nu} w_\nu(t) P(\nu \to \mu). \tag{7}$$

In matrix notation, this becomes

$$\mathbf{w}(t+1) = \mathbf{P} \cdot \mathbf{w}(t), \tag{8}$$

and assume time t goes to infinity, then we have

$$\mathbf{w}(\infty) = \mathbf{P} \cdot \mathbf{w}(\infty). \tag{9}$$

However, it is also able to find the process to reach a dynamic equilibrium. This kinds of rotation is called a limit cycle. In the case $w(\infty)$ would satisfy

$$\mathbf{w}(\infty) = \mathbf{P}^n \cdot \mathbf{w}(\infty). \tag{10}$$

where n is the length of the limit cycle. This limit cycle makes $\mathbf{w}$ hard to converge to the desired distribution $\mathbf{p}$. By applying the strong constraint which is called the detailed balance, we can eliminate the limit cycle. The detailed balance is the condition to our transitional probabilities:

$$p_\mu P(\mu \to \nu) = p_\nu P(\nu \to \mu). \tag{11}$$

This condition is the subset of the balance condition (6). It can be easily found by applying summation of μ to both sides. The detailed condition says that overall rate of change from one state μ to state ν is same with reversal rate that changed from μ to ν. So that the when t goes to infinity, there is no cyclic pattern[1].

From (10), we can see that largest eigenvalue of the Markov matrix P is 1. It can be generalized to the any stochastic (transitional) matrix.

Theorem. *The largest eigenvalue of a transitional matrix is 1.*

Proof. Firstly, for a transitional matrix $\mathbf{A}$, $\mathbf{A1} = \mathbf{1}$ where $\mathbf{1} = [1, \cdots, 1]^T$ since each row of A sums to 1. As a result, 1 is always eigenvalue of A. Secondly, suppose that there exists $\lambda > 1$ and nonzero x that satisfies $\mathbf{Ax} = \lambda\mathbf{x}$. Let m be a largest element of x. Each row of $\mathbf{Ax}$ satisfies $\sum_i A_{ij}x_i = \lambda x_i$. Since the sum of each row of $\mathbf{A}$ is 1, $\sum_i A_{ij} = 1$ and $0 < A_{ij} < 1$. So that $\sum_i A_{ij}x_i < m$ should be satisfied, but $\lambda x_i < m$ is contradiction when $x_i = m$ since $\lambda > 1$. So $\lambda <= 1$ and the largest eigenvalue should be 1. □

And we also can find matrix representation of (6) is

$$\mathbf{p} = \mathbf{P} \cdot \mathbf{p}. \tag{12}$$

In a nutshell, with the detailed balance condition, the Markov chain will converge to the equilibrium distribution as we want. More discussion is represented below box, but not rigorous.

From (11) and applying p_μ to Boltzmann distribution, then we have

$$\frac{P(\mu \to \nu)}{P(\nu \to \mu)} = \frac{p_\nu}{p_\mu} = \frac{e^{-\beta E_\nu}}{e^{-\beta E_\mu}} = e^{-\beta(E_\nu - E_\mu)}. \tag{13}$$

Note that we cannot easily assume that $P(\mu \to \nu) = A\exp[-\beta(E_\nu - E_\mu)/2]$ where A is constant, since $0 \le P \le 1$ so that $\beta(E_\nu - E_\mu) \le 2\ln(1/A)$. But this constraint is not good choice since it is hard to keep satisfying condition with $\sum_\mu P(\mu \to \nu) = 1$. What we can easily determine is just the ratio of the transitional probability.

[1] This limit cyclic concept is denoted in [NB99], but rigorous mathematical description is missing and I couldn't find related concepts.

Brief explanation of how w converge to p
From above equation (8) we can easily induce

$$\mathbf{w}(t) = \mathbf{P}^t \cdot \mathbf{w}(0), \tag{14}$$

where $\mathbf{w}(0)$ is the vector representation of initial weight of the states and $\mathbf{w}(t)$ is weight of the states at time t.

And using the fact that the eigenvectors of the transition matrix $\mathbf{P}$ can be the basis of the vector space of $\mathbf{w} \in Z^N$, $\mathbf{w}(0)$ can be written as the linear combination of eigenvectors:

$$\mathbf{w}(0) = \sum_i a_i \mathbf{v}_i, \tag{15}$$

where $\mathbf{v}_i$ is the eigenvector of $\mathbf{P}$. From (14) and (15), we have

$$\mathbf{w}(t) = \sum_i a_i \mathbf{P}^t \cdot \mathbf{v_i} = \sum_i a_i \lambda_i^t \mathbf{v}_i, \tag{16}$$

where λ_i is eigenvalue of $\mathbf{v}_i$. Since the largest eigenvalue of transition matrix is 1, only $\lambda_i = 1$ survives after long step of t and the other eigenvalue should be shrunken exponentially. Eventually, the $\mathbf{w}(t)$ will be proportional to $\mathbf{p}$.

This explanation is incomplete, since there are at least two eigenvectors whose eigenvalue is 1 ($\mathbf{p}$ and $\mathbf{1} = 1, \cdots, 1^T$). So it is hard to assert that $\mathbf{w}$ will be converge to $\mathbf{p}$ by above explanation. It should be fixed.

1.2.6 Acceptance ratio

With the Markov process and detailed balance, we get the desired set of probability distribution of the states. However, it is not enough to simulate the Monte Carlo methods. This is because we couldn't determine the exact translation probability $P(\mu \to \nu)$. If we knows all translation probability, we can choose the states and generate or calculate all possible next states and weights of each states. However, we only know the ratios of the transitional probability so that we have to deal with them.

To get to the point, we can break the transition probability down into the two part:

$$P(\mu \to \nu) = g(\mu \to \nu) A(\mu \to \nu), \tag{17}$$

where $g(\mu \to \nu)$ is selection probability, which is the probability to generate new state ν from the current state μ by the algorithms, and $A(\mu \to \nu)$ is acceptance ratio. The acceptance ratio says that if we start off in a state μ and our algorithm generates a new state ν from it, we should accept that state and change our system to the new state ν a fraction of the time $A(\mu \to \nu)$.

And we can fix above (13) as

$$\frac{P(\mu \to \nu)}{P(\nu \to \mu)} = \frac{g(\mu \to \nu) A(\mu \to \nu)}{g(\nu \to \mu) A(\nu \to \mu)} = e^{-\beta(E_\nu - E_\mu)}. \tag{18}$$

In order to create Monte Carlo algorithm as we considered before, we have to make up an algorithms which generates random new states ν from an old μ with some set of probabilities $g(\mu \to \nu)$, and we accept or reject those states with acceptance ratios $A(\mu \to \nu)$ which we choose to satisfy Equation (18). With all these concept, we can make initial probability of the states converge to the target distribution, Boltzmann distribution.

2 Introduction of the 2D Ising model

2.1 2D Ising model

The Ising model is the model that mimic the magnets which are dominated by the magnetic dipole moments. Each dipole moments can be in one of two states ($+1$ or -1).

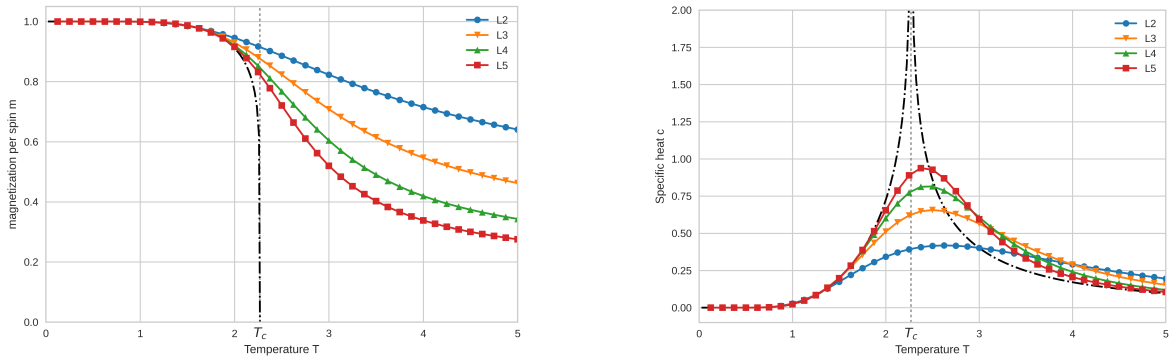

Figure 1: The magnetization and specific heat per spin of 2D square Ising model. There are L = 2,3,4,5 cases and exact solution by Onsager(dash).

The Hamiltonian of the 2D Ising model can be represented as

$$H(\mu) = -J \sum_{\langle ij \rangle} s_i s_j - H \sum_{\{s_i \in \mu\}} s_i, \tag{19}$$

where the first term denotes the interaction between all nearest neighbors (represented by a sum over $< i, j >$), and the second term accounts for the interaction of each site with an external magnetic field H, favoring the $s_i = +1$ state if $H > 0$.

For H = 0, there exist two ground states at $T = 0$, namely all spins up or all spins down. If $J < 0$, the system is called antiferromagnetic, because antiparallel spins lead to an energy minimization. No interaction occurs for $J = 0$. In the case of a ferromanetic Ising system, the first term in the Hamiltonian tries to create order by minimizing the overall energy as as consequence of aligning spins in the same direction. The second term tends to align the spins in the direction of the external field H.

2.2 Exact solution and Naive simulation

The exact solution of 2D Ising model was solved by Lars Onsager in 1944.

The critical temperature T_c is

$$\sinh \frac{2J}{k_B T_c} = 1 \tag{20}$$

or,

$$\frac{k_B T_c}{J} = \frac{2}{\ln\left(1+\sqrt{2}\right)} \approx 2.269. \tag{21}$$

And the magnetization M is

$$M = \left[1 - \sinh(2\beta J)^{-4}\right]^{\frac{1}{8}}. \tag{22}$$

Should be added more information about Onsager solution.

The **Naive simulation** in this paper implies that the full calculation of (Ising) model, so that calculate all possible states and exact partition function. When implementing this calculation, I used up-down symmetry and helical boundary condition(a.k.a HBC). By using up-down symmetry, the total calculation time and the number of iteration are reduced by half. And the magnetization is calculated as expectation of absolute of magnetization($\langle |m| \rangle$).

From the Figure12, the exact solution of the magnetization and specific heat of Ising model has extreme value at T_c. The magnetization steeply falls down to 0 and specific heat diverge to infinity.

This kinds of situations are related to phase transition. Particularly, the magnetization m being almost 1 before T_c and 0 after that. This phenomena mimics the ferromagnetic phase transition. However, The naive simulation of magnetization and specific heat is quite different to exact solution. Especially, there are no steep falls and divergence. This is simply because the phase transition arise on infinity lattice situation. So that when the number of the lattice increasing, also the result become more similar to the exact solution although 5 is not enough to depict phase transition.

But the problem is that "Naive simulation takes enormous times which grows as $O(2^{L^2})$"[2]. When calculating $L = 5$ simulation, it took about 40 seconds. So if we want to calculate $L = 100$, it will takes about 250 powers more - $40^{250} \approx 10^{400}$ - that cannot be calculated until suns blowing up. To solve this kinds of problem, we can introduce the Monte Carlo methods.

3 Metropolis Algorithms: single-spin update

The Metropolis(-Hastings) algorithms is one of the simplest and well known example of Monte Carlo simulation. The Metropolis single-spin updates algorithms is following this procedure: First, (randomly or not) select a spin from lattice. Second, calculate the energy(Hamiltonian) difference between before and after flips. Finally, according to acceptance ratio, determine whether flip the selected spin or not. Repeat above again and again until it reach to the proposed number of iteration.

From the Metropolis, we can consider the many general concept of the Monte Carlo calculation such as equilibrium, measurement of the expectation values, and the calculation of errors.

3.1 Backgrounds of Algorithms

The Metropolis algorithms also have to obey the principles of equilibrium such as Markov process, ergodicity and detailed balance with acceptance ratio.

In the Metropolis Algorithms, the selection probability $g(\mu \to \nu)$ for each state is equal. To be clear, let there are N spin in lattices, then there are N possible states(ν_i) that can be accessed from the previous stats(μ). Then the selection probability to each N possible states is

$$g(\mu \to \nu_i) = \frac{1}{N}, \quad (i = 1, \cdots, N), \tag{23}$$

and the selection probability to other non-accessible states is 0.[3]

With these selection probability to the possible states, the condition of detailed balance, equation (18) takes from

$$\frac{P(\mu \to \nu)}{P(\nu \to \mu)} = \frac{g(\mu \to \nu)A(\mu \to \nu)}{g(\nu \to \mu)A(\nu \to \mu)} = \frac{A(\mu \to \nu)}{A(\nu \to \mu)} = e^{-\beta(E_\nu - E_\mu)}. \tag{24}$$

And even the states that cannot access by the one step, the above transition probability is conserved.

This part can be imperfect since I couldn't find the references.

The selection probability is basically 0 for the states that cannot be access by one flip. So if we find the detailed balance condition to the state μ and the non-accessible state ν, then (18) can't be defined.

To solve this problem, I suggest advanced selection probability $\mathcal{G}(\mu \to \nu)$ this model. In the Metropolis algorithms, when considering the state transition μ to ν we have to consider not just one flip but have to consider every possible path to reach ν regardless the number of flips. So the advanced selection probability consider the every possible path of transition. Let the $g'(\mu \to \mathcal{P} \to \nu)$ means that selection probability to along path $\mathcal{P} = \alpha_1, \cdots, \alpha_{M-1}$ where $M > 1$, then we can find

$$g'(\mu \to \mathcal{P} \to \nu) = g(\mu \to \alpha_1)g(\alpha_1 \to \alpha_2) \cdots g(\alpha_{N-1} \to \nu) = \frac{1}{N^M}. \tag{25}$$

[2] L is the number of lattice for each edge.

[3] The meaning of non-accessible states implies that the states that cannot access by one flip step.

The advanced selection probability $\mathcal{G}(\mu \to \nu)$ include the every possible path:

$$\mathcal{G}(\mu \to \nu) = \sum_{\mathcal{P} \in A} g'(\mu \to mathcalP \to \nu). \tag{26}$$

The ratio of advanced selection probability μ to ν and ν to μ is

$$\frac{\mathcal{G}(\mu \to \nu)}{\mathcal{G}(\nu \to \mu)} = \frac{\sum_{\mathcal{P} \in A} g'(\mu \to \mathcal{P} \to \nu)}{\sum_{\mathcal{P}^* \in A^*} g'(\nu \to \mathcal{P}^* \to \mu)}. \tag{27}$$

And we can find every path $\mathcal{P} \in A$ and the inverse path $\mathcal{P}^* \in A^*$, the selection probability of the ptha and its inverse path satisfies

$$g'(\mu \to \mathcal{P} \to \nu) = g'(\nu \to \mathcal{P}^* \to \mu). \tag{28}$$

As a result,

$$\frac{\mathcal{G}(\mu \to \nu)}{\mathcal{G}(\nu \to \mu)} = 1, \tag{29}$$

and the (24) is generally accepted.

And we can choose the acceptance ratios $A(\mu \to \nu)$ to satisfies the detailed balance condition

$$A(\mu \to \nu) = \max(1, e^{-\beta(E_\nu - E_\mu)}), \tag{30}$$

then the algorithms satisfies the detailed balance completely which means that the model can depict equilibrium states.

Additionally, the ergodicity is easily satisfied since we can find the transition path of single flips for every states.

3.2 Implementation

In the implementation of the algorithms is obvious since this is well-known exercise. So I specify some special attributes of my own codes.

In the Ising models, the spin lattice should be specified. Since we are dealing with the simple square Ising model, we can use n×n matrices as the data set of spin lattice. But multi dimension array is less efficient than single array, so I rather choose to use the n^2 single dimension array.

In the process of Monte Carlo step, we need random number processing to describe the probability. There are many examples of pseudo random number generators in references. Among them, I used the `mt19937` from cpp basic `<random>` library which generate number based on **Mersenne Twister**. we can mimic the probability by generating the uniformly distributed random number between 0 and 1 and compare it to probability p and accept the change when the random number is below the p.

And each of the Metropolis step, we have to select a site to flip from the spin configuration first. There are majorly two ways of selecting the sites. The first one is choosing sites randomly from the spin configuration. It can be easily determined by using uniform random number generator between 0 to n^2. And the second one is choosing sites with patterns like selecting sites from 1 to N subsequently. But actually selecting the sequential lattice is not good way of selecting since the vary next spin will be easily influenced by the spin flip of previous one. So I used chess-board style for choosing the sites, so that select the sites one blocks away. By using this kinds of sequential selecting skill, we can uniformly select the spins so that it takes less time to converge to equilibrium.[4]

Then, we have to find the energy difference of current state and next state. In the mathematical

[4]In the next chapter, I found that the chess-board like styles for choosing sites has less autocorrelation step size.

notation, this is

$$E_\nu - E_\mu = -J \sum_{\langle ij \rangle} s_i^\nu s_j^\nu + J \sum_{\langle ij \rangle} s_i^\mu s_j^\mu \tag{31}$$

$$= -J \sum_{\text{i n.n to k}} s_i^\mu (s_k^\mu - s_k^\nu) \tag{32}$$

$$= -J \sum_{\text{i n.n to k}} 2 s_i^\mu s_k^\mu. \tag{33}$$

Therefore, we can find the energy difference of states by multiplying the value of spin and the summation of nearest neighbor of selected spin.

When calculating nearest neighbor, I choose the HBC at the boundary of the spin configuration. The HBC has some advantage that is easy to implement and converging fast. My C++ style code snippet is below:

```
1000    //Helical boundary condition
        //sc: n*n spin configuration
1002    //N: the number of lattices
        //L: the length of lattice edge
1004    //i: the index of the selected spin site
        int nn, sum = 0;
1006    int XNN = 1, YNN = L;

1008    if((nn = i - XNN) < 0) nn += N;
        sum += sc[nn];
1010    if((nn = i + XNN) >= N) nn -= N;
        sum += sc[nn];
1012    if((nn = i - YNN) < 0) nn += N;
        sum += sc[nn];
1014    if((nn = i + YNN) >= N) nn -= N;
        sum += sc[nn];
1016
        return sum;
```

Finally, I used the objected oriented style code since when we evaluate or measure the result, maybe multiple number of model would be used for cross-validation or ensemble evaluation.

3.3 Equilibrium

To see the model converge to the equilibrium, the spin configuration of the result is plotted below Figure 2, 3 and 4.

From the figures, we can intuitively see the whole configuration converge to the equilibrium. To be specific, as the number of Monte Carlo steps(mcs) is grown and reached to specific step, then the configuration looks like in the equilibrium states.

But these kinds of evaluation on eyes is not good method to quantitatively measure the property. To solve the problem, we can introduce the (auto)correlation time to evaluate how long does it takes to converge to equilibrium. As discussed in the 1.2.5, we can conjecture that there will be exponential property.

The autocorrelation function which is used for evaluating the speed of model convergence is

$$\chi(t) = \int dt' [m(t') - \langle m(t') \rangle] \times [m(t'+t) - \langle m(t'+t) \rangle], \tag{34}$$

or, in discrete format,

$$\chi(t) = \sum_{t'}^{t_{max}} \langle (m(t') - \langle m(t') \rangle) \times (m(t+t') - \langle m(t+t') \rangle \rangle . \tag{35}$$

The $(m(t') - \langle m(t') \rangle)$ term inside the equation implies that the consideration of pure fluctuation without the overall effect ($\langle m(t) \rangle$) at t'. So $(m(t'+t) - \langle m(t'+t) \rangle)$ implies the pure fluctuation at

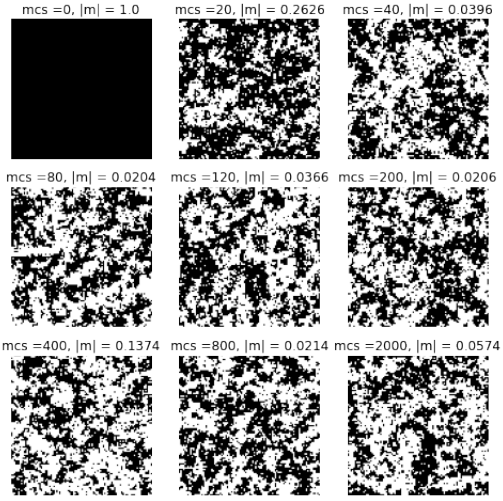

Figure 2: The equilibrium result of Metropolis algorithms at $T_c < T = 2.7$ on 100×100 lattice. From the top-left figure to bottom-right figure, the number of mcs is increasing. When mcs bigger than 200, the spin configuration seems to being equilibrium which means that statistical property of the spin configuration is conserved. Since T is bigger than T_c, whole configuration should be random so that expectation of magnetization would be almost 0.

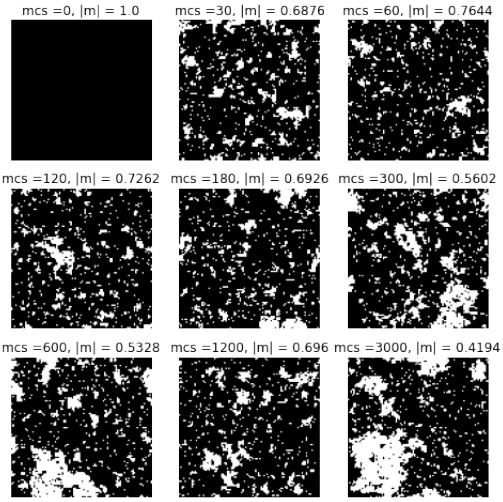

Figure 3: The equilibrium result of Metropolis algorithms at $T_c \approx T = 2.29$ on 100×100 lattice. When mcs bigger than 300, the spin configuration seems to being equilibrium. Since T is near T_c, configuration seems to have local clusters.

$t' + t$. The whole meaning of the above equation is the summation of multiple of pure fluctuations whose gap of step is t. If the $m(t')$ and $m(t + t')$ is unrelated, than the multiple of pure fluctuation will be almost random and the summation will be 0. However, if the $m(t')$ and $m(t + t')$ are related, than the multiple of pure fluctuation would be biased. The summation will be positive if the $m(t')$ and $m(t + t')$ align to same direction, and it will be negative if they align to opposite direction.

From (35), we can develop this as below:

$$\chi(t) = \sum_{t'}^{t_{max}} \left[\langle m(t')m(t + t')\rangle - \langle m(t')\rangle \langle m(t + t')\rangle\right]. \tag{36}$$

[5]

[5]The $\chi(0)$ is the same to the magnetic susceptibility χ_m in definition.

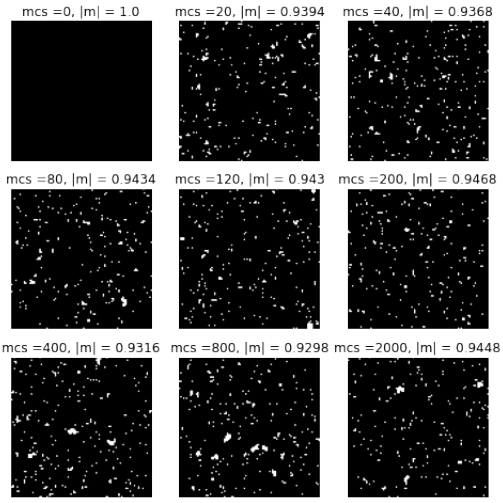

Figure 4: The equilibrium result of Metropolis algorithms at $T = 1.9 < T_c$ on 100×100 lattice. When mcs bigger than 200, the spin configuration seems to being equilibrium. Since T is smaller than T_c, configuration seems globally clusters so that expectation of magnetization would be almost 1.

There are two way to evaluate above autocorrelation function in [NB99]. First is setting the expectation of magnetization as

$$m_{exp} = \langle m(t') \rangle = \langle m(t + t') \rangle \,, \tag{37}$$

and second is set

$$\langle m(t') \rangle = \frac{1}{t'} \sum_{t'=0}^{t'} m(t) \tag{38}$$

thus $\langle m(t') \rangle$ and $\langle m(t + t') \rangle$ are different. I named first way as χ_1 measurement and second way as χ_2 measurement.

The autocorrelation is expected to fall off exponentially at long times (or steps) thus:

$$\chi(t) \sim e^{-t/\tau}. \tag{39}$$

With above definition, when time $t = \tau$ the autocorrelation is a factor of $1/e$ down from its down from its maximum value at $t = 0$. So that if we want independent samples, we may want to draw them at intervals of greater than one correlation time τ. Most common statistical independence turns out to be samples drawn at intervals of 2τ[6].

From the Figure 5, the magnetization which starts in all spins-up states seems to fluctuate based on the $\langle m \rangle \approx 0$.[7]

From the Figure 6, we can see the magnetization autocorrelation $\chi(t)$ variations. As the time is growing, the autocorrelation decay almost exponentially until 800 mcs steps.

The dashed line is the result of the $\chi(t) \approx e^{-t\tau}$ fitting from 0 to 150 steps.

There is an alternative way to calculate τ, that is calculating the integrated correlation time. If we assume that (39) is correct, then we have

$$\frac{\chi(t)}{\chi(0)} = \int e^{-t/\tau} dt = \tau. \tag{40}$$

From the integrated correlation time, we can reduce the bundles of calculation for τ. Especially, using FFT algorithms we can reduce the $O(n^2)$ calculation to $O(n)$ calculation.

[6] [NB99] Chapter 3.3, page. 60

[7] I thought it looks like the gap between the peaks of fluctuation has special patterns. But if the gap is really intrinsically equally spaced, then we can find that, the interval of equally spacing D, $\chi(nD)$ should be positive and large for integer n. However, there was no clear patterns of $\chi(nD)$

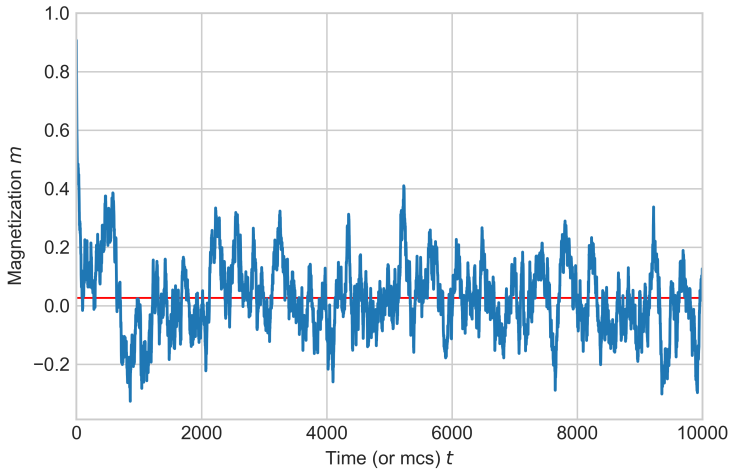

Figure 5: The magnetization autocorrelation function $\chi(t)$ for a 2D Ising model at $T = 2.4$ on a square lattice of 100×100 sites with $J = 1$ simulated using the Metropolis algorithms. The $\langle m \rangle \approx 0.02678$. Time is measured in mcs and maximum mcs is 10000.

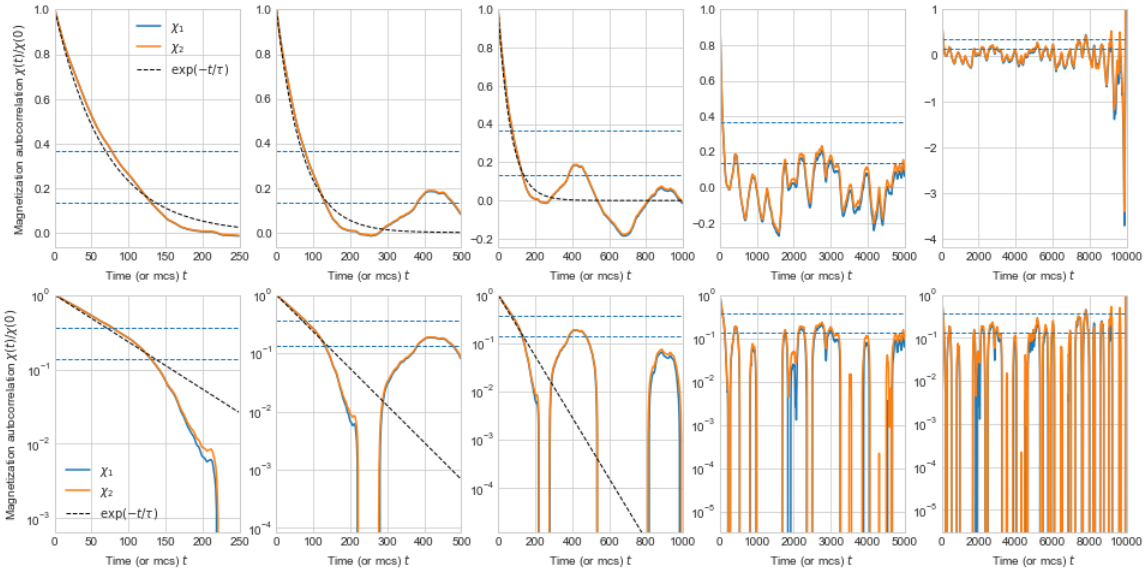

Figure 6: The magnetization autocorrelation function $\chi(t)$ at $T = 2.4$ on a square lattice of 100×100 sites with $J = 1$ simulated using the Metropolis algorithms. The magnetization m of this result is Figure 5. The spin configuration starts at $T = 0$ (all-spins up) and the time is measured in mcs and maximum mcs is 10000. The periodic boudary condition is used and the chess-board style sequential selecting was chosen for Metropolis algorithms. The blue line is measured in the way of (37) and the orange line is in the way of (38). The dashed line is the exponential fitting line between 0 to 250 steps. The correlation time $\tau \approx 68.61$.

The autocorrelation time is significantly higher near T_c than other region, this kinds of behaviors is called critical slowing down. The autocorrelation time can be differed by models and algorithms. Since we want the independent model so that we need to calculate and simulate more and more when autocorrelation time is large.

3.3.1 Fast Fourier Transformation

The Fast Fourier transformation (FFT) is the algorithm that conducts discrete Fourier transform and its inverse efficiently. The basic Fourier transformation requires $O(n^2)$ calculation but the FFT needs

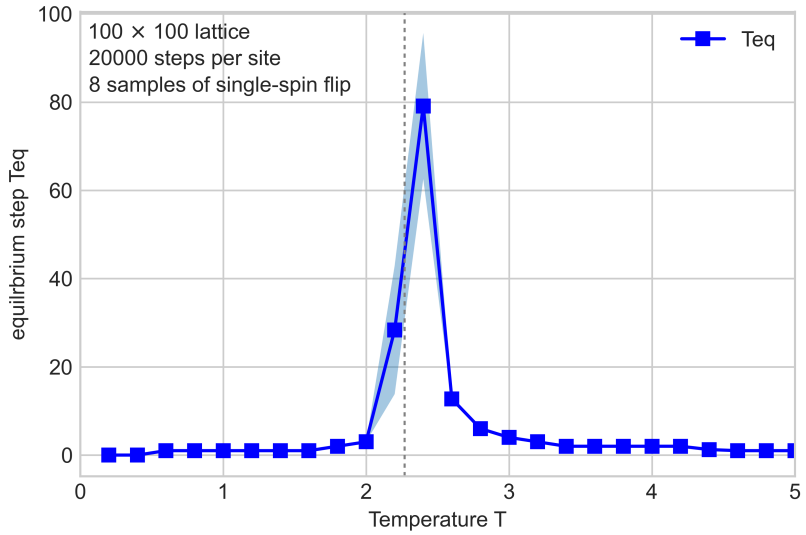

Figure 7: The autocorrelation time averaged by 8 samples with same conditions to 6 but change the temperatures. The autocorrelation function is calculated by FFT and iFFT technique which described in 3.3.1. The autocorrelation time near T_c is become significantly large than the other region.

only $O(n \log n)$ calculation.

The Fourier Transform is the transformation of arbitrary input function $f(t)$ to summation of the periodic function $(\cos, \sin)$ $F(\omega)$:

$$f(t) = \int F(\omega) e^{i2\pi\omega t} d\omega, \tag{41}$$

$$F\{f(t)\} = F(\omega) = \int f(t) e^{-i2\pi\omega t} dt, \tag{42}$$

and where frequency $\omega = 1/T$.

By using Fourier Transform to autocorrelation function, we have

$$\tilde{\chi}(\omega) = \int dt \cdot e^{-i2\pi\omega t} \chi(t) \tag{43}$$

$$= \int dt \cdot e^{i2\pi\omega t} \left[\int dt' [m(t') - \langle m(t') \rangle] \times [m(t'+t) - \langle m(t'+t) \rangle] \right] \tag{44}$$

$$= \int dt \cdot e^{-i2\pi\omega(t+t')} [m(t'+t) - \langle m(t'+t) \rangle] \int dt' \cdot e^{i2\pi\omega t'} [m(t') - \langle m(t') \rangle] \tag{45}$$

$$= \tilde{m}'(\omega) \cdot \tilde{m}'(-\omega) = |\tilde{m}'(\omega)|^2, \tag{46}$$

where $m'(t) = m(t) - \langle m(t) \rangle$.

There are two ways of exploiting above relation. First one is using the inverse Fourier Transform. From above equation, we can get original autocorrelation function $\chi(t)$:

$$\chi(t) = \int d\omega \cdot e^{i2\pi\omega t} \tilde{\chi}(\omega) \tag{47}$$

$$= \int d\omega \cdot e^{i2\pi\omega t} |\tilde{m}'(\omega)|^2. \tag{48}$$

We can easily implement above relation as codes by using FFT and inverse FFT from standard library. And second one is using the integrated correlation time. If we put $\omega = 0$ in the above equation, we can get the integrated correlation time form:

$$\tilde{\chi}(0) = \int dt \cdot e^0 \chi(t) = \int dt \chi(0) e^{-t/\tau} = \chi(0)\tau, \tag{49}$$

or

$$\frac{\tilde{\chi}(0)}{\chi(0)} = \tau. \tag{50}$$

But this way is not good way to measure τ, since zeroth coefficient of Fourier representation is unstable and will be 0 if the length of data set is finite.

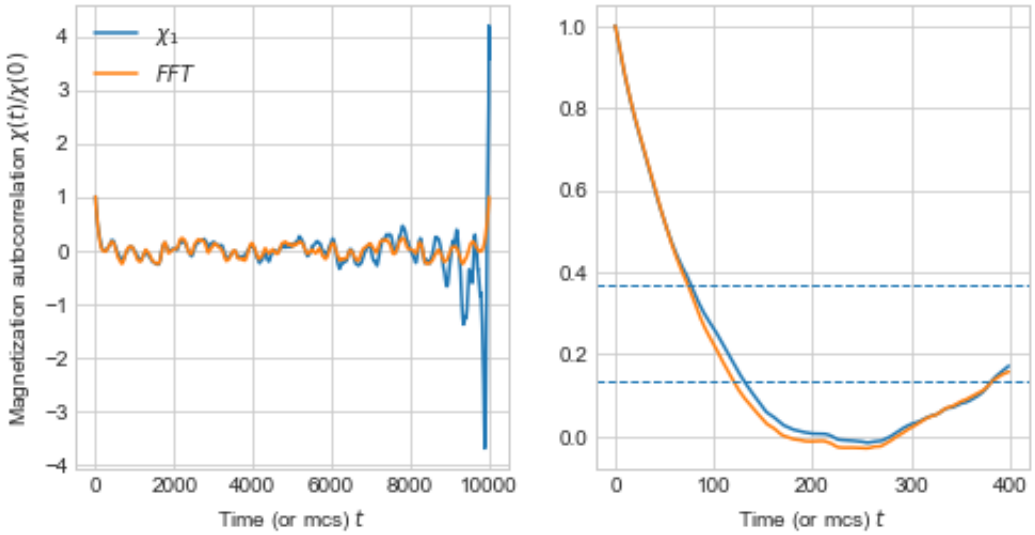

Figure 8: The FFT transform result and comparison with χ_1 of Figure 5. The left is plotted in range of mcs 0 to maximum, and the right is plotted in range of mcs 0 to 400.

From Figure 8, we can notice that autocorrelation function calculated by FFT is accurately fitted with the original $\chi(t)$. The FFT results seems to differ with the original when the mcs step goes to maximum. But at least mcs near the τ, the behavior seems to almost same. So we can conclude FFT is a quite good method to calculate the autocorrelation function and save the computational time at the same time.

3.4 Measurements

When the measuring the physical values from the model, we have to consider the correlation time so that we can use the independent result.

First, the magnetization and specific heat of 2D Ising model is can be easily calculated by Metropolis algorithms. The expectation value of some quantity which is measured by the just one spin configuration is the average value among whole measurements:

$$\langle Q \rangle = \frac{1}{N} \sum_{i=1}^{N} Q_i. \tag{51}$$

In this sense, the magnetization (per sites) is

$$\langle m \rangle = \frac{1}{N} \sum_{i=1}^{N} m_i, \tag{52}$$

which is the average value of total number of measurements N. But the specific heat is quite different since it is related with expectation value so that it needs multiple set of measurements. The specific heat per site is

$$c = \frac{k\beta^2}{N}(\langle E^2 \rangle - \langle E \rangle^2), \tag{53}$$

so that it requires to calculate expectation of E^2 and E. The magnetic susceptibility is also similar to c:

$$\chi = N\beta(\langle m^2 \rangle - \langle m \rangle^2). \tag{54}$$

In principles, the measurements have to use the independent data set, which means each measurement should away more than τ. But this is difficult to accounts since it is hard to find the *tau* before calculating since it need

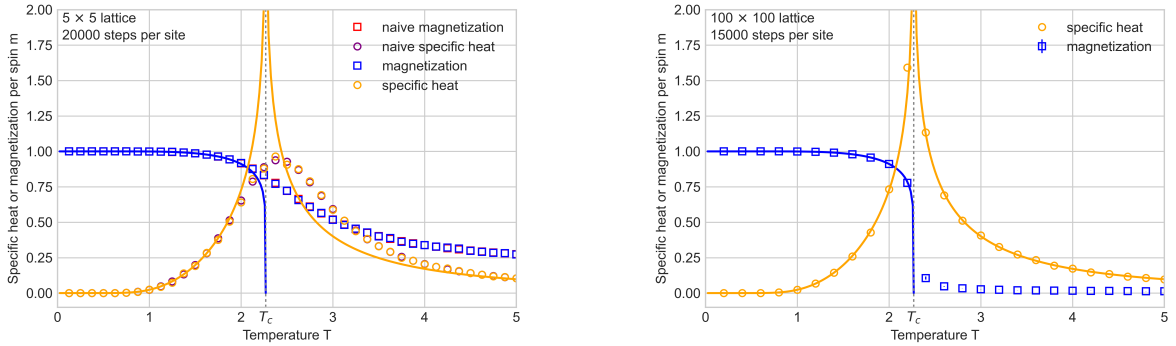

Figure 9: The magnetization and specific heat per spin of 2D square Ising model. The spin configuration starts at $T = 0$, J = 1 and it is simulated using the Metropolis algorithms. The left panel is 5×5 lattice results which is compared to Naive result Figure 12. The right pane is 100×100 lattice results. The solid line is the exact solution by Onsager.

By the Metropolis Algorithms, the total calculation time is significantly decreased and getting great estimation at the same time.

3.4.1 Error Analysis

The error of statistical fluctuation is can be accounted by the standard deviation:

$$\sigma = \sqrt{\frac{\frac{1}{n}\sum_{i=1}^{n}(q_i - \bar{q})^2}{n-1}} = \sqrt{\frac{1}{n-1}(\bar{q^2} - \bar{q}^2)} \tag{55}$$

where n is a number of the measurements, q_i is the ith measurements of q and $\bar{q}$ is expectation of q. By this standard deviation error, I calculated the error of Figure 7.

In the same manner, we can calculate error of magnetization m since each m_i can be measured independently. But when we account the real independent samples, it is good way to use n as $n = t_{max}/2\tau$ (Muller-Krumbhaar and Binder 1973).

So the magnetization error is simply

$$\sigma = \sqrt{\frac{2\tau}{t_{max}}(\bar{m^2} - \bar{m}^2)}. \tag{56}$$

But when we try to measure the error of specific heat, it cannot be easily represented just like magnetization. As shown in (54), It is related to the expectation value, so it needs to multiple of data sets.

There are several ways to calculate the those kinds of error. One of the method is called Jackknife method. The jackknife method is just like below. First, slice the measurements of data to M bins. Second, calculate the desiring physical value from the data set excepting ith data bin and let this value be c_i. And repeat this to 1st bin to Mth bin. Then the standard deviation of the c is

$$\sigma = \sqrt{\sum_{i=1}^{M}(c_i - c)^2}. \tag{57}$$

From the Figure 10, we can notice that error of measurements is larger near the T_c. This is related to critical slowing down. And to find out the accuracy of Jackknife error, I also implemented ensemble error of the Metropolis simulation.

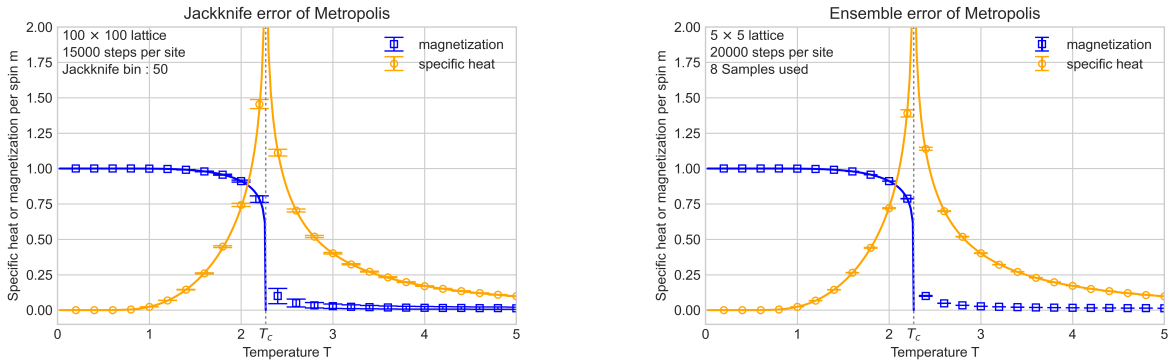

Figure 10: The error plotting result of Figure 9. The left one has error which is plotted by Jackknife method, and the right one has error which is plotted by ensemble error of 8 different random seed sample.

4 Wolff Cluster Algorithms

4.1 Equilibrium

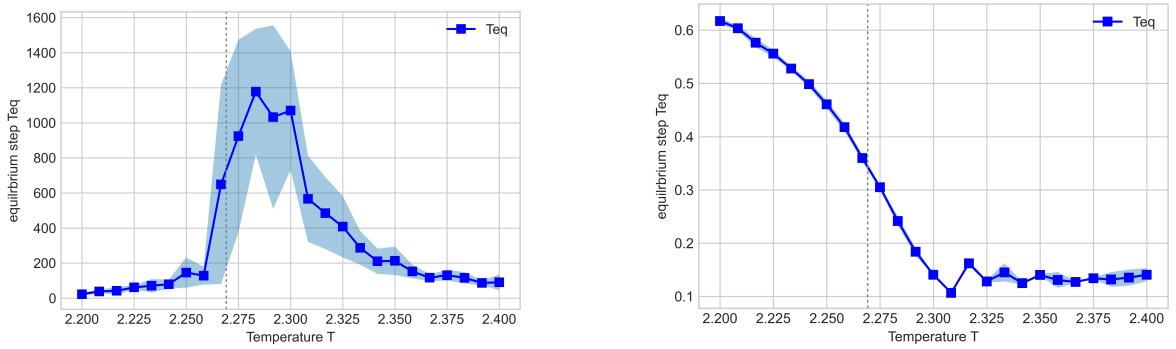

Figure 11: The autocorrelation step plots near the T_c is presented. The left one is autocorrelation step of Metropolis algorithm and right one is of Wolff algorithms. Near the T_c, Metropolis has enormous autocorrelation time due to critical slowing down, but Wolff has much less effects of critical slowing down. It needs almost less than 1 mcs for equilibrium.

4.2 Measurements

5 Finite Size scaling

A Backgrounds of Thermodynamics

A.1 Partition function

Equilibrium statistical mechanics is based upon the idea of a partition function. The general form for the partition function for a classical system is

$$Z = \sum_{\mu} e^{-E_\mu / k_B T}, \tag{58}$$

where E_μ is the energy for the system μ, T is the temperature and k_B is Boltzmann constant. Usually denote $\beta^{-1} = k_B T$ in the inverse unit of energy. The partition function is the sum of the all possibility of possible states of the system.

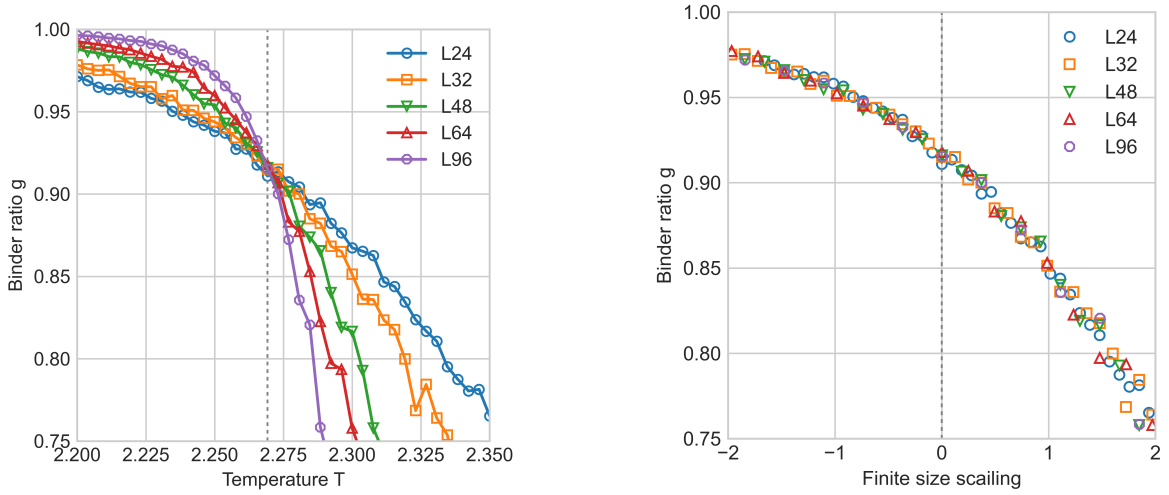

Figure 12: Binder ratio plotting by Wolff clustering

The possibility of each state p_μ is

$$p_\mu = \frac{1}{Z} e^{-\beta E_\mu}. \tag{59}$$

where $\sum_\mu p_\mu = 1$.

In Ising model which we will handle soon, the energy of the system E_μ can be represented as the Hamiltonian H of the system. This is a kind of approximation such that the dominating part of the energy E_μ is H suggested by the specific model, so that we can know the property of the statistical system by calculating dominant part of Energy, H.

A.2 Expectation

In thermodynamics, the expectation value of some quantity Q is

$$\langle Q \rangle = \frac{1}{Z} \sum_\mu . \tag{60}$$

This is quite intuitive when we think about the definition of expectation in statistics. But in the real situation such as measurement from experiments, above definition of expectation is not suitable since we can't access to every possible states in right time.

Therefore, the practical meaning of the expectation value is the average value of quantity from time-varying system. During a finite time difference, the state of the system will change (or stay remain) by time and the average value of Q among the system variations can be approximated expectation value.

A.3 Internal energy and Specific heat

The internal energy of the system is the expectation of the energy of the system, that is

$$U = \langle E \rangle = \frac{1}{Z} \sum_\mu E_\mu e^{-\beta E_\mu}. \tag{61}$$

From (61) we can rewrite equation in terms of derivation, that is

$$U = -\frac{1}{Z} \frac{\partial}{\partial \beta} Z = -\frac{\partial \log Z}{\partial \beta}. \tag{62}$$

The specific heat is important since it can represent the fluctuation(or deviation) of internal energy.

The specific heat is defined as the variation of internal energy over time:

$$C = \frac{\partial U}{\partial T} = \frac{\partial \beta}{\partial T}\frac{\partial U}{\partial \beta} = -\frac{1}{k_B T^2}\left[-\frac{1}{Z}\sum_\mu E_\mu^2 e^{-\beta E_\mu} + \left(\frac{1}{Z}\sum_\mu E_\mu e^{-\beta E_\mu}\right)^2\right] \tag{63}$$

$$= k_B\beta^2\left[\left\langle E^2\right\rangle - \left\langle E\right\rangle^2\right] = k_B\beta^2\frac{\partial^2 \log Z}{\partial \beta^2}. \tag{64}$$

The phase transition of the system is closely related to the fluctuation of order parameter. This is mainly because the phase transition arise at the discontinuity of derivation of free energy at critical temperature T_c. There are first order and second order phase transition according to order of derivation of discontinuity. The specific heat is the derivative of the internal energy, we can study the phase transition by inspecting discontinuous property of the specific heat.

A.4 Free energy and Entropy

The (Helmholtz) free energy is defined as

$$\mathcal{F} = U - TS, \tag{65}$$

where $U = \langle E\rangle$ is internal energy of the system, T is temperature of the system, and the S is entropy of the system. The entropy of the system can be represented in terms of the specific heat (when the volume V is constant):

$$C = \frac{\partial U}{\partial T} = T\frac{\partial S}{\partial T} = -\beta\frac{\partial S}{\partial \beta} = k_B\beta^2\frac{\partial^2 \log Z}{\partial \beta^2}. \tag{66}$$

By integrating the terms by β, we have the entropy in terms of the partition function Z:

$$S = -k_B\beta\frac{\partial \log Z}{\partial \beta} + k\log Z. \tag{67}$$

From (65) and (67), we can also rewrite free energy in term of Z:

$$\mathcal{F} = \frac{\log Z}{\beta} = \beta^{-1}\log Z \tag{68}$$

Thus, by finding the partition function of the system, we can find $\mathcal{F}$, U, S and C.

A.5 Conjugated variables and Susceptibility

Conjugated variables is the intensive variables that coupled with the specific extensive variable. The external variable is the variable related to the size of the system or derived from external factor (such as external magnetic field B, volume of the matter V, and etc..) and the intensive variable is the variable independent to the size of the system (such as magnetization M, pressure of the system p ...). The conjugated variables can be accounted in terms of the hamiltonian. For an arbitrary conjugated variable X and the coupled extensive variable Y, the hamiltonian includes $-X_\mu Y$ term (and maybe $-XY$ is the only dependent term to Y in Hamiltonian):

$$H_\mu = -X_\mu Y + (\cdots) \tag{69}$$

$$\frac{\partial H}{\partial Y} = -X_\mu. \tag{70}$$

Thus, we can represent the expectation X in terms of derivation by Y:

$$\langle X\rangle = \frac{1}{Z}\sum_\mu X_\mu e^{-\beta H_\mu} = -\frac{1}{\beta Z}\frac{\partial Z}{\partial Y} = -\frac{1}{\beta}\frac{\partial \log Z}{\partial Y} = -\frac{\partial \mathcal{F}}{\partial Y} \tag{71}$$

Susceptibility can be understood as the general physical definition of the magnetic susceptibility $\chi = \frac{\partial M}{\partial B}$. The susceptibility is the response of the $\langle X\rangle$ to Y:

$$\chi_X = \frac{\partial \langle X\rangle}{\partial Y}. \tag{72}$$

The LHS of (72) is related to the fluctuation of X:

$$\frac{\partial \langle X \rangle}{\partial Y} = \frac{1}{\beta} \frac{\partial^2 \log Z}{\partial Y^2} = \beta \left[\langle X^2 \rangle - \langle X \rangle^2 \right] \tag{73}$$

So we can calculate the susceptibility of the system by calculating the fluctuation of the variable. The susceptibility is related to the second derivation of the free energy $\mathcal{F} = \beta^{-1} \log Z$, we can also study the phase transitional property of the system from susceptibility.

A.6 General susceptibility and Correlation function

The generalized definition of susceptibility is also related to the conjugated value X and the coupled field Y, but the difference is the Y can be differd by the site. In terms of hamiltonian, H includes $-\sum_i X_i Y_i$ where i indicates the location among sites. The general susceptibility is the response of X_i to the Y_i:

$$\chi_{ij} = \frac{\partial \langle X_i \rangle}{\partial Y_j} = \frac{1}{\beta} \frac{\partial^2 \log Z}{\partial Y_j \partial Y_i}. \tag{74}$$

By mathematical development we can find the relation,

$$\chi_{ij} = \beta \langle (X_i - \langle X_i \rangle) \cdot (X_j - \langle X_j \rangle) \rangle = \beta \left[\langle X_i X_j \rangle - \langle X_i \rangle \langle X_j \rangle \right] = \beta G^2(X_i, X_j), \tag{75}$$

where $G^2(X_i, X_j)$ is the correlation function of the X_i and X_j. The correlation function indicates the response of parameter variation of one site to the other site. The term $(X_i - \langle X_i \rangle)$ indicate the difference of value of X_i to external influence of X_i ($\langle X_i \rangle$), so it is pure variation of X_i. Thus multiplying $(X_i - \langle X_i \rangle)$ and $(X_j - \langle X_j \rangle)$ means finding the multiple of pure variation of the two distinct sites. If the two sites vary same way, then the expectation of pure variation will be positive. But if they vary opposite way, the expectation will be negative. If they vary in randomly (or irreverently), the expectation will be near 0. By studying the correlation function, we can find the phase transitional property such as correlation length.

References

[BH21] Lucas. Böttcher and HANS J. Herrmann. Computational statistical physics. 2021.

[LB09] David P. Landau and Kurt Binder. A guide to monte-carlo simulations in statistical physics. 2009.

[NB99] M.E.J. Newman and G.T. Barkema. Monte carlo methods in statistical physics. 1999.

[Yeo92] J.M. Yeomans. Statistical mechanics of phase transition. 1992.