Master's Thesis Updates

Monte Carlo Simulations of Ising Model

Kunal Vema

5th Year BS-MS Physics Major

Under the supervision of

Prof. Vijay B. Shenoy

April 2, 2023

Contents

Chapter 1

Monte Carlo simulations for spin systems

The purpose of this chapter is to give a brief introduction to Monte Carlo simulations and their applications in the numerical analysis of phase transitions in statistical physics models. We will start by illustrating the general idea of importance sampling Monte Carlo methods, followed by the different type of update schemes (Metropolis, Wolff-cluster, etc.) to sample non-uniform distributions efficiently. We will also pay attention to the topic of statistical analysis of the generated data via autocorrelation analysis and statistical error analysis. For illustration purposes, we will primarily focus our attention on the simplest spin model, i.e., the two-dimensional Ising model (without external field) and discuss the finite-temperature critical phenomena and phase transition. Finally, we will demonstrate the calculation of the critical exponents of the two-dimensional Ising model using finite-size scaling analysis of relevant physical observables. Although we start with a classical statistical model, these ideas can be generalized and are applicable in simulations of quantum-spin systems, where they become the *Quantum Monte Carlo* (QMC) methods [Janke, Sandvik].

1.1 Introduction to Monte Carlo methods

Monte Carlo simulations are an important and broad class of stochastic methods that utilize randomness to solve deterministic problems efficiently. To show their utility, we will start with a simple and illustrative example. Consider the calculation of a thermal expectation value of an observable $Q(x)$ in statistical physics;

$$\langle Q \rangle = \int_0^L \mathrm{d}x\, \rho(x) Q(x), \qquad \int_0^L \mathrm{d}x\, \rho(x) = 1, \tag{1.1.1}$$

where $\rho(x)$ is the underlying probability distribution.

The most naive way to estimate this integral numerically is to use *Euler's method* to discretize the integration range into N pieces and perform a discretized sum

$$\langle Q \rangle \approx \Delta x \sum_{j=1}^{N} \rho(x_0 + j\Delta x)\, Q(x_0 + j\Delta x) \tag{1.1.2}$$

where $\Delta x = L/N$. Such grid-based methods are very accurate for low-dimensional integrals, but as we go to higher-dimensional integrals, both the error-scaling and computational costs grow significantly.

A more efficient method for performing high-dimensional integrals is known as *Monte Carlo integration* where the points are uniformly sampled across the integration range instead of a grid. If the uniformly sampled random points are denoted by the set $\{x_1, x_2, x_3, \ldots, x_N\}$, then the integral in Eq. (1.1.1) can be estimated as

$$\langle Q \rangle \approx \frac{L}{N} \sum_{i=1}^{N} \rho(x_i)\, Q(x_i) \tag{1.1.3}$$

The error in a Monte Carlo estimate goes as $1/\sqrt{N}$ in any number of dimensions, and hence it is more efficient for estimating $6\mathcal{N}$-dimensional integrals characteristic of statistical mechanics of $\mathcal{N}$ particles. However, this straightforward unbiased Monte Carlo integration only works well in practice if the integrand isn't sharply peaked in some small region of the integration range.

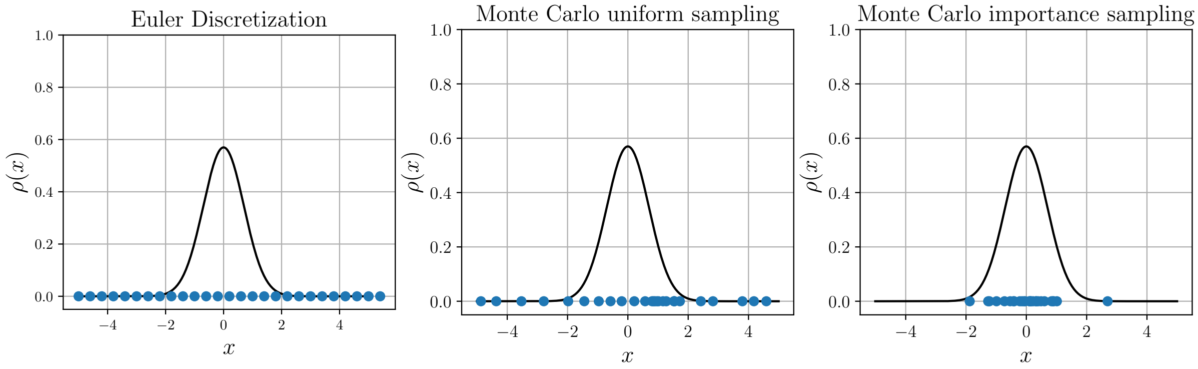


Figure 1.1: Comparision of sampling techniques.

If $\rho(x)$ is sharply peaked in a small region, then uniform sampling of points can result in large statistical fluctuations as only a small fraction of points will fall within the dominant range. Therefore, the next improvement can be performed by sampling the points according to a probability distribution $W(x)$ which is peaked in the same region as $\rho(x)$. This gives the estimate of the expectation value as

$$\langle Q \rangle \approx \frac{L}{N} \sum_{i=1}^{N} \frac{\rho(x_i)}{W(x_i)} Q(x_i)\, W(x_i) \approx \frac{1}{N} \sum_{i=1}^{N}{}^{(W)} \frac{\rho(x_i)}{W(x_i)}\, Q(x_i) \tag{1.1.4}$$

where $\sum^{(W)}$ denotes points being sampled from the distribution $W(x)$, and we write $L\sum W(x_i) \to \sum^{(W)}$. Often, a good solution is to use $W(x) = \rho(x)$, and the expectation value becomes an arithmetic average of $Q(x)$ over the configurations sampled by $\rho(x)$

$$\langle Q \rangle \approx \frac{1}{N} \sum_{i=1}^{N}{}^{(\rho)} Q(x_i) \tag{1.1.5}$$

This technique is known as the *Monte Carlo Importance Sampling* method since we are only sampling the points lying in the "important" region of the probability distribution $\rho(x)$.

In statistical mechanics, the probability distribution is generally the Boltzmann distribution $\rho(\vec{x}, \vec{p}) = e^{-\beta H(\vec{x}, \vec{p})}$, and we can use Monte Carlo importance sampling to calculate expectation values of physical observables. However, in the above discussion, we have ignored the problem of sampling points according to a given probability distribution. We discuss this in the following subsection.

1.1.1 Markov Chains and Detailed Balance condition

In order to calculate integrals via the method of importance sampling, we need a way to sample points according to the probability distribution $\rho(x)$. The theory of Markov Chains provides us with the necessary tools to generate a Markov Chain process which evolves towards a desired equilibrium distribution.

In physicists' language, a Markov chain is a discrete chain of events $C_1 \to C_2 \to C_3 \to \ldots \to C_N$ that evolves stochastically and satisfies the Markovian property, i.e., the probability of $C_{i-1} \to C_i$ transition is independent of its history. Put together, this implies the probability of obtaining the above sequence is

$$P(C_1 \to C_2 \to \ldots \to C_N) = P(C_1) \cdot P(C_2|C_1) \cdot P(C_3|C_2) \ldots P(C_N|C_{N-1}) \tag{1.1.6}$$

Roughly speaking, if the Markov chain doesn't repeat itself and can reach *any* configuration starting from *any other* configuration, then it is *ergodic* and settles onto a stationary distribution. By designing an appropriate Markov chain, it is possible to obtain any desired stationary distribution $\rho(C)$.

Let us now assume we have a set of all possible configurations $\{X\} = \{X_1, X_2, \ldots, X_n\}$ in the configuration space. Assume we start with some configuration $X_{i(0)}$ and stochastically generate a Markov chain $X_{i(1)}, X_{i(2)}, X_{i(3)}, \ldots, X_{i(M)}$. We can do the same for an ensemble of configurations initially distributed according to $\rho(X)$. At the update 0 , the number of configurations X_i in the initial ensemble is $N_0(X_i) \propto \rho(X_i) \Rightarrow N_0(X_i) = \mathcal{N}\rho(X_i)$. The given Markov chain must have an update scheme which stochastically evolves the ensemble to the next set of states. The number of configurations after the update 1 is

$$N_1(X_i) = N_0(X_i) + \sum_{j \neq i} \left[N_0(X_j)\, P(X_j \to X_i) - N_0(C_i)\, P(X_i \to X_j) \right] \tag{1.1.7}$$

The first term in the sum represents configurations updating into X_i and the second term represents X_i updating out to other configurations. If we want the ensemble to remain distributed according to the initial ensemble distribution $\rho(X)$, then, for all $i = 1, 2, \ldots M$,

$$\sum_{j \neq i} \left[\rho(X_j)\, P(X_j \to X_i) - \rho(C_i)\, P(X_i \to X_j) \right] \overset{!}{=} 0 \tag{1.1.8}$$

One possible solution of this condition is to satisfy it term-by-term $\forall\ j$

$$\rho(X_j)\, P(X_j \to X_i) - \rho(C_i)\, P(X_i \to X_j) \overset{!}{=} 0 \tag{1.1.9}$$

which can be written as a ratio

$$\frac{P(X_j \to X_i)}{P(X_i \to X_j)} = \frac{\rho(X_i)}{\rho(X_j)}, \tag{1.1.10}$$

also known as the *detailed balance condition.* Although here we start with an ensemble distributed according to the probability distribution ρ, for practical purposes, neither do we need to start from the same distribution, nor do we require an ensemble of configurations. In practice, the master equation Eq. (1.1.7) takes care of the excess or deficiency and equilibrates after a characteristic *equilibration time* of Markov chain updates.

The transition probability $P(X_i \to X_j)$ can further be written as a product of the update proposal and the proposal acceptance probabilities

$$P(X_i \to X_j) = \underbrace{\mathcal{A}(X_i \to X_j)}_{\text{propose } X_i \to X_j} \cdot \underbrace{P_{\text{accept}}(X_i \to X_j)}_{\text{accept } X_i \to X_j} \tag{1.1.11}$$

Since the proposal probabilities should be uniform $\mathcal{A}(X_i \to X_j) = \text{constant}$, the detailed balance condition in terms of acceptance probabilities becomes

$$\frac{P_{\text{accept}}(X_j \to X_i)}{P_{\text{accept}}(X_i \to X_j)} = \frac{\rho(X_i)}{\rho(X_j)} \tag{1.1.12}$$

Starting from the detailed balance condition, one can define stochastic algorithms, such as Monte Carlo simulations, which generate configurations according to a desired distribution. One such common algorithm is the *Metropolis algorithm* with the solution to Eq. (1.1.12) as

$$P_{\text{accept}}(X_i \to X_j) = \min\left[\frac{\rho(X_j)}{\rho(X_i)}, 1\right] \tag{1.1.13}$$

For statistical mechanics, the desired equilibrium distribution is the Boltzmann distribution which gives rise to the well-known Metropolis acceptance probability

$$P_{\text{accept}}(E_i \to E_j) = \min\left[e^{-\beta(E_j - E_i)}, 1\right]. \tag{1.1.14}$$

1.2 Metropolis and the Ising model

As discussed in the last section, we can generate samples distributed according to the Boltzmann distribution if we choose the acceptance probability as defined in Eq. (1.1.14) for the Markov chain. These drawn configurations can then be utilized to calculate thermal expectation values via importance sampling.

We will discuss this simulation method in the context of the 2-dimensional ferromagnetic Ising model. The Ising model is the paradigmatic model for systems exhibiting continuous phase transition from a ferromagnetic to a paramagnetic phase at a critical temperature T_c. In the absence of an external field ($h = 0$) and only nearest-neighbor interactions, the energy of the ferromagnetic Ising model is

$$E = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j \tag{1.2.1}$$

where $\langle i, j \rangle$ indicates nearest-neighbors i, j and the expression for a 2-dimensional square lattice can be more suggestively written as

$$E = -J \sum_{i} \sigma_{i_x,i_y} \left[\sigma_{i_x,i_y+1} + \sigma_{i_x+1,i_y}\right] \tag{1.2.2}$$

For a Markov chain transition $\sigma_i \to -\sigma_i$, the difference in energy between the configurations is given by

$$\Delta E = 2J\, \sigma_{i_x,i_y} \left[\sigma_{i_x,i_y+1} + \sigma_{i_x+1,i_y} + \sigma_{i_x,i_y-1} + \sigma_{i_x-1,i_y}\right] \tag{1.2.3}$$

This leads to the Metropolis acceptance probability for the Ising model spin flips being given by

$$P_{\text{accept}}(\sigma_i \to -\sigma_i) = \min\left[e^{-\beta \Delta E}, 1\right] \tag{1.2.4}$$

with ΔE being defined by Eq. (1.2.3). In a nutshell, a *Metropolis Monte Carlo* simulation of the Ising model consists of performing such spin flip proposals by selecting the spin σ_i to be flipped at random and generating a Markov chain distributed according to the Boltzmann distribution. A single *Monte Carlo sweep* then consists of L^2 such spin flip proposals so that roughly each lattice site gets an equal chance to flip, where L denotes the side length of the square lattice. As a step-by-step procedure, a Monte Carlo sweep is defined as follows:

1. Randomly choose a spin σ_i on the lattice.

2. Propose the spin flip $\sigma_i \to -\sigma_i$.
3. Calculate the difference in energy $\Delta E = E_{\text{final}} - E_{\text{initial}}$.
4. Accept the proposed move with a probability of $\min\left[e^{-\beta\Delta E}, 1\right]$.
5. Steps 1 to 4 are then repeated $\mathcal{N} = L^2$ times.

The Boltzmann sampling performed using Markov Chain Monte Carlo (MCMC) can then be utilized in calculating expectation values of observables as simple statistical averages over the sampled points

$$\langle \mathcal{O} \rangle \approx \frac{1}{N} \sum_{\{\sigma\}} \mathcal{O}(\sigma) \tag{1.2.5}$$

where we have suppressed the superscript $^{(\rho)}$ on the sum denoting sampling over the distribution $\rho(\{\sigma\}) = e^{-\beta E(\{\sigma\})}$. Note that such Monte Carlo averages are always supposed to be taken over *equilibrated* configurations in a Monte Carlo simulation. As discussed earlier, the master equation (1.1.7) ensures the equilibration of the Markov chain's probability distribution after a few steps by taking care of appropriate excess or deficiencies during the initial steps. This initial equilibration period where no measurements are performed is known as the *equilibration time* or the *thermalization time*, often denoted by t_{eq}.

The Metropolis MCMC method is thus a conceptually simple and quite universally applicable algorithm. Its applications range from simulations of simple Ising chains to highly non-trivial spin systems. However, the Metropolis algorithm comes with its fair share of problems which we will discuss later. In the next section, we will discuss the relevant statistical physics of the Ising model and the measurement of physical observables.

1.3 Physical observables and derived quantities

To set the scenery for calculating observable expectation values, we will first begin by discussing the relevant physical observables for the Ising model. Since we are dealing with spin systems, a natural quantity of interest is the magnetization, which acts as the *order parameter* of the Ising phase transition at $T_c \neq 0$. The magnetization per site is defined by the observable

$$m \equiv \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \sigma_i \tag{1.3.1}$$

where $\mathcal{N} = L^2$ is the total number of lattice sites. One can explicitly check that an operation which rotates all spins $\sigma_i \to -\sigma_i \ \forall\, i$ is a symmetry of the Ising Hamiltonian (Eq. (1.2.1)). Although we are technically not yet dealing with quantum mechanics, this operation can be denoted as $F = \prod_{i=1}^{N} X_i$ where X_i denotes a flip of spin at site i. Since applying this operation twice is equivalent to identity ($F^2 = \mathbb{1}$), we often say that the Ising model possesses a global $\mathbb{Z}_2$ symmetry.

As a result of this global $\mathbb{Z}_2$ symmetry, both negative and positive magnetization are equally probable. However, in the thermodynamic limit $\mathcal{N} \to \infty$ (infinitely large lattice), if the system attains a state with spins predominantly in one direction below $T < T_c$, then it cannot spontaneously (in a finite time) fluctuate through a series of local spin flips into a state with the opposite magnetization. This is known as the *spontaneous symmetry-breaking* of the global $\mathbb{Z}_2$ symmetry in the Ising model, and can be physically thought of as arising from the limit of a very weak external field. However, in a finite system, such fluctuations are possible and it is possible for $\langle m \rangle = 0$ for all temperatures T, making it difficult to study the phase transition which appears in the thermodynamic limit.

Therefore, in simulations of finite lattice sizes, **we instead define the order parameter as $\langle |\boldsymbol{m}| \rangle$.** As the system size increases, the $\langle |m| \rangle$ curve sharpens close to T_c and approaches the thermodynamic limit symmetry-broken $\langle m(T) \rangle$ curve.

Similarly, the other relevant physical observable is the energy density which is simply defined as the energy per site

$$e = \frac{E}{\mathcal{N}} = \frac{-J}{\mathcal{N}} \sum_{\langle i,j \rangle} \sigma_i \sigma_j \tag{1.3.2}$$

We can also use magnetization and energy density observables to calculate other derived quantities of interest. In the context of magnetization, we can calculate the magnetic susceptibility

$$\chi \equiv \left. \frac{\partial \langle M \rangle}{\partial h} \right|_{h=0} \tag{1.3.3}$$

i.e. the linear response of magnetization in the weak field limit, i.e., $E' = \lim_{h \to 0} (E - hM)$. It is easy to show that the expression for susceptibility simplifies to

$$\chi = \frac{1}{T} \left(\left\langle M^2 \right\rangle - \langle |M| \rangle^2 \right) = \frac{L^2}{T} \left(\left\langle m^2 \right\rangle - \langle |m| \rangle^2 \right) \tag{1.3.4}$$

where we have replaced the $\langle m \rangle$ with $\langle |m| \rangle$ for reasons discussed previously. Similarly, the specific heat capacity is another quantity of interest which is defined as

$$C_V = \frac{\partial \langle E \rangle}{\partial T} = \frac{L^2}{T^2} \left(\left\langle e^2 \right\rangle - \langle e \rangle^2 \right) \tag{1.3.5}$$

Finally, we are also interested in the derived *Binder cumulant* which is defined as the kurtosis of the order parameter $\langle |m| \rangle$

$$U_L = 1 - \frac{\langle m^2 \rangle_L}{3 \left\langle m^2 \right\rangle_L}, \tag{1.3.6}$$

and the intersection of U_L curves for different lattice side L is used to accurately determine the phase transition point.

Therefore, the measurements of the set of physical observables $\{|m|, m^2, m^4, e, e^2\}$ is sufficient to determine the Q vs. T curves for all the relevant observables $(e, |m|)$ and derived quantities (χ, C_V, U_L). Using Monte Carlo simulations, we can calculate the expectation values and statistical errors in these quantities, and use the simulation data to extract the critical exponents, which we will discuss in later sections.

1.4 Autocorrelation functions

All MCMC algorithms are based upon the central idea of performing local or cluster updates to arrive at the next configuration (or the next state in a Markov chain). Naturally, it is reasonable to expect that the observable measurements are temporally correlated and these correlations are quite significant near the transition points. The only way to get around this is to perform the measurements of the observables after a few time steps so as to ensure that subsequent measurements are statistically uncorrelated. The *autocorrelation function* is a quantitative measure of temporal correlations between measurements. For an observable $\mathcal{O}$, the autocorrelation function is defined as

$$A_{\mathcal{O}}(t) = \frac{1}{N-t} \sum_{k=0}^{N-t-1} \frac{(\mathcal{O}(k) - \langle \mathcal{O} \rangle) \cdot (\mathcal{O}(k+t) - \langle \mathcal{O} \rangle)}{\langle \mathcal{O}^2 \rangle - \langle \mathcal{O} \rangle^2} \tag{1.4.1}$$

where $\mathcal{O}(k)$'s are the measurements performed at each MC sweep, and N is the total number of MC sampling sweeps. Moreover, the averages in Eq. (1.4.1) are calculated only over the first $N-t$ measurements.

$$\langle \mathcal{O}^n \rangle = \frac{1}{N-t} \sum_{k=0}^{N-t-1} [\mathcal{O}(k)]^n \tag{1.4.2}$$

The autocorrelation function is suitably normalized to give values $\in [-1, 1]$. For an ergodic simulation, we expect $A_{\mathcal{O}}(t) \to 0$ as $t \to \infty$; infact, $A_{\mathcal{O}}(t)$ is expected to fall off as an exponential

$$A_{\mathcal{O}}(t) \sim e^{-|t|/\tau}, \tag{1.4.3}$$

and τ is called the *autocorrelation time*. The autocorrelation time is roughly the number of MC steps after which the temporal correlations drop to $1/e \approx 0.37$. A general rule of thumb in MCMC simulations is to **perform measurements of the observables after every $2\boldsymbol{\tau}$ steps.** For an observable $\mathcal{O}$, we define the *integrated autocorrelation time* τ_{int} as a discretized integration sum

$$\tau_{\text{int},\mathcal{O}} = \frac{1}{2} \sum_{t=-\infty}^{\infty} A_{\mathcal{O}}(t) = \frac{1}{2} + \sum_{t=1}^{\infty} A_{\mathcal{O}}(t) \tag{1.4.4}$$

However, this natural estimator isn't *quite* correct because the autocorrelations $A_{\mathcal{O}}(t)$ for $t \gg \tau_{\text{int}}$ contain much noise and little signal. To fix this, we implement the *automatic* *windowing algorithm* [Sokal] by introducing a cutoff M on the sum

$$\tau_{\text{int},\mathcal{O}} = \frac{1}{2} + \sum_{t=1}^{M} A_{\mathcal{O}}(t) \tag{1.4.5}$$

M is chosen as the smallest integer such that $M \geq 6\tau_{\text{int},\mathcal{O}}(M)$. The algorithm can be summarized as follows

1. Start with some small value of M, say $M = 50$, and evaluate $\tau_{\text{int},\mathcal{O}}(50)$.
2. Check if $50 \geq 6\tau_{\text{int},\mathcal{O}}(50)$.
3. If not, increase the value of M and repeat until you find a M where $M \geq 6\tau_{\text{int},\mathcal{O}}(M)$.

We take the autocorrelation time τ_{int} of a given (L, T) simulation as the maximum of autocorrelation times for all observables except magnetization m. This is because the lack of symmetry-breaking in finite lattices causes the time taken to tunnel between positive and negative m states to be very long, leading to extremely high autocorrelations.

$$\tau_{\text{int}} = \max\left\{\tau_{\text{int},|m|},\ \tau_{\text{int},m^2},\ \tau_{\text{int},m^4},\ \tau_{\text{int},e},\ \tau_{\text{int},e^2}\right\} \tag{1.4.6}$$

The first run of the simulation is only used to calculate the integrated autocorrelation time τ_{int}, followed by a second run where measurements are taken after every $2\tau_{\text{int}}$ steps. These statistically independent measurements can then be used to compute the Monte Carlo expectation values and errors.

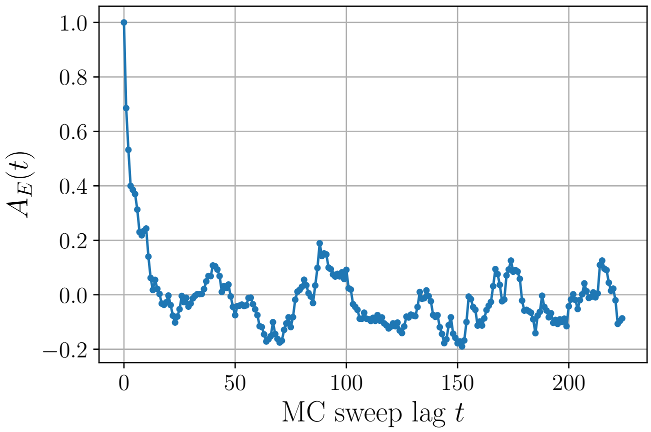


Figure 1.2: Exponential decay of energy density autocorrelation function.

1.5 Estimation of expectation values and errors

Once we have a set of uncorrelated Monte Carlo observable measurements, it is a simple task to calculate the mean and standard deviation of the uncorrelated observable measurement data $(|m|, e)$. However, for quantities that are not measured directly in the simulation but are computed as non-linear combinations of basic observables (χ, C_V, U_L), deriving a propagated error is a highly non-trivial task. Therefore, we utilize *Jackknife analysis* to create large blocks/bins of the data and perform the required computations.

1.5.1 Jackknife binning analysis

We begin by binning the N measurements of observable $\mathcal{O}$ into N_B non-overlapping bins of length k such that $N_B k = N$, and construct a shorter time series of bin averages. The bin average for the j^{th} bin is defined as

$$\mathcal{O}_j^{(B)} \equiv \frac{1}{k} \sum_{i=0}^{k-1} \mathcal{O}_{jk+i} \tag{1.5.1}$$

Below is a visual representation of the binning process

$$\{\underbrace{\{\mathcal{O}_0, \mathcal{O}_1 \ldots \mathcal{O}_{k-1}\}}_{\overline{\mathcal{O}}_0^{(B)}}, \underbrace{\{\mathcal{O}_k, \mathcal{O}_{k+1} \ldots \mathcal{O}_{2k-1}\}}_{\overline{\mathcal{O}}_1^{(B)}}, \ldots\ldots, \underbrace{\{\mathcal{O}_{(N_B-1)k}, \mathcal{O}_{(N_B-1)k+1} \ldots \mathcal{O}_{N-1}\}}_{\overline{\mathcal{O}}_{N_B-1}^{(B)}}\}$$

$$\Downarrow$$

$$\{\overline{\mathcal{O}}_0^{(B)}, \overline{\mathcal{O}}_1^{(B)}, \overline{\mathcal{O}}_2^{(B)} \ldots, \overline{\mathcal{O}}_{N_B-1}^{(B)}\}$$

Knowing the bin averages for the observables $\mathcal{O} \in \{|m|, m^2, m^4, e, e^2\}$, the bin estimates of the derived quantities are estimated as

$$\chi_j^{(B)} = \beta\, L^2 \left(\overline{m^2}_j^{(B)} - \left[\overline{|m|}_j^{(B)} \right]^2 \right) \tag{1.5.2}$$

$$C_V{}_j^{(B)} = \beta^2\, L^2 \left(\overline{e^2}_j^{(B)} - \left[\overline{e}_j^{(B)} \right]^2 \right) \tag{1.5.3}$$

$$U_L{}_j^{(B)} = 1 - \frac{\overline{m^4}_j^{(B)}}{3\left[\overline{m^2}_j^{(B)} \right]^2} \tag{1.5.4}$$

We denote these derived quantities by ρ such that $\rho \in \{\chi, C_V, U_L\}$. The means over the bin averages (for $\mathcal{O}$) and bin estimates (for ρ) are also calculated

$$\overline{\mathcal{O}} = \overline{\overline{\mathcal{O}}_j^{(B)}} = \frac{1}{N_B} \sum_{i=0}^{N_B-1} \overline{\mathcal{O}}_j^{(B)} \tag{1.5.5a}$$

$$\overline{\rho} = \overline{\rho_j^{(B)}} = \frac{1}{N_B} \sum_{i=0}^{N_B-1} \rho_j^{(B)} \tag{1.5.5b}$$

For the Jackknife error analysis, we begin by constructing the same number (N_B) of Jackknife bins containing all data but the j^{th} bin of the previously mentioned binning method. The Jackknife averages for these new bins are defined as

$$\overline{\mathcal{O}}_j^{(J)} = \frac{N\overline{\mathcal{O}} - k\overline{\mathcal{O}}_j^{(B)}}{N-k} \tag{1.5.6}$$

$$\overline{\rho}_j^{(J)} = \frac{N\overline{\rho} - k\rho_j^{(B)}}{N-k} \tag{1.5.7}$$

Since we are using the same data again and again while forming the Jackknife bins, these bins are trivially correlated, and as a result, the Jackknife bin variance is much smaller than is expected. This can be corrected by multiplying the sample variance by another factor of $(N_B - 1)^2$ resulting in

$$\delta\overline{\mathcal{O}} = \sqrt{\frac{N_B - 1}{N_B}\sum_{j=0}^{N_B-1}\left(\overline{\mathcal{O}}_j^{(J)} - \overline{\mathcal{O}}\right)^2} \tag{1.5.8a}$$

$$\delta\rho = \sqrt{\frac{N_B - 1}{N_B}\sum_{j=0}^{N_B-1}\left(\overline{\rho}_j^{(J)} - \overline{\rho}\right)^2} \tag{1.5.8b}$$

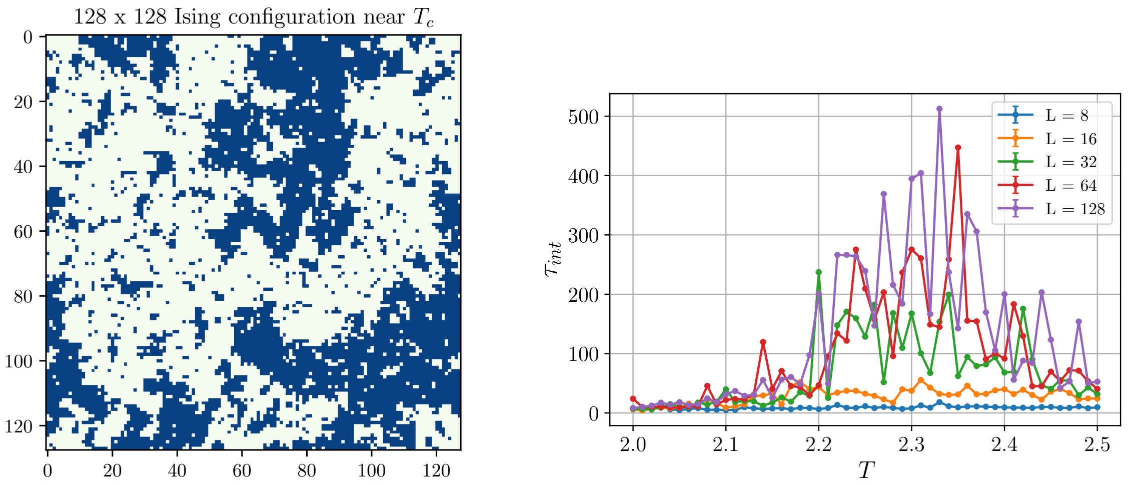
Therefore, using the calculated values from Eq. (1.5.5) and Eq. (1.5.8), we get our mean and error estimates of the observable expectation values and the derived physical quantities

$$\text{Estimate of } \langle\mathcal{O}\rangle \to \overline{\mathcal{O}} \pm \delta\overline{\mathcal{O}}, \qquad \text{Estimate of } \rho \to \overline{\rho} \pm \delta\rho$$

1.6 Critical slowing down near phase transition

Phase transitions, in general, are associated with a discontinuity in the first derivative (for first-order phase transitions) or a divergence in the second derivative (for second-order phase transitions). For example, the classical Ising model is characterized by a second-order phase transition at a critical temperature T_c. Above this temperature ($T > T_c$), the spins are randomly aligned and correspond to the *paramagnetic* phase. Below this temperature ($T < T_c$), the spins majorly align in one of the directions to form the *ferromagnetic* phase of the system.

In the paramagnetic phase, at high temperatures, the spin configuration is random and there are little to no spatial correlations. However, as the temperature decreases, we enter the paramagnetic phase and spins tend to align themselves with their neighbors. As a result, we form large *domains* or *clusters* of spins and we have a high degree of spatial correlation (Fig. 1.3a), often measured by the parameter ξ, known as the correlation length. As one approaches the critical temperature T_c from the above, the onset of strong correlations results in *domain flips* which cause high fluctuations in magnetization and energy. Since the Metropolis algorithm performs local updates and the spins have a strong spatial correlation, we need to wait for a long autocorrelation time to get independent



(a) Domain formation near T_c ($T = 2.30$).

(b) High autocorrelation times near T_c for Metropolis.

Figure 1.3: Critical slowing down near the transition point in the Ising model.

subsequent measurements (Fig. 1.3b). This is known in literature as *critical slowing down.*

The phenomena of critical slowing down is primarily dependent on the algorithm used. Since local update (such as Metropolis) take a long time to flip the domains and generate statistically independent configurations, they have a significantly higher computational cost as a tradeoff to their conceptual simplicity.

Therefore, it should be intuitively clear that some sort of non-local updates should be able to alleviate this problem by flipping entire clusters of spins. This class of algorithms is generally known as *Cluster-update algorithms.* Although the critical slowing down can be significantly reduced with cluster algorithms, they are still far less general applicable than local update algorithms.

1.7 Wolff Cluster algorithm

The acceptance probability of local update algorithms is very low around the critical regions making it an inefficient way to probe the critical properties of the system. In this section, we will discuss the *Wolff cluster algorithm*, which is a generalization of the Swendsen-Wang clyster dynamics and is even more efficient. The Wolff cluster algorithm works by identifying domains of identically oriented spins and performing a flip operation on the entire cluster [Krauth].

1.7.1 Deriving the probabilities for Wolff algorithm

We will start by considering the example of two states a and b which differ by flipping of a cluster of identically oriented spins.

This can be done by starting the construction of a cluster which starts by seeding a random spin and iteratively adding neighboring sites of the same spin with a probability p. This means if a spin i is in the cluster, and a neighboring site j is not (and if $\sigma_i = \sigma_j$), then the link $\langle i, j \rangle$ should be added into the cluster with a probability p. Once the construction has stopped, we will still have n_{same} identical spin links just outside the cluster boundary which were not added into the cluster. Similarly, we say the number of opposite spin links which have a spin opposite to that of the cluster is n_{opp}. Also, $n_{\text{same}}+n_{\text{opp}} = N$ where N is the total number of links the cluster makes with outside spins.

Since the probability of a site not being accepted into the cluster is $(1-p)$, the "a-priori" probability of proposing a cluster flip from $a \to b$ is

$$\mathcal{A}(a \to b) = \mathcal{A}_{\text{interior}} \cdot (1-p)^{n_{\text{same}}}, \tag{1.7.1}$$

where $\mathcal{A}_{\text{interior}}$ is just a product of probabilities for links that were accepted into the cluster. In the reverse direction, if we would have started with the state b and tried constructing a cluster and flipping it to go to state a, the "a-priori" probability to propose a cluster flip from $b \to a$ is

$$\mathcal{A}(b \to a) = \mathcal{A}_{\text{interior}} \cdot (1-p)^{n_{\text{opp}}} \tag{1.7.2}$$

because in state b, the unaccepted but identical spins were exactly those which had the opposite spin in state a. Therefore, if we invoke the detailed balance condition Eq. (1.1.10) for the cluster flips to get

$$\frac{\mathcal{A}(a \to b) \cdot P_{\text{accept}}(a \to b)}{\mathcal{A}(b \to a) \cdot P_{\text{accept}}(b \to a)} = e^{-\beta(E_b - E_a)} \tag{1.7.3}$$

Now the energies of states a and b can be computed as nearest neighbor interaction *inside the cluster* and the interaction *outside the cluster*. Since a and b only differ by a cluster flip, the only difference appears in the interaction at the cluster boundary

$$E_a = E_{\text{interior}} + E_{\text{exterior}} - n_{\text{same}} J + n_{\text{opp}} J, \tag{1.7.4a}$$

$$E_b = E_{\text{interior}} + E_{\text{exterior}} - n_{\text{opp}} J + n_{\text{same}} J \tag{1.7.4b}$$

$$\implies E_b - E_a = 2J(n_{\text{same}} - n_{\text{opp}}) \tag{1.7.4c}$$

Therefore, putting in the expressions into the detailed balance condition, we get

$$\frac{P_{\text{accept}}(b \to a)}{P_{\text{accept}}(a \to b)} = e^{-2\beta J(n_{\text{same}} - n_{\text{opp}})} (1-p)^{n_{\text{opp}} - n_{\text{same}}} \tag{1.7.5a}$$

$$\implies \frac{P_{\text{accept}}(b \to a)}{P_{\text{accept}}(a \to b)} = \left(\frac{e^{-2\beta J}}{1-p} \right)^{n_{\text{same}} - n_{\text{opp}}} \tag{1.7.5b}$$

Thus, given a value of p (which can be a function of β and J), we can get the corresponding acceptance probability for the cluster algorithm. An interesting coincidence occurs when

$$p = 1 - \exp(-2\beta J). \tag{1.7.6}$$

This special choice of probability function yields a *rejection-free* algorithm whose acceptance probability is always 1.

$$\left.\frac{P_{\text{accept}}(b \to a)}{P_{\text{accept}}(a \to b)}\right|_{p=1-\exp(-2\beta J)} = \left(\frac{e^{-2\beta J}}{e^{-2\beta J}}\right)^{n_{\text{same}}-n_{\text{opp}}} = 1 \tag{1.7.7}$$

Thus, any proposed cluster construction and flipping is an always accepted move in the cluster algorithm. The cluster algorithm with the particular choice of $p = 1 - e^{-2\beta J}$ is known as the Wolff cluster algorithm.

1.7.2 Implementation of the Wolff algorithm

In this subsection, we will try to discuss an algorithmic implementation of Wolff clusters so that it can be easily coded into a Monte Carlo program.

Before starting, we will assume we are working with a 2-dimensional spin configuration and we aim to generate configurations distributed according to the Boltzmann distribution using Monte Carlo evolution with transition probabilities given by the Wolff scheme, which is the fastest currently known simulation method for the Ising model. One Wolff algorithm step is then defined in the following step-by-step procedure,

1. Randomly choose a site i on the lattice with the spin σ_i.
2. Label the neighbors of i as j. Add them to a temporary set. Call this set N_{temp}.
3. If $\sigma_i = \sigma_j$, add the neighbor j from set N_{temp} to the cluster $\mathcal{C}$ with a probability of $p = 1 - e^{-2\beta J}$.
4. Repeat step 2 for all the neighbors in set N_{temp}.
5. For sites that are *newly added* to the cluster, check for neighboring identical spins and add them to the cluster $\mathcal{C}$ by repeating steps 2 to 4.
6. Keep repeating till no more spins are added to the cluster $\mathcal{C}$.
7. Once the cluster is constructed, flip all the spins, i.e., $\forall\, j\ \in \mathcal{C} \mid \sigma_j \to -\sigma_j$.

More concisely, a single sweep of the Monte Carlo Wolff algorithm in `psuedocode language` is given below from [Krauth]. The sets $\mathcal{F}_{\text{old}}$ and $\mathcal{F}_{\text{new}}$ denote the frontier of the cluster at $(n-1)^{\text{th}}$ and n^{th} steps, respectively. The set $\mathcal{C}$ contains the sites belonging to the cluster.

```
algorithm wolff-cluster
begin
    $i$ := random site;
    $\mathcal{C} := \{i\}$;
    $\mathcal{F}_{\text{old}} := \{i\}$;
    while $\mathcal{F}_{\text{old}} \neq \{\}$ do
    begin
        $\mathcal{F}_{\text{new}} := \{\}$;
        for $\forall\ i\ \in\ \mathcal{F}_{\text{old}}$ do
        begin
            for $\forall\ j$ neighbor of $i$ with $\sigma_i = \sigma_j,\ j \notin \mathcal{C}$ do
            begin
                if $\text{ran}[0,1] < p$ then
                begin
                    $\mathcal{F}_{\text{new}} := \mathcal{F}_{\text{new}} \cup \{j\}$;
                    $\mathcal{C} := \mathcal{C} \cup \{j\}$;
                end
            end
        end
        $\mathcal{F}_{\text{old}} := \mathcal{F}_{\text{new}}$;
    end
    for $\forall\ i\ \in\ \mathcal{C}$ do
    $\sigma_i := -\sigma_i$;
end
```

The resulting dynamics is ergodic and obeys detailed balance.

1.8 Results

We have now setup our complete machinery for calculating physical observable expectation values (and errors) using Monte Carlo Wolff algorithm simulations. The workflow of the Monte Carlo code is summarized in the following section.

1.8.1 Structure of the code

The `ising` executable program calculates the expectation values of various observables and physical quantities derived from them, along with error bars. Each iteration of the program runs for a single (L, T) pair. The program is structured into four parts

Part 1 A preliminary Monte Carlo run to collect autocorrelated observable measurements after the system equilibrates.

Part 2 Calculation of the integrated autocorrelation time τ_{int} using the autocorrelated measurements.

Part 3 The main Monte Carlo run with observable measurements being taken after every $2\tau_{\text{int}}$ MC sweeps to ensure collection of uncorrelated measurements.

Part 4 Calculating observable expectation values and derived physical quantities with error bars using jackknife binning method.

After obtaining the results of expectation values and derived physical quantities (with error bars) for a range of temperatures, we perform finite-size scaling analysis on the data to extract the critical exponents of the 2D Ising Model. The `ising-2d` source code is written in `C++` and the files can be accessed on the following link - `ising-2d` on GitHub.

1.8.2 Simulation results

The following parameters were used in the 2D Ising model simulations for lattice sizes of $L = 8, 16, 32, 64, 128$. Here, we have used the Wolff scheme to simulate the Markov chain Monte Carlo procedure instead of the Metropolis algorithm for reasons discussed in earlier sections. As a short remark on units, we are using the natural units where $k_B = 1$, so our units of temperature and energy are identical. We have also set the coupling paramater as unity `J = 1.0`.

For preliminary Monte Carlo (Part 1)

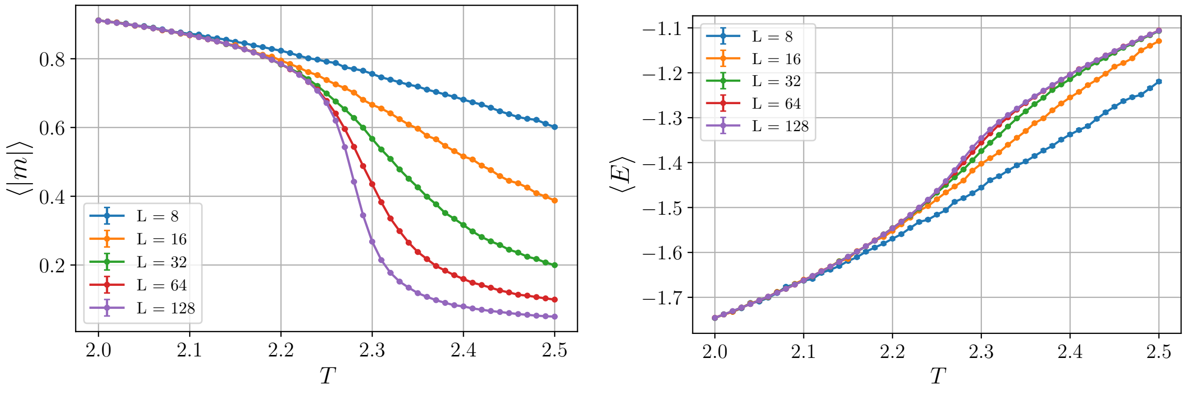
```
no of equilibration sweeps = 1.0e3
no of sampling sweeps = 1.0e4
sampling step size = 1.0e0
```

For the main Monte Carlo run (Part 3)

no of equilibration sweeps = 1.0e3

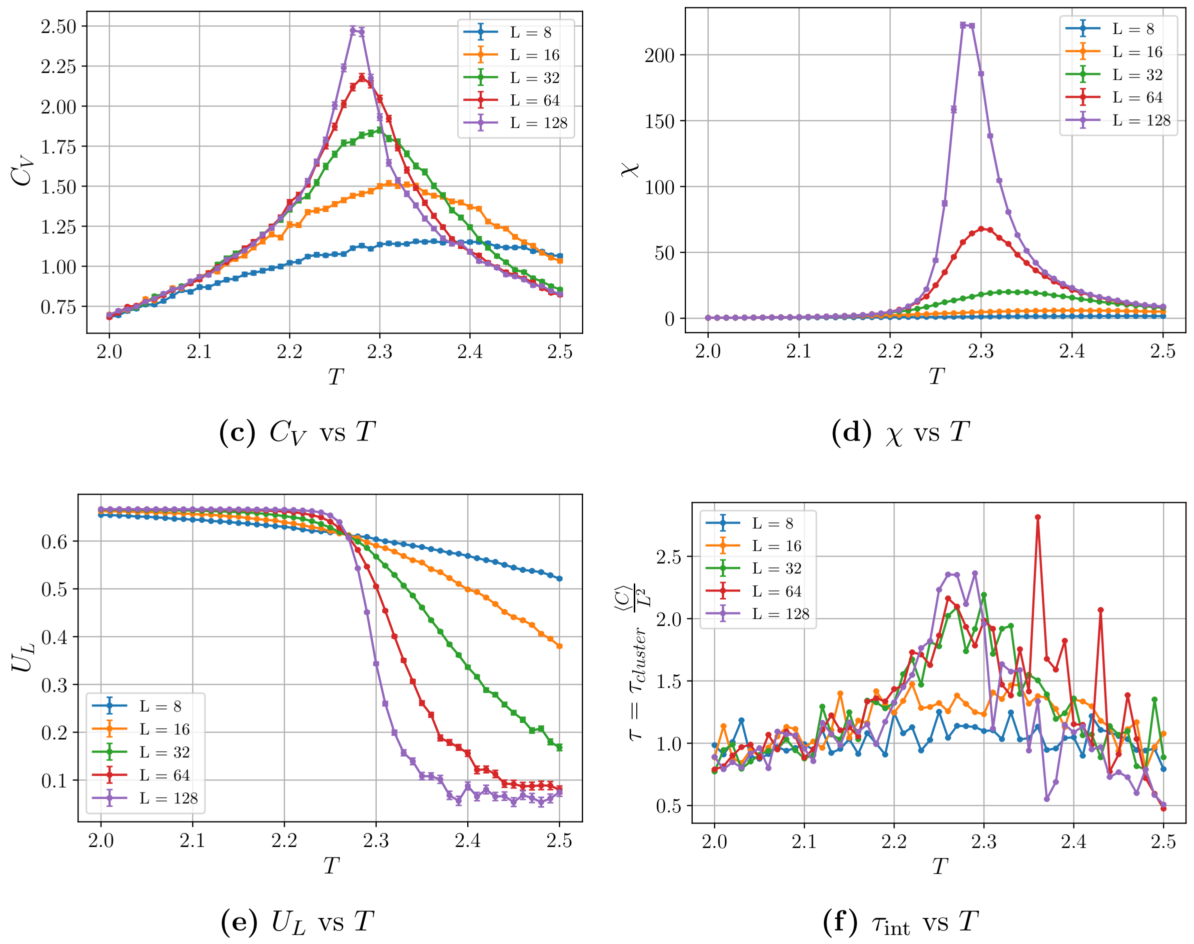
no of sampling sweeps = $2\tau \cdot$ 2.0e4

sampling step size = 2τ



(a) $\langle |m| \rangle$ vs T

(b) $\langle e \rangle$ vs T



(c) C_V vs T

(d) χ vs T

(e) U_L vs T

(f) τ_{int} vs T

Figure 1.4: Variation of physical quantities with T for different lattice sizes L.

The Figure 1.4 above shows plots of expectation values of relevant observables, derived physical quantities and the autocorrelation times (scaled by $\langle C \rangle / L^2$) as they vary with T for different lattice sizes L. These plots summarize the entire critical structure of Ising model tansition.

The average (absolute) magnetization curve as a function of temperature (Fig. 1.4a) is one of the most interesting results of the Ising model. The average (absolute) magnetization is the order parameter for the Ising model which distinguishes the paramagnetic and ferromagnetic phases of the model. As we go to higher and higher lattice sizes, the transition points becomes progressively sharper and shows a clear transition from ferromagnetism ($\langle |m| \rangle \approx 1$) to paramagnetism ($\langle |m| \rangle \approx 0$) after a certain transition temperature T_c at $L = 128$.

Another interesting feature of the Ising phase transitions are the specific heat C_V (Fig. 1.4c) and magnetic susceptibility χ (Fig. 1.4d) curves. They show an approximate divergence at the critical temperature T_c, which becomes exact in the thermodynamic limit $L \to \infty$. Hence, our numerical studies confirm that the Ising phase transition is second-order in nature.

The way to numerically estimate the critical temperature T_c of the phase transition was given by Binder [Binder 1981]. According to Binder, the transition temperature T_c is the temperature at which different U_L curves (corresponding to different lattice sizes L) cross in the thermodynamic limit (Fig. 1.4e). In our studies, we estimated the critical temperature $T_c \approx 2.269$, which is very close to the exact value of $2/\ln\left(1+\sqrt{2}\right) \approx 2.26918$.

Finally, a short note on autocorrelation times. As one can see in Fig. 1.4f, we have scaled the τ_{cluster} by $\langle C\rangle/L^2$, where $\langle C\rangle$ is the average cluster size. We claim that this is the correct autocorrelation time to compare with the Metropolis autocorrelation times [Tamayo et al]. This is because N Metropolis sweeps are fundamentally different than N Wolff sweeps. If we go through N sweeps,

$$\begin{aligned}\text{For Metropolis } &\longrightarrow N \cdot L^2 \text{ spin flips}\\ \text{For Wolff cluster } &\longrightarrow N \cdot \langle C\rangle \text{ spin flips}\end{aligned}$$

So if we were to define an equivalent unit of time as (# of spin flips)/L^2, then for N sweeps

$$T_{\text{Metro}} = \frac{N \cdot L^2}{L^2} = N$$
$$T_{\text{Wolff}} = N\frac{\langle C\rangle}{L^2}$$

Hence, we also scale our Wolff autocorrelation time τ_{cluster} by a factor of $\langle C\rangle/L^2$, and we can see in Fig. 1.4f that it's much more efficient compared to the Metropolis algorithm (Fig. 1.3b). Further, the autocorrelation times are observed to scale with system size as a power law

$$\tau_{\text{int}} \sim L^z$$

where z is the dynamical critical exponent, and can be used to quantify the efficiency of the algorithm. We will, however, not go in that direction and instead focus on the remaining critical exponents which characterize the Ising phase transition.

1.9 Critical phenomena and Finite-size scaling

We will now give a brief review of critical phenomena and the finite-size scaling properties that can be quantitatively used to study critical properties of the problem using Monte Carlo simulations. In general, phase transitions only happen in the thermodynamic limit ($L\infty$). However, since we only possess finite computational memory and processing time, it is not possible to numerically simulate arbitrarily large system sizes to study the critical properties of a system. Therefore, we use the theory of *finite-size scaling analysis* which uses the results for finite system sizes to deduce conclusions for the thermodynamic limit.

1.9.1 Critical Exponents

The critical phenomena are generally scaling forms that occur near the critical point. For the classical Ising model, we obtain scaling relations for observable expectations near the classical critical point, i.e., the critical temperature T_c. As mentioned earlier, we can use the data from finite-size studies to estimate the critical exponents using the *data-collapse method*. However, this analysis is only possible for second-order phase transitions and it fails for first-order transitions.

The most fundamental concept underlying the theory of critical phenomena is that of a *correlation length*, denoted by ξ. It roughly denotes the typical size of ordered domains in the lattice. Close to the critical point T_c, the correlation length diverges according to a power law

$$\xi \sim t^{-\nu}, \tag{1.9.1}$$

where t is defined as the reduced temperature

$$t \equiv \frac{|T - T_c|}{T_c}, \tag{1.9.2}$$

and ν is one of the critical exponents, and is the same whether we approach T_c from above or below. Similarly, the critical exponent corresponding to the onset of magnetic order is given by β. The order parameter $\langle |m| \rangle$ is zero above T_c, and below T_c it scales as

$$\langle |m| \rangle \sim t^{\beta} \quad (\sim \xi^{-\beta/\nu}) \tag{1.9.3}$$

The susceptibility, as noted previously, diverges at T_c in the thermodynamic limit and scales as

$$\chi \sim t^{-\gamma} \quad (\sim \xi^{\gamma/\nu}) \tag{1.9.4}$$

The specific heat is also diverges from both above and below as it approaches T_c

$$C_V \sim t^{-\alpha} \quad (\sim \xi^{\alpha/\nu}) \tag{1.9.5}$$

It is, however, to be noted that the above scaling relations near the critical point hold true in the infinite size limit ($L \to \infty$). Systems belonging to the same *universality class* have the same set of critical exponents $\{\nu, \beta, \gamma, \delta, \alpha\}$. The universality class does not depend on the microscopic details but instead on the behavior of the system near the critical point.

1.9.2 Finite-size scaling hypothesis

How would the scaling relations modify if the lattice sizes L are finite? The order parameter $\langle |m| \rangle$ does not exactly vanish for $T \geq T_c$, and the divergent quantities (like χ and C_V) saturate to a finite value.

One possible way is to perform finite-size simulations for higher and higher L until the results converge, and extract the critical exponents by estimating the exponent in the power law. However, a much more systematic way is to perform a *finite-size scaling* (FSS) study which uses the regularity in the deviations to extract the critical exponents.

Consider the susceptibility in an infinite size limit which scales as

$$\chi \sim \xi^{\gamma/\nu}$$

Then the *finite-size scaling hypothesis* (which can be proven via renormalization group theory) states that observables close to T_c scale with an additional non-divergent function of ξ/L to account for the deviations in critical behavior at finite sizes

$$\chi(\xi, L) = \xi^{\gamma/\nu}\, f(\xi/L) \tag{1.9.6}$$

where $f(x) \to$ const. as $x \to 0$ ($L \to \infty$). Since the deviations from the $L \to \infty$ critical behavior occurs when the correlation length ξ is comparable to the system size L, and the fact that $\xi \sim t^{-\nu}$, we obtain the much more familiar expression of the FSS hypothesis

$$\chi(t, L) = L^{\gamma/\nu}\, g(tL^{1/\nu}) \tag{1.9.7}$$

where g is another scaling function. More generally, for any observable (or derived quantity) Q, the quantity scales as

$$Q(t, L) = L^{\zeta/\nu}\, g_Q(tL^{1/\nu}) \tag{1.9.8}$$

where ζ is the critical exponent corresponding to the scaling of the quantity Q.

Finite-size scaling relations for 2D Ising model (1.9.9)

$$\begin{aligned} U_L &= \quad g_U(tL^{1/\nu}) \\ |m| &= L^{-\beta/\nu}\, g_m(tL^{1/\nu}) \\ \chi &= L^{\gamma/\nu}\, g_\chi(tL^{1/\nu}) \\ C_V &= \ln(L)\, g_C(tL^{1/\nu}) \end{aligned}$$

In the case of the 2D Ising model, $\alpha = 0$, and hence the power law scaling doesn't hold. Instead, we have a logarithmic scaling as indicated in the scaling of C_V as shown above.

1.9.3 Data collapse method

Since the master curve $g_Q(x)$ for a quantity Q is an unknown function, one can plot $Q(t, L)\, L^{-\zeta/\nu}$ versus $x = tL^{1/\nu}$ for different sizes to extract the master curve. If the finite-size scaling hypothesis is correct, data for all the different lattice sizes should *collapse* onto each other. However, this would require inputting the correct value of the critical exponents ζ and ν.

Therefore, the problem of *estimating the critical exponents* can be rephrased as a data collapse problem of the observable data ($Q(t,L)\,L^{-\zeta/\nu}$ versus $x = tL^{1/\nu}$) by fine-tuning the exponents ζ and ν. We use the following cost function P_b to as a measure of the collapse, which is essentially a normalized sum of residues, which we get by slightly modifying the cost function given in [1].

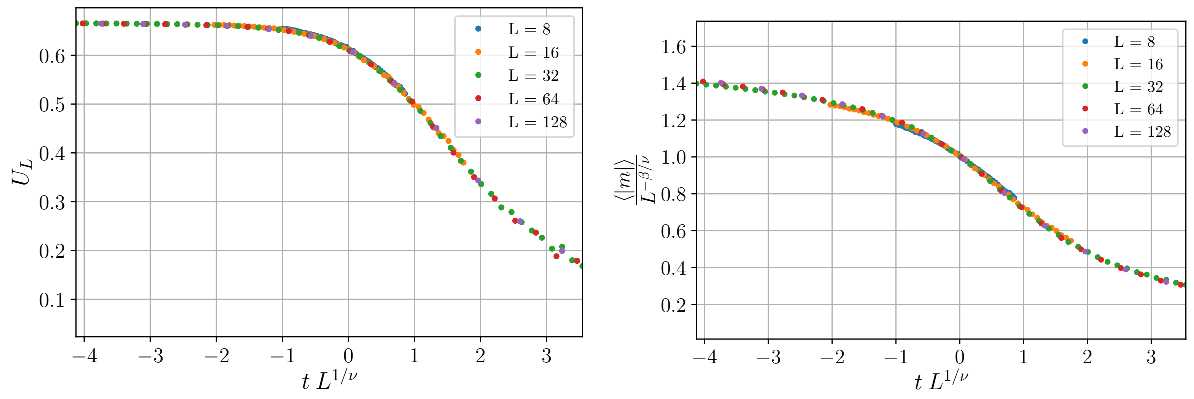
$$P_b = \frac{1}{N_{\text{over}}} \sum_p \sum_{j \neq p} \sum_i \frac{\left| L_j^{-\zeta/\nu} Q_{ij} - \varepsilon_p\left(L_j^{1/\nu} t_{ij}\right) \right|}{L_j^{-\zeta/\nu} Q_{ij} + \varepsilon_p\left(L_j^{1/\nu} t_{ij}\right)} \tag{1.9.10}$$

- p indexes the data-set associated to length L_p.
- j also indexes the data-set associated to length L_j, but doesn't include the set p.
- i indexes the data-points inside the set associated to L_j.
- N_{over} is the number of terms we sum over.
- ε_p is the interpolating function based on the values of set p bracketing the argument in question.

The critical exponents are obtained by minimizing the cost funtion P_b with respect to β, γ, ν

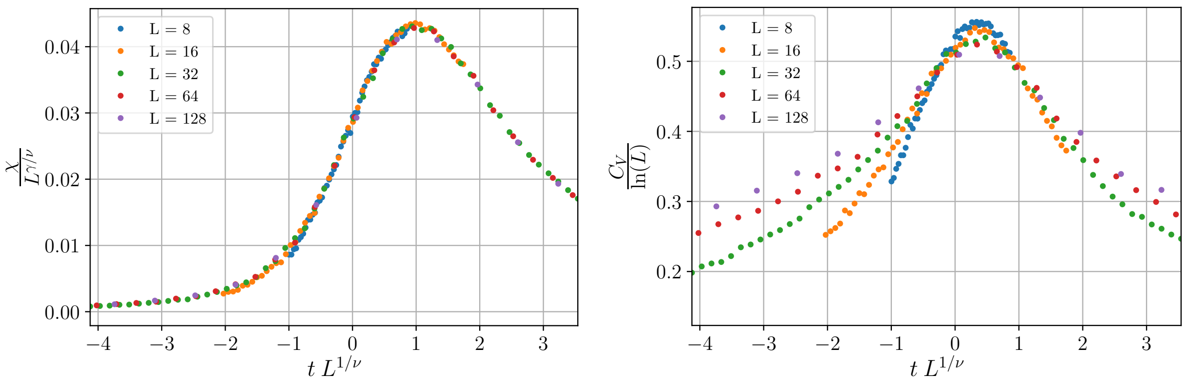
- $Q = U_L$, minimize $P_b(0,\nu)$ with respect to ν. $\longrightarrow \nu_0$
- $Q = \langle |m| \rangle$, minimize $P_b(-\beta, \nu_0)$ with respect to β. $\longrightarrow \beta_0$
- $Q = \chi$, minimize $P_b(\gamma, \nu_0)$ with respect to γ. $\longrightarrow \gamma_0$

The data collapse of the quantities obtained from the Monte Carlo studies is shown in Fig. 1.5. The appropriate fine-tuning of the critical exponents results in an almost perfect collapse of the data near the critical point $t = 0$.



(a) Data collapse for U_L.

(b) Data collapse for $\langle |m| . \rangle$



(c) Data collapse for χ.

(d) Data collapse for C_V.

Figure 1.5: Data-collapse to extract the critical exponents.

The following table (Table 1.1) summarizes the results of the critical exponents computed from a finite-size scaling study of the data obtained from Monte Carlo simulations compared to the exact values.

Physical Quantity	Exponent	FSS estimate	Exact Value
Specific Heat (C_V)	α	–	0
Order Parameter ($\langle\lvert m\rvert\rangle$)	β	0.1204	1/8
Magnetic Susceptibility (χ)	γ	1.7301	7/4
Correlation Length (ξ)	ν	0.9765	1
Critical Temperature	T_c	2.269	$2/\ln\left(1+\sqrt{2}\right)$

Table 1.1: Summary of critical exponents for the 2D Ising model.

1.10 What now?

In this chapter, we provided a comprehensive introduction for implementing Monte Carlo simulations to classical spin system models. We explicitly discussed the example of the 2D Ising model, constructed both local and cluster update algorithm to compute expectation values of the observables, and described the framework of extracting the critical exponents via finite-size scaling analysis. This methodology can further be extended to other spin models for computing the critical properties near second-order phase transitions. Moreover, in the upcoming chapters, we'll discuss a scheme to generalize this approach to quantum spin models and use *Quantum Monte Carlo* methods to explore the critical properties of the quantum phase transition.

Chapter 2

Singlet constraint

The $\mathbb{Z}_2$ gauge theory Hamiltonian is given by

$$H = -\sum_p J_p B_p - \sum_l h_e X_e \tag{2.0.1}$$

where $B_p = \prod_{l \in p} Z_e$ with p denoting a plaquette of spin links e in the lattice. On a torus (periodic boundary) conditions, we get an additional constraint

$$\prod_p B_p = \mathbb{I}$$

If we now perform a dual mapping of this theory by identifying the following

$$B_p = \sigma_p^x, \qquad X_e = \sigma_{p_e}^z \sigma_{p_e'}^z$$

The dual Hamiltonian then becomes

$$H_{\text{dual}} = -\sum_p J_p \sigma_p^x - \sum_e h_e \sigma_{p_e}^z \sigma_{p_e'}^z \tag{2.0.2}$$

which is the Transverse Field Ising Model (TFIM) Hamiltonian. However, now the constraint of periodic boundary conditions from $\mathbb{Z}_2$ gauge theory carries on forward to generate the following constraint on the TFIM

$$\prod_p B_p = \boxed{\prod_p \sigma_p^x = \mathbb{I}}$$

The above is known as the **singlet constraint**.

2.1 Quantum-to-Classical correspondance

To perform Monte Carlo on the singlet-Ising model,

$$H = -\sum_p J_p \sigma_p^x - \sum_e h_e \sigma_{p_e}^z \sigma_{p_e'}^z \ , \qquad \prod_p \sigma_p^x = \mathbb{I} \tag{2.1.1}$$

we map the d-dimensional quantum Hamiltonian to a $d+1$-dimensional generalised classical Ising model using the path integral. In the end, we obtain the effective action with singlet constraint imposed

$$S_s[\{\sigma\}] = -\sum_l \sum_e (\Delta\tau) h_e \sigma_{p_e}(l) \sigma_{p_e'}(l) - \sum_l \ln\cosh\left[-\sum_p K_p \sigma_p(l+1)\sigma_p(l)\right] \tag{2.1.2}$$

where σ's are now the classical spin variables $\in \{-1, 1\}$, and the label l denotes the summation over the imaginary time domain such that $l \in \{1, \ldots N_\tau\}$ and $\sigma_i(N_\tau + 1) = \sigma_i(1)$ with $N_\tau \Delta\tau = \beta$. The newly defined K_p is

$$K_p = -\frac{1}{2} \ln\tanh\left(\frac{\beta J_p}{N_\tau}\right)$$

For a 1-dimensional TFIM (with singlet constraint), the corresponding 2-dimensional generalised Ising model comes out as

$$S[\{\sigma\}] = -\sum_{l=1}^{N_\tau} \sum_{i=1}^{N_x} (\Delta\tau) h_i \sigma_i(l) \sigma_{i+1}(l) - \sum_{l=1}^{N_\tau} \ln\cosh\left[-\sum_{i=1}^{N_x} K_i \sigma_i(l+1)\sigma_i(l)\right] \tag{2.1.3}$$

2.2 Alignment observable

Carefully analyzing the term $\sigma_i(l)\sigma_i(l+1)$ in Eqn. (2.1.3), we see that

- $\sigma_i(l)\sigma_i(l+1) = -1$ if $\sigma_i(l) = -\sigma_i(l+1)$ $\implies$ **anti-aligned pair** ,
- $\sigma_i(l)\sigma_i(l+1) = +1$ if $\sigma_i(l) = +\sigma_i(l+1)$ $\implies$ **aligned pair**.

Therefore, the term $\sum_i \sigma_i(l)\sigma_i(l+1)$ acts like a "measure of alignment" between spins in adjacent layers, contributing $+1$ to the sum if a pair is aligned, and -1 if the pair is anti-aligned. Hence, we define a new **alignment observable** $A(l)$

$$A(l) \equiv \frac{1}{N_x} \sum_{i=1}^{N_x} \sigma_i(l)\sigma_i(l+1) = \frac{1}{N_x}(N_a(l) - N_o(l)) \quad \in [-1, 1] \tag{2.2.1}$$

where $N_a(l)$ is the number of aligned and $N_o(l)$ is the number of opposite (anti-aligned) spin pairs in adjacent layers l and $l+1$. If we further keep the coupling strengths constant, our effective action becomes

$$S = -h\Delta\tau \sum_{i=1}^{N_x} \sum_{l=1}^{N_\tau} \sigma_i(l)\sigma_i(l+1) - \sum_{l=1}^{N_\tau} \ln\cosh\left[-KN_xA(l)\right]. \tag{2.2.2}$$

An interesting point to note is that the action is agnostic to the sign of the alignment $A(l)$, which implies that an "alignment flip" is a subsystem symmetry of Eqn. (2.2.2).

2.3 Subsystem symmetry breaking

The effective action in Eqn. (2.2.2) possesses the following subsystem symmetry

$$\hat{F}(l) = \prod_{i \in \text{layer } l} \hat{X}_{(i,l)}$$

i.e. flipping all the spins in layer l, or an **alignment flip** $A(l) \to -A(l)$, leaves the S invariant. Therefore, both $A(l) = +A$ and $A(l) = -A$ states should be equally probable.

However, as we increase the system size $N_x \gg 1$, the minima of S get deeper, and the system chooses either the $A(l) \approx +1$ or $A(l) \approx -1$ state and stays there, unable to cross the energy barrier to go to the other side of the configuration space.

Even though S possesses the subsystem symmetry, these states don't. This is known as **subsystem symmetry breaking**, and $A(l)$ acts like an order parameter for this system (just like m in 2D Ising model with $\mathbb{Z}_2$ symmetry breaking in the low temperature limit).

2.4 Metropolis Monte Carlo algorithm

To demonstrate the process of subsystem symmetry breaking in the model, we show how single spin-flips in Metropolis algorithm are unable to let the system cross the energy barrier and explore the other parts of the configuration space.

In our algorithm, we keep $\Delta\tau = 1$ fixed, $\implies$ $\beta = N_\tau$. So the ground state of the model can be analysed by choosing a large enough N_τ to emulate $\beta \to \infty$ limit.

An algorithm for performing Metropolis on our model of interest is as follows

```
algorithm singlet-metropolis
begin
    define $\sigma[N_x, N_\tau]$;
    initialize spins (all up, hot-start, as input) $\sigma[i, l]$;
    define weighted alignment array $\varepsilon[N_\tau]$;
    initialize $\varepsilon[l] = -\sum_i K_i \sigma_i(l+1)\sigma_i(l), \quad \forall\, l$;
    // Metropolis updates
    for $n \in \{1, 2, \ldots N_\text{samp}\}$
    begin
        // Single Monte Carlo step
        for $m \in \{1, 2, \ldots N_x \cdot N_\tau\}$
        begin
            choose site (random, sequential) := $(i_0, l_0)$;
            define $\varepsilon'_{l_0},\ \varepsilon'_{l_0-1}$;
            define $\Delta S,\ \Delta S_x,\ \Delta S_t$;

            $\varepsilon'_{l_0-1} = \varepsilon[l_0 - 1] + 2K_{i_0} \cdot \sigma(i_0, l_0) \cdot \sigma(i_0, l_0 - 1)$;
            $\varepsilon'_{l_0}\ \ \ = \varepsilon[l_0]\qquad\ + 2K_{i_0} \cdot \sigma(i_0, l_0) \cdot \sigma(i_0, l_0 + 1)$;

            $\Delta S_x = 2\Delta\tau \cdot h_{i_0} \cdot \sigma(i_0, l_0) \cdot [\sigma(i_0 - 1, l_0) + \sigma(i_0 + 1, l_0)]$;
            $\Delta S_t = \ln\left[\cosh(\varepsilon[l_0]) / \cosh\left(\varepsilon'_{l_0}\right)\right] + \ln\left[\cosh(\varepsilon[l_0 - 1]) / \cosh\left(\varepsilon'_{l_0-1}\right)\right]$;
            $\Delta S = \Delta S_x + \Delta S_t$;

            define $p$ = random number;
            if $p < \exp(-\Delta S)$
            begin
                $\varepsilon[l_0 - 1] = \varepsilon'_{l_0-1}$;
                $\varepsilon[l_0]\qquad = \varepsilon'_{l_0}$;
                $\sigma[i_0, l_0] = (-1) \cdot \sigma[i_0, l_0]$;
            end
        end
    end
end
```

2.5 Metropolis simulation results

2.5.1 Expectation values of alignment

To show that the subsystem symmetry is indeed broken for large spatial lattice sizes $N_x \gg 1$, we compute expectation value $\langle A(l) \rangle$, where l labels the layers $\in \{1, 2, \ldots N_\tau\}$.

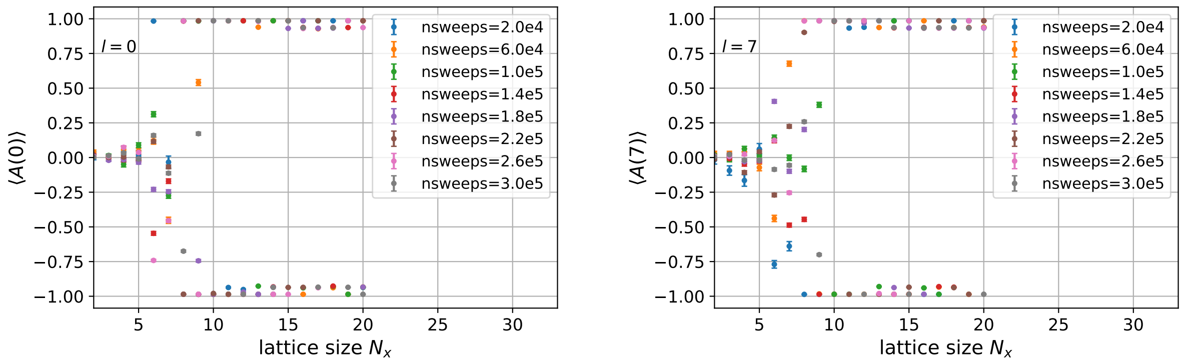
For small spatial lattice sizes, we expect $\langle A(l) \rangle \approx 0$ since the energy barrier to jump between different parts of the configuration space isn't high enough. However, for larger spatial lattic sizes, we expect the action minima to get deeper, leading to the system state getting stuck in either of the $\langle A(l) \rangle \approx \pm 1$ states and unable to cross to the other side.

For the simulations, we use the following parameters spatial lattice size $N_x \in \{2,\ 3,\ 4\ \ldots,\ 20\}$

no of sampling sweeps $N_{\text{samp}} \in \{2.0\text{e}4,\ 6.0\text{e}4,\ 1.0\text{e}5,\ \ldots,\ 2.6\text{e}5,\ 3.0\text{e}5\}$

imaginary time lattice size $N_\tau = 40,\quad \Delta\tau = 1.0$

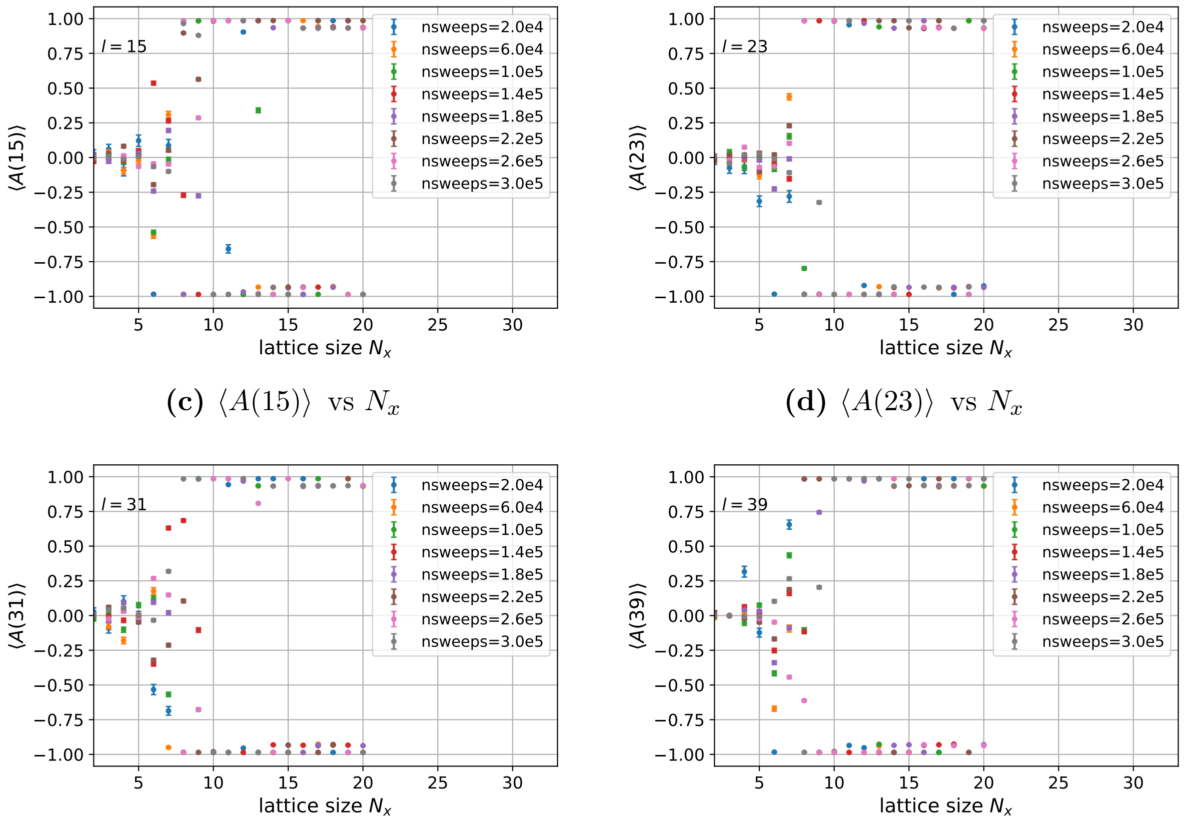
coupling constants $K, h = 1.0$



(a) $\langle A(0)\rangle$ vs N_x

(b) $\langle A(7)\rangle$ vs N_x

Figure 2.1: $\langle A(l)\rangle$ for different values of l calculated for different number of Monte Carlo sweeps.



(c) $\langle A(15)\rangle$ vs N_x

(d) $\langle A(23)\rangle$ vs N_x

(e) $\langle A(31)\rangle$ vs N_x

(f) $\langle A(39)\rangle$ vs N_x

Figure 2.1: $\langle A(l)\rangle$ for different values of l calculated for different number of Monte Carlo sweeps.

As can be seen from Fig. 2.1, the plots for $\langle A(l)\rangle$ vs N_x look roughly the same, suggesting that the expectation values themselves are independent of the chosen layer. This can also be seen in Fig. 2.2 where we have plotted $\langle A(l)\rangle$ vs N_x for two different layers (for the same number of Monte Carlo sweeps). With the same number of MC sweeps to explore the configuration space, we see that the plots for both the layers l_1 and l_2 fall into the broken symmetry state around the same value of $N_x \approx 8$. This demonstrates that all the layers l show roughly the same dynamics, i.e. in the high spatial lattice size limit $N_x \gg 1$, they get stuck in either of the $\langle A(l)\rangle \approx \pm 1$ wells, and aren't able to escape it, resulting in a bifurcation-like envelope stucture.

Another interesting point to note from Fig. 2.1 is the dependence of $\langle A(l)\rangle$ on the number of Monte Carlo sweeps `nsweeps`. As we increase `nsweeps`, we see that it takes a relatively higher spatial lattice size N_x to obtain $\langle A(l)\rangle \not\approx 0$ i.e. the bifurcation to non-zero $\langle A(l)\rangle$ is delayed. This is because as we increase `nsweeps`, the Monte Carlo run gets the opportunity to explore a larger part of the configuration space, and the probability to explore the other side of the configuration space increases, hence resulting in $\langle A(l)\rangle \approx 0$ for relatively larger N_x. However, beyond a certain N_x, the Metropolis acceptance probability is too low to explore configuration space even with very large `nsweeps`.

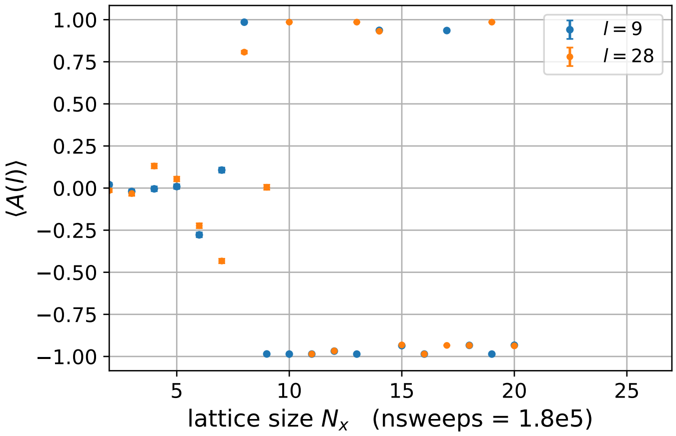


Figure 2.2: $\langle A(l) \rangle$ vs N_x for two different layers $l_1 = 9$, $l_2 = 28$ ran over `nsweeps = 1.8e5`.

2.5.2 Autocorrelation Times of alignment

Another way to demonstrate the expected subsystem symmetry breaking of this system is to look at autocorrelation times of $\langle A(l) \rangle$. Since we expect the system to get stuck in either of the $A(l) \approx \pm 1$ minima, the temporal evolution of $A(l)$ should be highly autocorrelated.

For the ease of writing, we will briefly switch to writing $A(l)$ as A_l. So, for the observable A_l, the autocorrelations are computed as

$$\text{Autocorr}[A_l] = \frac{\langle A_l(k) A_l(k+T) \rangle}{\langle A_l(k)^2 \rangle} = \frac{1}{N-T} \sum_{k=0}^{N-T-1} \frac{A_l(k) \cdot A_l(k+T)}{\langle A_l(k)^2 \rangle} \tag{2.5.1}$$

where the average is taken over the first $N - T$ measurements, i.e.

$$\left\langle A_l(k)^2 \right\rangle = \frac{1}{N-T} \sum_{k=0}^{N-T-1} [A(k)]^2$$

One might notice that this definition of autocorrelation function looks a bit different, as it is computing the sum over the measurements themselves, and not over their deviations i.e. $\langle A_l(k) A_l(k+T) \rangle - \langle A_l(k) \rangle^2$. This is because we have forced $\langle A_l \rangle = 0$, since that is the general expectation due to the subsystem symmetry. If we don't force $\langle A_l \rangle = 0$, we would calculate autocorrelations for the system oscillating in one of the minima wells, which isn't what we desire.

In the following simulation, we have `nsweeps = 2.0e4`, $N_x = 20$, $N_\tau = 40$, $\Delta\tau = 1.0$, and the autocorrelations are computed for $T \in \{$ `1, 2, ..., 3000` $\}$.

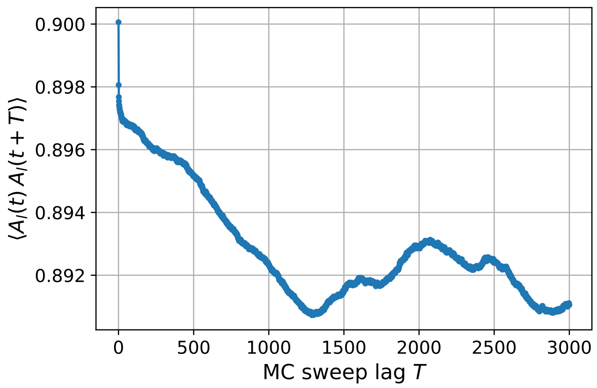


Figure 2.3: Autocorrelation function $\langle A_l(k) A_l(k+T) \rangle$.

As can be seen in Fig. 2.3, the autocorrelation function stays nearly constant ≈ 0.89 even for $T = 3000$, which implies that autocorrelation time $\boxed{\tau \to \infty}$ for $A(l)$. Hence, once the system gets stuck in one of the minimas, it almost never gets the chance to escape it just by using single spin-flips.

2.6 Fixing the subsystem symmetry

To ensure that our subsystem symmetry remains unbroken (or alternatively, to stay in the singlet sector of the Hilbert space), we need the classical system to explore the configuration space equally on both sides of $A(l)$. A possible resolution is to perform fN_x **random spin flip proposals** followed by a **random alignment flip**, where $f \in \mathbb{R}^+$ with $f \to \infty$ limit implying no alignment flips.

The above combination in theory still equilibriates to the Boltzmann distribution because an alignment flip is a $\Delta S = 0$ change, i.e. it is a proposal which is **always accepted** in Metropolis MCMC. This new prescription defines a single Monte Carlo sweep as a combination of $N_x N_\tau$ spin flip proposals and $\lfloor N_\tau / f \rfloor$ alignment flips. We expect alignment flips with $f \sim \mathcal{O}(1)$ to restore the subsystem symmetry in our classical system.

2.6.1 Simulation results with alignment flips

We use the following parameters for the simulation :

`alignment flip fraction` $f =$ `2.0`

`spatial lattice size` $N_x \in \{$`2, 4, 6 ..., 32`$\}$

`no of sampling sweeps` $N_{\text{samp}} =$ `5.0e4`

`imaginary time lattice size` $N_\tau = 40, \quad \Delta\tau = 1.0$

`coupling constants` $K, h = 1.0$

We perform the simulations both with and without alignment flips, and compare the expectation value of the alignment observable $A(l)$.

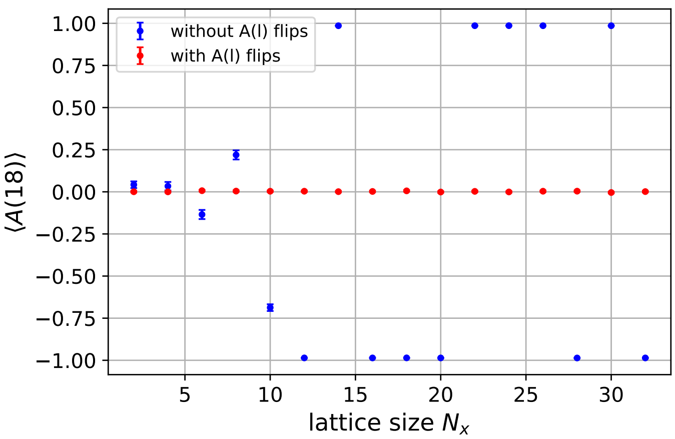
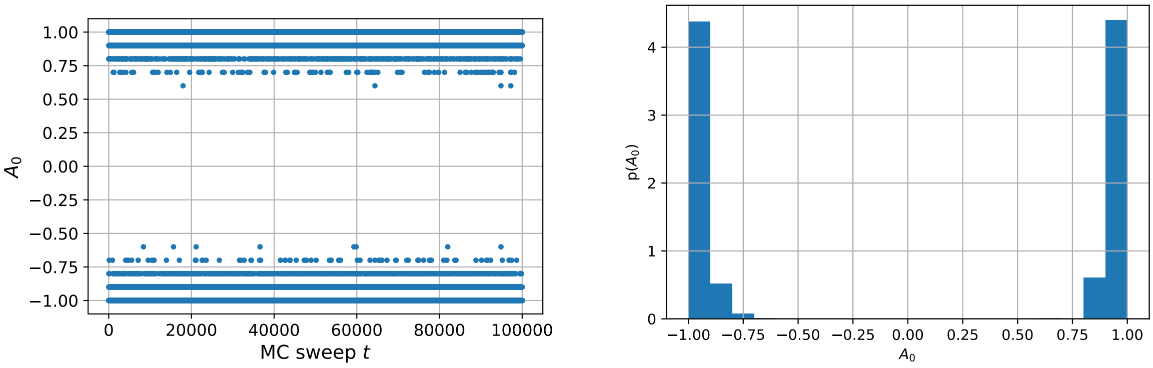


Figure 2.4: $\langle A(l=18)\rangle$ both with and without alignment flips for different spatial lattice sizes.

As expected, the alignment flips make the configuration space equally accessible towards both the $A(l) \approx \pm 1$ sides resulting in $\langle A(l)\rangle \approx 0$ as shown in Fig. 2.4. This is further confirmed if one plots the $A(l)$ measurements as a function of Monte Carlo sweeps for a given layer l when we introduce alignment flips.



(a) $A(l)$ measurements vs MC time.

(b) $A(l)$ measurements histogram.

Figure 2.5: Configuration space exploring both $A(l) > 0$ and $A(l) < 0$ sectors **with** alignment flips.

Fig. 2.5 clearly shows that introducing alignment flips makes the configuration space traverse both the $A(l) > 0$ and $A(l) < 0$ sectors of the state space, hence ensuring that the ergodicity and consequently the subsystem symmetry of the classical system is preserved.

2.6.2 Autocorrelation times

Since we were successfully able to explore both the sides of the configuration space using alignment flips, we consequently expect the autocorrelation times to also go down substantially since the system is no longer stuck.

Since the system isn't stuck in a minima well now, we can now use the usual definition for the autocorrelation function

$$\text{Autocorr}[A_l] = \frac{1}{N-T} \sum_{k=0}^{N-T-1} \frac{(A_l(k) - \langle A_l \rangle) \cdot (A_l(k+T) - \langle A_l \rangle}{\langle A_l(k)^2 \rangle} \tag{2.6.1}$$

where the average is taken over the first $N - T$ measurements, i.e.

$$\left\langle A_l(k)^2 \right\rangle = \frac{1}{N-T} \sum_{k=0}^{N-T-1} [A(k)]^2$$

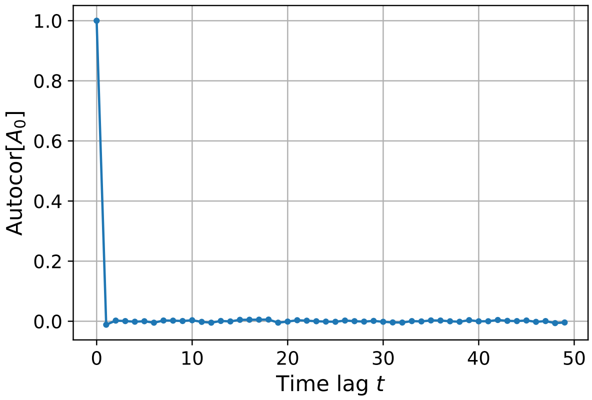


Figure 2.6: Autocorrelation function **with** alignment flips.

As is apparent from Fig. 2.6, the autocorrelations drop extremely quickly, and for this particular simulation we obtained $\boxed{\tau_{\text{int}} = 0.529}$ implying that the alignment measurements $A(l)$ are highly uncorrelated in the Monte Carlo simulation.

2.7 Mapping quantum operators to classical observables

So far, we have been able to conclude that performing Metropolis MCMC as a combination of spin flips and alignment flips keeps the subsystem symmetry of our system intact, and can be a reliable way to perform calculations on using the Quantum to Classical correspondance duality.

However, the only way to cross-check our results is to calculate the expectation value of an operator $\hat{\mathcal{O}}$ in the system described by $\hat{H}_{\text{TFIM}}$ (2.1.1) in the singlet basis using an Exact Diagonalization approach, and compare it to the expectation value of the corresponding classical observable $\mathcal{O}_{\text{classical}}$ using Metropolis MCMC. In this subsection, we'll describe a method of mapping the quantum operators to the corresponding classical observables.

2.7.1 Types of operators in consideration

To begin with, any operator $\hat{\mathcal{O}}$ defined on the singlet sector Hilbert space must satisfy

$$\hat{S}\hat{\mathcal{O}}\hat{S} = \hat{\mathcal{O}} \tag{2.7.1}$$

where $\hat{S} = \prod_i \hat{X}_i$ is the global flip operator. This is because the singlet constraint $\hat{S} = \mathbb{I}$ must be satisfied on $\mathcal{H}_s$. This can also be seen by the following argument - any operator $\mathcal{O}$ defined on $\mathcal{H}_s$ can be written in the outer product notation as

$$\hat{\mathcal{O}} = \sum_{i,j} c_{ij}\hat{P}\left|\{\sigma\}\right\rangle_i \left\langle\{\sigma'\}\right|_j \hat{P}$$

where $\hat{P} = (\mathbb{I} + \hat{S})/2$, so the sandwich between $\hat{S}$'s will give

$$\hat{S}\hat{\mathcal{O}}\hat{S} = \sum_{ij} c_{ij} \underbrace{\hat{S}\hat{P}}_{=\hat{P}} \left|\{\sigma\}\right\rangle_i \left\langle\{\sigma'\}\right|_j \underbrace{\hat{P}\hat{S}}_{=\hat{P}} = \sum_{i,j} c_{ij}\hat{P}\left|\{\sigma\}\right\rangle_i \left\langle\{\sigma'\}\right|_j \hat{P} = \hat{\mathcal{O}}$$

Therefore, any operator on the singlet sector $\mathcal{O} : \mathcal{H}_s \to \mathcal{H}_s$ must satisfy (2.7.1).

Some elementary operators which satisfy the above constraint are

- $\hat{Z}_i\hat{Z}_j$
- $\hat{X}_i$

We can now try mapping such relevant quantum operators to their corresponding classical analogues. We first note that the Hamiltonian is given by the form

$$\hat{H} = -h\sum_i \hat{Z}_i\hat{Z}_j - J\sum_i \hat{X}_i$$

As always, the quantum expectation value of an operator is given by

$$\begin{aligned}\langle\hat{\mathcal{O}}\rangle &= \frac{\mathrm{Tr}\left(e^{-\beta\hat{H}}\hat{\mathcal{O}}\right)}{\mathrm{Tr}\left(e^{-\beta\hat{H}}\right)} \\ &= \frac{1}{\mathcal{Z}}\sum_{\{\sigma_0\}}\sum_{\mu_0,\lambda_0\in\{\pm1\}} \langle\{\mu_0\sigma_0\}|e^{-\Delta\tau\hat{H}}e^{-\Delta\tau\hat{H}}\dots e^{-\Delta\tau\hat{H}}\hat{\mathcal{O}}|\{\lambda_0\sigma_0\}\rangle\end{aligned}$$

Using cyclicity of trace, we can say that $\hat{\mathcal{O}}$ does not have to be in the special position at the end of the operator product, therefore

$$\begin{aligned}\mathcal{Z}\langle\hat{\mathcal{O}}\rangle = &\sum_{\{\sigma_0\},\mu_0,\lambda_0} \langle\{\mu_0\sigma_0\}|e^{-\Delta\tau\hat{H}}e^{-\Delta\tau\hat{H}}\dots e^{-\Delta\tau\hat{H}}\hat{\mathcal{O}}|\{\lambda_0\sigma_0\}\rangle \\ &+\sum_{\{\sigma_0\},\mu_0,\lambda_0} \langle\{\mu_0\sigma_0\}|e^{-\Delta\tau\hat{H}}e^{-\Delta\tau\hat{H}}\dots\hat{\mathcal{O}}e^{-\Delta\tau\hat{H}}|\{\lambda_0\sigma_0\}\rangle \\ &+\sum_{\{\sigma_0\},\mu_0,\lambda_0} \langle\{\mu_0\sigma_0\}|e^{-\Delta\tau\hat{H}}\dots\hat{\mathcal{O}}e^{-\Delta\tau\hat{H}}e^{-\Delta\tau\hat{H}}|\{\lambda_0\sigma_0\}\rangle \\ &\dots+\sum_{\{\sigma_0\},\mu_0,\lambda_0} \langle\{\mu_0\sigma_0\}|\hat{\mathcal{O}}e^{-\Delta\tau\hat{H}}e^{-\Delta\tau\hat{H}}\dots e^{-\Delta\tau\hat{H}}|\{\lambda_0\sigma_0\}\rangle\end{aligned}$$

Defining $\mu_{N_\tau}=\mu_0$, $\lambda_{N_\tau}=\lambda_0$, and $\{\sigma_{N_\tau}\}=\{\sigma_0\}$, and inserting the singlet identity

$$\mathbb{I}_s = \sum_{\{\sigma_l\},\mu_l,\lambda_l} |\{\lambda_l\sigma_l\}\rangle\langle\{\mu_l\sigma_l\}|$$

for $l\in\{1,2,3\dots,N_\tau-1\}$, we get the following expression

$$\mathcal{Z}\langle\hat{\mathcal{O}}\rangle = \left(\prod_{l=0}^{N_\tau-1}\sum_{\{\sigma_l\},\mu_l,\lambda_l}\right)\frac{1}{N_\tau}\sum_{l_0=0}^{N_\tau-1}\left[\langle\{\mu_{l_0+1}\sigma_{l_0+1}\}|e^{-\Delta\tau\hat{H}}\hat{\mathcal{O}}|\{\lambda_{l_0}\sigma_{l_0}\}\rangle\cdot\prod_{l\neq l_0}\langle\{\mu_{l+1}\sigma_{l+1}\}|e^{-\Delta\tau\hat{H}}|\{\lambda_l\sigma_l\}\rangle\right] \tag{2.7.2}$$

2.7.2 Classical analogue of $\hat{Z}_i\hat{Z}_j$

We can now find the classical analogue of the $\hat{\mathcal{O}}=\hat{Z}_i\hat{Z}_j$ by performing the substitution in Eqn. (2.7.2)

$$\begin{aligned}&\mathcal{Z}\langle\hat{Z}_i\hat{Z}_j\rangle \\ &=\left(\prod_{l=0}^{N_\tau-1}\sum_{\{\sigma_l\},\mu_l,\lambda_l}\right)\frac{1}{N_\tau}\sum_{l_0=0}^{N_\tau-1}\left[\langle\{\mu_{l_0+1}\sigma_{l_0+1}\}|e^{-\Delta\tau\hat{H}}\hat{Z}_i\hat{Z}_j|\{\lambda_{l_0}\sigma_{l_0}\}\rangle\cdot\prod_{l\neq l_0}\langle\{\mu_{l+1}\sigma_{l+1}\}|e^{-\Delta\tau\hat{H}}|\{\lambda_l\sigma_l\}\rangle\right] \\ &=\left(\prod_{l=0}^{N_\tau-1}\sum_{\{\sigma_l\},\mu_l,\lambda_l}\right)\frac{1}{N_\tau}\sum_{l_0=0}^{N_\tau-1}\left[\langle\{\mu_{l_0+1}\sigma_{l_0+1}\}|e^{-\Delta\tau\hat{H}}|\{\lambda_{l_0}\sigma_{l_0}\}\rangle\,(\lambda_{l_0}\sigma^i_{l_0})(\lambda_{l_0}\sigma^j_{l_0})\right. \\ &\qquad\qquad\left.\prod_{l\neq l_0}\langle\{\mu_{l+1}\sigma_{l+1}\}|e^{-\Delta\tau\hat{H}}|\{\lambda_l\sigma_l\}\rangle\right] \\ &=\left(\prod_{l=0}^{N_\tau-1}\sum_{\{\sigma_l\},\mu_l,\lambda_l}\right)\left(\frac{1}{N_\tau}\sum_{l_0=0}^{N_\tau-1}\sigma^i_{l_0}\sigma^j_{l_0}\right)\left[\prod_{l=0}^{N_\tau-1}\langle\{\mu_{l+1}\sigma_{l+1}\}|e^{-\Delta\tau\hat{H}}|\{\lambda_l\sigma_l\}\rangle\right]\end{aligned}$$

After manipulations on the $\prod_l$ product of terms, and performing a sum $\sum_{\mu_l,\lambda_l\in\{\pm 1\}}$, we get the usual partition function with the classical basis states

$$\langle \hat{Z}_i \hat{Z}_j \rangle = \frac{1}{\mathcal{Z}} \left(\prod_{l=0}^{N_\tau - 1} \sum_{\{\sigma_l\}} \right) \left(\frac{1}{N_\tau} \sum_{l_0=0}^{N_\tau-1} \sigma^i_{l_0} \sigma^j_{l_0} \right) e^{-S_{\text{eff}}}$$

where $S_{\text{eff}} = -h\Delta\tau \sum_i \sum_l \sigma^i_l \sigma^{i+1}_l - \sum_l \ln\cosh\big(K \sum_i \sigma^i_l \sigma^i_{l+1}\big)$. Therefore, the classical observable corresponding to quantum operator $\hat{Z}_i \hat{Z}_j$ is

$$\boxed{O_{Z_i Z_j} = \frac{1}{N_\tau} \sum_{l=0}^{N_\tau-1} \sigma^i_l \sigma^j_l} \tag{2.7.3}$$

2.7.3 Classical analogue of $\hat{X}_i$

Following a similar procedure as above, we also calculate the classical observable corresponding to $\hat{X}_i$, although the calculation isn't as straightforward here

$$\begin{aligned}
&\mathcal{Z}\langle \hat{X}_i \rangle \\
&= \left(\prod_{l=0}^{N_\tau-1} \sum_{\{\sigma_l\},\mu_l,\lambda_l} \right) \frac{1}{N_\tau} \sum_{l_0=0}^{N_\tau-1} \left[\langle \{\mu_{l_0+1}\sigma_{l_0+1}\} | e^{-\Delta\tau \hat{H}} \hat{X}_i | \{\lambda_{l_0}\sigma_{l_0}\} \rangle \cdot \prod_{l\neq l_0} \langle \{\mu_{l+1}\sigma_{l+1}\} | e^{-\Delta\tau \hat{H}} | \{\lambda_l \sigma_l\} \rangle \right]
\end{aligned} \tag{2.7.4}$$

$$\tag{2.7.5}$$

Let's first analyze the matrix element with $\hat{X}_i$

$$\langle \{\mu_{l_0+1}\sigma_{l_0+1}\} | e^{-\Delta\tau \hat{H}} \hat{X}_i | \{\lambda_{l_0}\sigma_{l_0}\} \rangle \approx \langle \{\mu_{l_0+1}\sigma_{l_0+1}\} | e^{-\Delta\tau \hat{H}_I} e^{-\Delta\tau \hat{H}_T} \hat{X}_i | \{\lambda_{l_0}\sigma_{l_0}\} \rangle$$

The above approximation is valid in the limit $\Delta\tau \to 0$ and $N_\tau \to \infty$. If we make the Ising interaction term $\hat{H}_I$ act on the bra at the left, we get

$$\begin{aligned}
&= \langle \{\mu_{l_0+1}\sigma_{l_0+1}\} | e^{-\Delta\tau \hat{H}_T} \hat{X}_i | \{\lambda_{l_0}\sigma_{l_0}\} \rangle \, e^{-\Delta\tau \hat{H}_I(\{\mu_{l_0+1}\sigma_{l_0+1}\})} \\
&= \langle \{\mu_{l_0+1}\sigma_{l_0+1}\} | e^{-\Delta\tau \hat{H}_T} \hat{X}_i | \{\lambda_{l_0}\sigma_{l_0}\} \rangle \, e^{-\Delta\tau \hat{H}_I(\{\sigma_{l_0+1}\})} \\
&= \langle \{\mu_{l_0+1}\sigma_{l_0+1}\} | e^{-\Delta\tau \hat{H}_T} \hat{X}_i | \{\lambda_{l_0}\sigma_{l_0}\} \rangle \, e^{-\Delta\tau \hat{H}_I(\{\sigma_{l_0}\})}
\end{aligned}$$

Ignoring the Ising term for now, we can write $\langle \{\mu_{l_0+1}\sigma_{l_0+1}\} | e^{-\Delta\tau \hat{H}_T} \hat{X}_i | \{\lambda_{l_0}\sigma_{l_0}\} \rangle$ term as follows

$$\begin{aligned}
&\langle \{\mu_{l_0+1}\sigma_{l_0+1}\} | e^{-\Delta\tau \hat{H}_T} \hat{X}_i | \{\lambda_{l_0}\sigma_{l_0}\} \rangle \\
&= \langle \{\mu_{l_0+1}\sigma_{l_0+1}\} | e^{\Delta\tau \sum_j \hat{X}_j} \hat{X}_i | \{\lambda_{l_0}\sigma_{l_0}\} \rangle \\
&= \left\langle \mu_{l_0+1}\sigma^i_{l_0+1} \middle| e^{\Delta\tau \hat{X}_i} \hat{X}_i \middle| \lambda_{l_0}\sigma^i_{l_0} \right\rangle \prod_{j\neq i} \left\langle \mu_{l_0+1}\sigma^j_{l_0+1} \middle| e^{\Delta\tau \hat{X}_j} \middle| \lambda_{l_0}\sigma^j_{l_0} \right\rangle
\end{aligned} \tag{2.7.6}$$

Now, on simplifying the operator products above,

$$
\begin{aligned}
e^{\Delta\tau J\hat{X}_j} &= \cosh(\Delta\tau J)\mathbb{I} + \sinh(\Delta\tau J)\hat{X}_j \\
e^{\Delta\tau J\hat{X}_i}\hat{X}_i &= \sinh(\Delta\tau J)\mathbb{I} + \cosh(\Delta\tau J)\hat{X}_i
\end{aligned}
$$

Therefore, when we evaluate the matrix products (2.7.6), we get

$$
\begin{aligned}
\left\langle \mu_{l_0+1}\sigma^j_{l_0+1} \middle| e^{\Delta\tau\hat{X}_j} \middle| \lambda_{l_0}\sigma^j_{l_0} \right\rangle &= Ae^{+K\mu_{l_0+1}\lambda_{l_0}\sigma^j_{l_0+1}\sigma^j_{l_0}} \\
\left\langle \mu_{l_0+1}\sigma^i_{l_0+1} \middle| e^{\Delta\tau\hat{X}_i}\hat{X}_i \middle| \lambda_{l_0}\sigma^i_{l_0} \right\rangle &= Ae^{-K\mu_{l_0+1}\lambda_{l_0}\sigma^j_{l_0+1}\sigma^j_{l_0}}
\end{aligned}
$$

i.e. the $\hat{X}$ operator ends up flipping the sign of the factor in the exponential when $j = i, l = l_0$. This is what causes all the difference compared to $\mathcal{Z}$. In the above expressions, $A = \sqrt{\frac{1}{2}\sinh(2J\Delta\tau)}$ and $K = \frac{1}{2}\ln\tanh(J\Delta\tau)$. Combining it all, the matrix product in Eqn. (2.7.6) becomes

$$
\begin{aligned}
&\langle\{\mu_{l_0+1}\sigma_{l_0+1}\}|e^{-\Delta\tau\hat{H}_T}\hat{X}_i|\{\lambda_{l_0}\sigma_{l_0}\}\rangle \\
&= A^{N_x}e^{-K(\mu_{l_0+1}\lambda_{l_0})\sigma^i_{l_0+1}\sigma^i_{l_0}}\prod_{j\neq i}e^{+K(\mu_{l_0+1}\lambda_{l_0})\sigma^j_{l_0+1}\sigma^j_{l_0}} \\
&= A^{N_x}\exp\left(-2K(\mu_{l_0+1}\lambda_{l_0})\sigma^i_{l_0+1}\sigma^i_{l_0}\right)\exp\left(K(\mu_{l_0+1}\lambda_{l_0})\sum_{j=0}^{N_x-1}\sigma^j_{l_0+1}\sigma^j_{l_0}\right) \qquad (2.7.7)
\end{aligned}
$$

For all other layers $l \neq l_0$, the other matrix element (without $\hat{X}_i$) from (2.7.5) is similarly calculated as

$$
\langle\{\mu_{l+1}\sigma_{l+1}\}|e^{-\Delta\tau\hat{H}_T}|\{\lambda_l\sigma_l\}\rangle = A^{N_x}\exp\left(K(\mu_{l+1}\lambda_l)\sum_{j=0}^{N_x-1}\sigma^j_{l+1}\sigma^j_l\right) \qquad (2.7.8)
$$

Combining the matrix elements of both (2.7.7) and (2.7.8), we can write the expectation value sum as follows

$$
\begin{aligned}
&\mathcal{Z}\langle\hat{X}_i\rangle \\
&\sim \left(\prod_{l=0}^{N_\tau-1}\sum_{\{\sigma_l\},\mu_l,\lambda_l}\right)\left(\frac{1}{N_\tau}\sum_{l_0=0}^{N_\tau-1}e^{-2K\mu_{l_0+1}\lambda_{l_0}\sigma^i_{l_0}\sigma^i_{l_0+1}}\right)\prod_{l=0}^{N_\tau-1}e^{K\mu_{l+1}\lambda_l\sum_j\sigma^j_{l+1}\sigma^j_l+h\Delta\tau\sum_j\sigma^j_l\sigma^{j+1}_l}
\end{aligned}
$$

Since the expectation value $\langle\hat{X}_i\rangle$ only depends on the product $\mu_{l+1}\lambda_l$, and we are summing over $\lambda_l, \mu_{l+1} \in \{\pm 1\}$, we can replace it by just μ_l with the values lying $\in \{\pm 1\}$.

$$\sim \left(\prod_{l=0}^{N_\tau-1}\sum_{\{\sigma_l\}}\right)\frac{1}{N_\tau}\sum_{l_0=0}^{N_\tau-1}\left[\left(\sum_{\mu_{l_0}} e^{K\mu_{l_0+1}\sum_j \sigma^j_{l_0+1}\sigma^j_{l_0} - 2K\mu_{l_0+1}\sigma^i_{l_0}\sigma^i_{l_0+1}}\right)e^{h\Delta\tau\sum_j\sigma^j_{l_0}\sigma^{j+1}_{l_0}}\right.$$
$$\left.\prod_{l\neq l_0}\sum_{\mu_l} e^{K\mu_{l+1}\sum_j\sigma^j_{l+1}\sigma^j_l + h\Delta\tau\sum_j\sigma^j_l\sigma^{j+1}_l}\right]$$
$$\sim \left(\prod_{l=0}^{N_\tau-1}\sum_{\{\sigma_l\}}\right)\frac{1}{N_\tau}\sum_{l_0=0}^{N_\tau-1}\left[\cosh\left(K\sum_j\sigma^j_{l_0+1}\sigma^j_{l_0} - 2K\sigma^i_{l_0}\sigma^i_{l_0+1}\right)e^{h\Delta\tau\sum_j\sigma^j_{l_0}\sigma^{j+1}_{l_0}}\right.$$
$$\left.\prod_{l\neq l_0}\cosh\left(K\sum_j\sigma^j_{l+1}\sigma^j_l\right)e^{h\Delta\tau\sum_j\sigma^j_l\sigma^{j+1}_l}\right] \tag{2.7.9}$$

Finally, we can expand the $\cosh(x-y)$ term in Eqn. (2.7.9), take out $\cosh\left(K\sum_j\sigma^j_{l_0+1}\sigma^j_{l_0}\right)$ $e^{h\Delta\tau\sum_j\sigma^j_{l_0}\sigma^{j+1}_{l_0}}$ as a common factor, and restore the $\prod_{l\neq l_0}$ to $\prod_{l=0}^{N_\tau-1}$. The product then converts into the usual $e^{-S_{\text{eff}}}$ leaving behind the classical observable

$$\mathcal{Z}\langle\hat{X}_i\rangle \sim \left(\prod_{l=0}^{N_\tau-1}\sum_{\{\sigma_l\}}\right)\frac{1}{N_\tau}\sum_{l_0=0}^{N_\tau-1}\left(\cosh\left(2K\sigma^i_{l_0}\sigma^i_{l_0+1}\right) - \tanh\left(K\sum_j\sigma^j_{l_0+1}\sigma^j_{l_0}\right)\sinh\left(2K\sigma^i_{l_0}\sigma^i_{l_0+1}\right)\right)e^{-S_{\text{eff}}} \tag{2.7.10}$$

Therefore, we can now extract the classical observable corresponding to the quantum operator $\hat{X}_i$ as

$$\boxed{O_{X_i} = \frac{1}{N_\tau}\sum_{l=0}^{N_\tau-1}\left[\cosh\left(2K\sigma^i_l\sigma^i_{l+1}\right) - \tanh\left(K\sum_{j=0}^{N_x-1}\sigma^j_{l+1}\sigma^j_l\right)\sinh\left(2K\sigma^i_l\sigma^i_{l+1}\right)\right]} \tag{2.7.11}$$

One can also write it a little compactly as

$$\boxed{O_{X_i} = \frac{1}{N_\tau}\sum_{l=0}^{N_\tau-1}\left[\frac{\cosh\left(2K\sigma^i_l\sigma^i_{l+1} - K\sum_{j=0}^{N_x-1}\sigma^j_{l+1}\sigma^j_l\right)}{\cosh\left(K\sum_{j=0}^{N_x-1}\sigma^j_{l+1}\sigma^j_l\right)}\right]} \tag{2.7.12}$$

2.8 Singlet sector with $N_x = 2$

As a first check for our algorithm, we start by verifying the results for the 2 site problem i.e. $N_x = 2$. For the 2 site problem, we can calculate the Monte Carlo expectation values of $O_{Z_iZ_j}$ and O_{X_i}, and compare it with the analytical expressions for the expectation values of $\hat{Z}_i\hat{Z}_j$ and $\hat{X}_i$ by diagonalizing the 2×2 Hamiltonian in the singlet basis.

2.8.1 2 site problem in the quantum realm

For 2 spin sites, the Hilbert space can be defined to be spanned by the singlet basis

$$\mathcal{H}_s = \text{span}\left\{\frac{|\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle}{\sqrt{2}}, \frac{|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle}{\sqrt{2}}\right\}$$

We label the above states as

$$|a\rangle = \frac{|\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle}{\sqrt{2}}, \qquad |b\rangle = \frac{|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle}{\sqrt{2}}$$

The expression for the expectation value of an operator is given by

$$\langle\hat{O}\rangle = \text{Tr}\left(e^{-\beta\hat{H}}\hat{O}\right) = \frac{e^{-\beta E_0}\langle w_0|\hat{O}|w_0\rangle + e^{-\beta E_1}\langle w_1|\hat{O}|w_1\rangle}{e^{-\beta E_0} + e^{-\beta E_1}}$$

where we have evaluated the trace in the yet-unknown energy eigenbasis. To now calculate the energy eigenvalues E_i and the eigenvectors $|w_i\rangle$, we can diagonalize the Hamiltonian in the singlet basis representation.

$$H := \begin{pmatrix} \langle a|\hat{H}|a\rangle & \langle a|\hat{H}|b\rangle \\ \langle b|\hat{H}|a\rangle & \langle a|\hat{H}|b\rangle \end{pmatrix}$$

For $N_x = 2$, the quantum Hamiltonian with periodic boundary conditions just becomes

$$\hat{H} = -2h\hat{Z}_0\hat{Z}_1 - J(\hat{X}_0 + \hat{X}_1)$$

and after a bit of algebra, the matrix representation in the singlet basis comes out to be

$$H := \begin{pmatrix} -2h & -2J \\ -2J & 2h \end{pmatrix}$$

This results in the eigenvalues of $\hat{H}$ as

$$\lambda_\pm = \pm 2\sqrt{h^2 + J^2}$$

If we now define $\boxed{h = B\cos\theta}$ and $\boxed{J = B\sin\theta}$, then $\lambda_\pm = \pm 2B$, and the $\hat{H}$ matrix becomes

$$H := 2B\begin{pmatrix} -\cos\theta & -\sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

Now to find the eigenvectors, say that one of them is given by

$$|w_0\rangle = u\,|a\rangle + v\,|b\rangle$$

This implies that the other eigenvector must be of the form

$$|w_1\rangle = -v\,|a\rangle + u\,|b\rangle$$

with u and $v \in \mathbb{R}$. So, for the eigenvector $|w_0\rangle$ with the eigenvalue $E_0 = -2B$, the matrix equation looks like

$$2B \begin{pmatrix} -\cos\theta & -\sin\theta \\ -sin\theta & \cos\theta \end{pmatrix} = -2B \begin{pmatrix} u \\ v \end{pmatrix}$$
$$2B \begin{pmatrix} -\cos\theta + 1 & -\sin\theta \\ -\sin\theta & \cos\theta + 1 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$
$$2B \begin{pmatrix} 2\sin^2(\theta/2) & -2\sin(\theta/2)\cos(\theta/2) \\ -2\sin(\theta/2)\cos(\theta/2) & 2\cos^2(\theta/2) \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Solving for u and v, we obtain the relation

$$\sin(\theta/2)u = \cos(\theta/2)v$$

so we choose

$$\boxed{u = \cos(\theta/2), \qquad v = \sin(\theta/2)}$$

to make sure the eigenvectors normalize to 1.

Now, let's calculate the matrix elements $\langle w_i|\hat{O}|w_i\rangle$ for $\hat{O} = \hat{X}_i$, which we'll finally use to calculate the thermal expectation value of $\hat{X}_i$.

$$\begin{aligned} \langle w_0|\hat{X}_0|w_0\rangle &= (\langle a| u + \langle b| v)\hat{X}_i(u|a\rangle + v|b\rangle) \\ &= u^2 \langle a| \underbrace{\hat{X}_0 |a\rangle}_{|b\rangle} + uv \langle a| \underbrace{\hat{X}_0 |b\rangle}_{|a\rangle} + uv \langle b| \underbrace{\hat{X}_0 |a\rangle}_{|b\rangle} + v^2 \langle b| \underbrace{\hat{X}_0 |b\rangle}_{|a\rangle} \\ &= u^2 \langle a|b\rangle + uv \langle a|a\rangle + uv \langle b|b\rangle + v^2 \langle b|a\rangle \\ &= 2uv \end{aligned}$$

$$\begin{aligned} \langle w_1|\hat{X}_0|w_1\rangle &= (-\langle a| v + \langle b| u)\hat{X}_i(-v|a\rangle + u|b\rangle) \\ &= v^2 \langle a| \underbrace{\hat{X}_0 |a\rangle}_{|b\rangle} - uv \langle a| \underbrace{\hat{X}_0 |b\rangle}_{|a\rangle} - uv \langle b| \underbrace{\hat{X}_0 |a\rangle}_{|b\rangle} + u^2 \langle b| \underbrace{\hat{X}_0 |b\rangle}_{|a\rangle} \\ &= v^2 \langle a|b\rangle - uv \langle a|a\rangle - uv \langle b|b\rangle + u^2 \langle b|a\rangle \\ &= -2uv \end{aligned}$$

The calculation is exactly similar for $\hat{X}_1$. The expectation value for $\hat{X}$ then evaluates to

$$\begin{aligned} \langle\hat{X}\rangle &= 2uv\left(\frac{e^{2\beta B} - e^{-2\beta B}}{e^{2\beta B} + e^{-2\beta B}}\right) = 2uv\tanh(2\beta B) \\ \implies & \boxed{\langle\hat{X}\rangle = \frac{J}{B}\tanh(2\beta B)} \end{aligned} \tag{2.8.1}$$

The calculation for $\hat{Z}_0\hat{Z}_1$ is analogous and the matrix elements are evaluated as follows

$$\begin{aligned}
\langle w_0|\hat{Z}_0\hat{Z}_1|w_0\rangle &= (\langle a|\, u + \langle b|\, v)\hat{Z}_0\hat{Z}_1(u\,|a\rangle + v\,|b\rangle) \\
&= u^2\,\langle a|\,\underbrace{\hat{Z}_0\hat{Z}_1\,|a\rangle}_{|a\rangle} + uv\,\langle a|\,\underbrace{\hat{Z}_0\hat{Z}_1\,|b\rangle}_{-|b\rangle} + uv\,\langle b|\,\underbrace{\hat{Z}_0\hat{Z}_1\,|a\rangle}_{|a\rangle} + v^2\,\langle b|\,\underbrace{\hat{Z}_0\hat{Z}_1\,|b\rangle}_{-|b\rangle} \\
&= u^2\,\langle a|a\rangle - uv\,\langle a|b\rangle + uv\,\langle b|a\rangle - v^2\,\langle b|b\rangle \\
&= u^2 - v^2
\end{aligned}$$

$$\begin{aligned}
\langle w_1|\hat{Z}_0\hat{Z}_1|w_1\rangle &= (-\,\langle a|\, v + \langle b|\, u)\hat{Z}_0\hat{Z}_1(-v\,|a\rangle + u\,|b\rangle) \\
&= v^2\,\langle a|\,\underbrace{\hat{Z}_0\hat{Z}_1\,|a\rangle}_{|a\rangle} - uv\,\langle a|\,\underbrace{\hat{Z}_0\hat{Z}_1\,|b\rangle}_{-|b\rangle} - uv\,\langle b|\,\underbrace{\hat{Z}_0\hat{Z}_1\,|a\rangle}_{|a\rangle} + u^2\,\langle b|\,\underbrace{\hat{Z}_0\hat{Z}_1\,|b\rangle}_{-|b\rangle} \\
&= v^2\,\langle a|a\rangle - uv\,\langle a|b\rangle + uv\,\langle b|a\rangle - u^2\,\langle b|b\rangle \\
&= -(u^2 - v^2)
\end{aligned}$$

Therefore, the expectation value of $\hat{Z}_0\hat{Z}_1$ evaluates to

$$\langle\hat{Z}_0\hat{Z}_1\rangle = (u^2 - v^2)\left(\frac{e^{2\beta B} - e^{-2\beta B}}{e^{2\beta B} + e^{-2\beta B}}\right) = (u^2 - v^2)\tanh(2\beta B)$$

$$\implies \boxed{\langle\hat{Z}_0\hat{Z}_1\rangle = \frac{h}{B}\tanh(2\beta B)} \qquad (2.8.2)$$

2.8.2 2 site problem in the classical realm

Now that we have the analytical expressions for $\langle\hat{X}\rangle$ and $\langle\hat{Z}_0\hat{Z}_1\rangle$, we can compare the results with $\langle O_X\rangle$ and $\langle O_{Z_0Z_1}\rangle$ i.e. the expectation values of the corresponding classical observables using Monte Carlo.

The quantum-to-classical correspondance effective action for $N_x = 2$ simplifies to

$$S = -2h\Delta\tau\sum_{l=1}^{N_\tau}\sigma_l^0\sigma_l^1 - \sum_{l=1}^{N_\tau}\ln\cosh\left[K\left(\sigma_l^0\sigma_{l+1}^0 + \sigma_l^1\sigma_{l+1}^1\right)\right]$$

Thus, for a spin flip $\sigma_{l_0}^{i_0} \to -\sigma_{l_0}^{i_0}$, we can decompose the change in action as

$$\Delta S_x = 4h\Delta\tau\sigma_{l_0}^0\sigma_{l_0}^1$$

$$\Delta S_\tau = \ln\left[\frac{\cosh\left[K(\sigma_l^0\sigma_{l+1}^0 + \sigma_l^1\sigma_{l+1}^1)\right]\cdot\cosh\left[K(\sigma_l^0\sigma_{l-1}^0 + \sigma_l^1\sigma_{l-1}^1)\right]}{\cosh\left[K(\sigma_l^0\sigma_{l+1}^0 - \sigma_l^1\sigma_{l+1}^1)\right]\cdot\cosh\left[K(\sigma_l^0\sigma_{l-1}^0 - \sigma_l^1\sigma_{l-1}^1)\right]}\right]$$

$$\Delta S = \Delta S_x + \Delta S_\tau$$

As before, the relevant operators are the mappings of $\hat{X}_i$ and $\hat{Z}_0\hat{Z}_1$ onto classical observables O_X and $O_{Z_0Z_1}$ respectively.

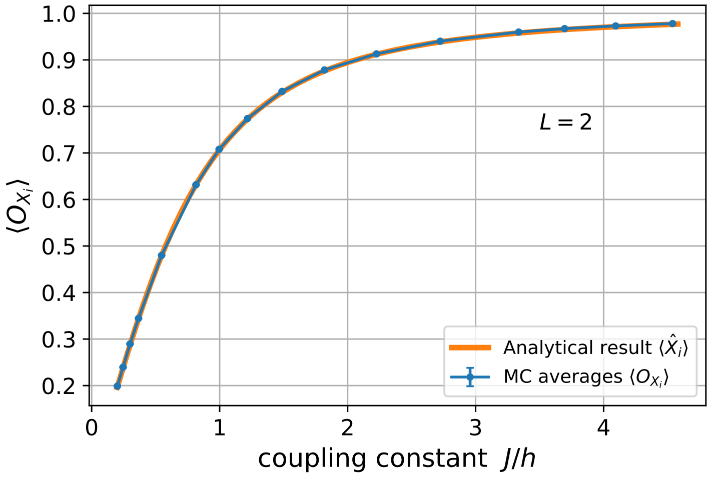
$$O_{Z_0Z_1} = \frac{1}{N_\tau}\sum_{l=0}^{N_\tau-1}\sigma_l^0\sigma_l^1$$

$$O_X = O_{X_0} = O_{X_1} = \frac{1}{N_\tau}\sum_{l=0}^{N_\tau-1}\left[\frac{\cosh\left(K\sigma_l^0\sigma_{l+1}^0 - K\sigma_l^1\sigma_{l+1}^1\right)}{\cosh\left(K\sigma_l^0\sigma_{l+1}^0 + K\sigma_l^1\sigma_{l+1}^1\right)}\right]$$

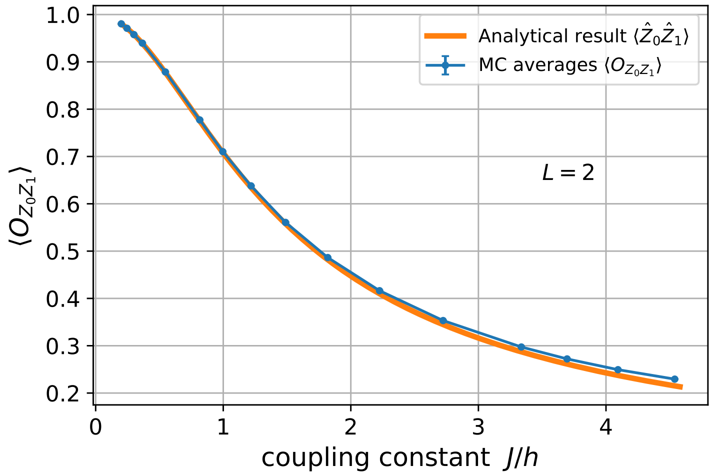
Therefore, we can now use Metropolis Monte Carlo with an acceptance probability of $\min(1, e^{-\Delta S})$ for spin flips, combined with alignment flips. Finally, we compare the MC expectation values of O_X and $O_{Z_0Z_1}$ with the analytical results for $\langle\hat{X}\rangle$ and $\langle\hat{Z}_0\hat{Z}_1\rangle$.

We perform the Monte Carlo simulation for $\beta = N_\tau = 50$, keeping $\Delta\tau = 1$ and alignment flip fraction $f = \texttt{1.0}$. However, care needs to be taken while choosing the values of J (consequently K) and h. Since the phase diagram only depends on the ratio J/h, we have the freedom of choosing the value of the parameters as large or as small as we want. To explore the system near the critical point ($J/h = 1$), we have to adjust the parameter values accordingly. If h is too small, then the values of K near the critical point are too large and that results in a low *acceptance rate* and high *trotter error*. The values of K near the critical point are small if h is large, but it again results in the same problem. On top of that, having disproportionate orders of magnitude of K and h can lead to freezing of spins in one of the directions.

Therefore, we require an optimum value of the parameters K and h to explore the system near the critical point so as to minimize the *trotter error* and maximize the *acceptance rate*. Therefore, we choose $h = \texttt{0.05}$ and vary K. The quantum critical point appears at $K_c = 1.498$, so we choose values of K so as to explore the system near this critical point.



(a) Monte Carlo average $\langle O_X\rangle$ as a function of J/h.



(b) Monte Carlo average $\langle O_{Z_0 Z_1}\rangle$ as a function of J/h.

Figure 2.7: Comparing analytical results obtained by exactly diagonalizing the quantum Hamiltonian (d) with the numerical results obtained from performing Monte Carlo on the effective classical model ($d + 1$).

As can be seen from Fig. 2.7, the Monte Carlo averages of the corresponding classical observables matches with the thermal expectation values of the quantum operators. At higher values of J/h, we can see some deviations arising due to the *trotter error* despite the high MC acceptance rate.

Therefore, the algorithm made a first pass on the 2-site problem, and we expect the algorithm to work for higher values of N_x as well.

2.9 Comparing Monte Carlo with Exact Diagonalization

Since it is not practical to calculate the analytical expressions for operator thermal expectation values, we perform an Exact Diagonalization (ED) procedure on the quantum Hamiltonian, and calculate the thermal expectation values as

$$\langle\hat{\mathcal{O}}\rangle = \frac{1}{Z}\,\mathrm{Tr}\Big(e^{-\beta\hat{H}}\hat{\mathcal{O}}\Big) = \frac{\sum_i \langle E_i|\hat{\mathcal{O}}|E_i\rangle\; e^{-\beta E_i}}{\sum_i \langle E_i|E_i\rangle\, e^{-\beta E_i}}$$

where we calculate the trace over the energy eigenbasis. We then compare these quantum thermal expecatation values with the Monte Carlo expectation values of the corresponding classical observables.

For the purposes of our model, $N_x = 13$ is an important lattice size because that is where the subsystem symmetry starts breaking (alignment freezing), and we have to introduce alignment flips to restore ergodicity. Therefore, we compare our ED results for $N_x = 13$ with the MC results, at $\beta = 50$, hence doing sufficiently low-temperature studies.

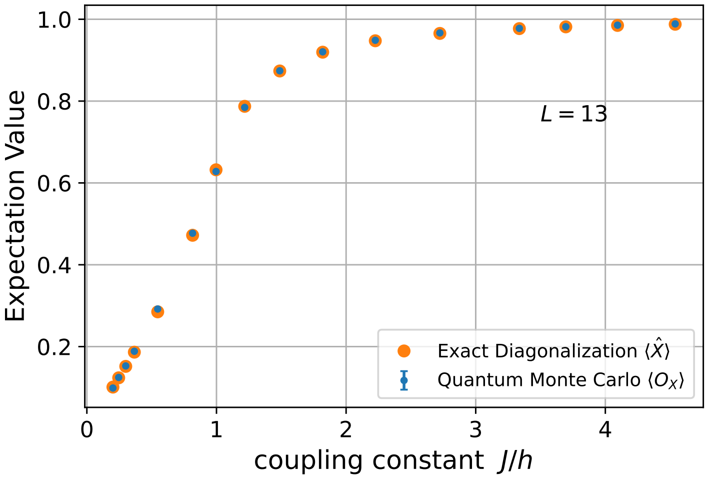
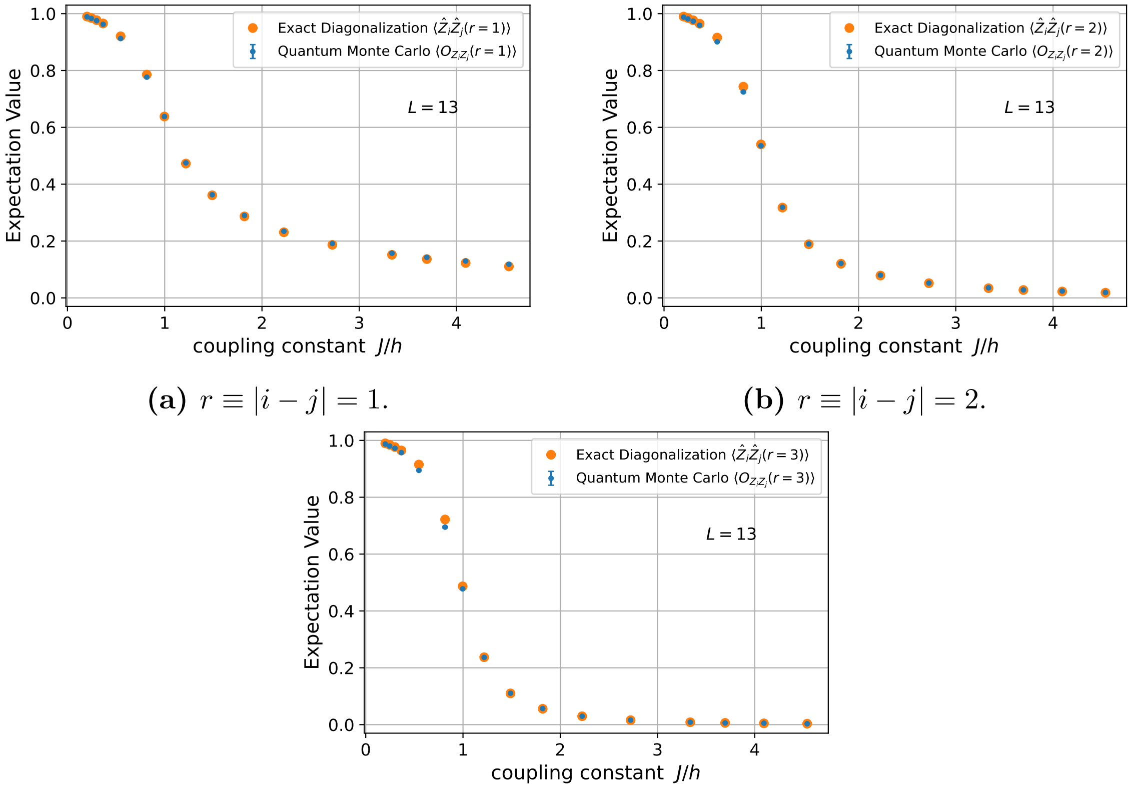


Figure 2.8: Comparing ED results for $\langle \hat{X} \rangle$ with MC average $\langle O_X \rangle$ vs. J/h.



(a) $r \equiv |i - j| = 1$.

(b) $r \equiv |i - j| = 2$.

(c) $r \equiv |i - j| = 3$.

Figure 2.9: Comparing ED results for $\langle \hat{Z}_i \hat{Z}_j \rangle$ with MC average $\langle O_{Z_i Z_j} \rangle$ vs. J/h.

As can be seen, the results from ED and MC simulations match closely, except for the slight variations in $Z_i Z_j$ expectation values as $|i - j|$ increases, which appear due to periodic boundary effects.

Need to explain exact diagonalization procedure. Add notes.

2.10 Simulations in the limit $\beta \to 0$

With our machinery set up, we can now attempt to simulate systems in the $\beta < 1$ (or $T > 0$) range. However, on staring at the action S of the system for long enough,

$$S = -h\Delta\tau \sum_{i=0}^{N_x-1} \sum_{l=0}^{N_\tau-1} \sigma_l^i \sigma_l^{i+1} - \sum_{l=0}^{N_\tau-1} \ln\cosh\left(KN_x A(l)\right)$$

we realize that for a fixed N_x and N_τ, the action primarily depends on two parameters

$$Y \equiv h\Delta\tau, \qquad K \equiv -\frac{1}{2}\ln\tanh(J\Delta\tau)$$

We can rewrite our coupling constants in terms of the above. Therefore, we realize that keeping the same values of Y, K and N_τ, we essentially end up solving an entire class of problems (parameterized by the value of $\Delta\tau$) with the following parameters

$$h = \frac{Y}{\Delta\tau}, \qquad J = \frac{\operatorname{arctanh}(e^{-2K})}{\Delta\tau}, \qquad \beta = N_\tau\Delta\tau.$$

Taking a concrete example, say we perform a MC simulation with $N_\tau = 50$, $h = 0.05$, $K = 2.0$, and $\Delta\tau = 1$. This sets $Y = 0.05$, $K = 2.0$ and $N_\tau = 50$ as fixed values, and the results of this simulation are essentially valid for a class of problems with

$$h = \frac{0.05}{\Delta\tau}, \qquad J = \frac{0.18}{\Delta\tau}, \qquad \beta = 50\Delta\tau.$$

with any arbitatry $\Delta\tau$. If we choose $\Delta\tau = 0.01$, then our simulation results are valid for the set of parameters $h = 5$, $J = 18$, and $\beta = 0.5$ (or $T = 2$), hence obtaining the results of a high-temperature problem from an equivalent low temperature simulation (with different values of h, J).

Bibliography

[1] Somendra M Bhattacharjee and Flavio Seno. “A measure of data collapse for scaling”. In: *Journal of Physics A: Mathematical and General* 34.33 (Aug. 2001), pp. 6375–6380. DOI: 10.1088/0305-4470/34/33/302. URL: https://doi.org/10.1088%2F0305-4470%2F34%2F33%2F302.