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

March 27, 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 generate non-uniformly sampled 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.

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 is more efficient. 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.

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 instead 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 distri-

bution $\rho(x,p) = e^{-\beta H(x,p)}$, 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 $i = 1, 2, \ldots M$,

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

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

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

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 attempt and the proposal acceptance probabilities. Since the proposal probabilities should be uniform, the detailed balance condition in terms of acceptance probabilities becomes

$$\frac{P_{\text{accept}}(X_j \to X_i)}{P_{\text{accept}}(X_i \to X_j)} = \frac{\rho(X_i)}{\rho(X_j)} \tag{1.1.11}$$

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. (1.1.11) as

$$P_{\text{accept}}(X_i \to X_j) = \min\left[\frac{\rho(X_j)}{\rho(X_i)}, 1\right] \tag{1.1.12}$$

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

$$P_{\text{accept}}(E_i \to E_j) = \min\left[e^{-\beta(E_j - E_i)}, 1\right]. \tag{1.1.13}$$

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. (1.1.13) 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 Ising model. The Ising model is the paradigm model for systems exhibiting continuous phase transition from a ferromagnetic to a paramagnetic state at critical temperature T_c. In the absence of an external field ($h = 0$) and only nearest-neighbor interactions, the energy of the 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 can be more suggestively written as

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

H

Chapter 2

Classical Ising Model

2.1 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 using the autocorrelated measurements.

Part 3 The main Monte Carlo run with observable measurements being taken after every 2τ 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.

2.2 Wolff Cluster algorithm

To perform the Monte Carlo sweeps, we use the Wolff Cluster updates which satisfies the required detailed balance condition. The algorithm that we use to implement this update scheme is outlined below, and is taken from [3].

algorithm wolff-cluster
begin
$i :=$ random particle;
$\mathcal{C} := \{i\}$;
$\mathcal{F}_{\text{old}} := \{i\}$;

```
while F_old ≠ {} do
begin
    F_new := {};
    for ∀ i ∈ F_old do
    begin
        for ∀ j neighbor of i with S_i = S_j, j ∉ C do
        begin
            if ran[0,1] < p then
            begin
                F_new := F_new ∪ {j};
                C := C ∪ {j};
            end
        end
    end
    F_old := F_new;
end
for ∀ i ∈ C do
S_i := −S_i;
end
```

The above algorithm shows how a single cluster update/flip is implemented and is termed as one MC sweep.

2.3 Autocorrelation Time

Since Monte Carlo in general generates statistically correlated configurations (and measurements), we take measurements after every few time steps to get statistically uncorrelated measurements. The *autocorrelation function* gives us a measure of correlations between subsequent measurements. For an observable $\mathcal{O}$, the autocorrelation function is

$$A_{\mathcal{O}}(t) = \frac{1}{N-t} \sum_{k=0}^{N-t-1} \frac{(\mathcal{O}(k) - \langle \mathcal{O} \rangle) \cdot (\mathcal{O}(k+t) - \langle \mathcal{O} \rangle)}{\langle \mathcal{O}^2 \rangle - \langle \mathcal{O} \rangle^2} \tag{2.3.1}$$

where $\mathcal{O}(k)$'s are the measurements taken for each Monte Carlo step, and N is the total number of Monte Carlo sampling steps. Also, the averages in (2.3.1) are only taken over the first $N - t$ measurements.

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

The autocorrelation function $A_{\mathcal{O}}(t)$ is expected to fall off as an exponential

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

where τ is defined as the *autocorrelation time*. For an observable $\mathcal{O}$, we define the *integrated autocorrelation time* τ_{int} as a discretized integration sum

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

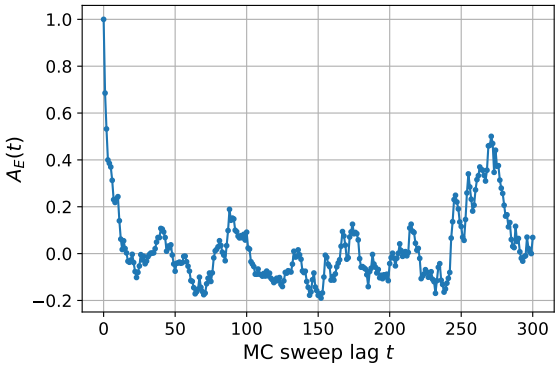


Figure 2.1: Energy density autocorrelation function $A_E(t)$ for $L = 128$, $T = 2.26$.

However, this natural estimator isn't *quite* correct because the autocorrelations $A_{\mathcal{O}}(t)$ for $t \gg \tau_{\text{int}}$ contain much noise but little signal. To fix this, we implement the *automatic windowing algorithm* [5] by introducing a cutoff M on the sum

$$\tau_{\text{int},\mathcal{O}} = \frac{1}{2} + \sum_{t=1}^{M} A_{\mathcal{O}}(t) \tag{2.3.2}$$

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

The autocorrelation time τ_{int} of a given (L, T) configuration is the maximum of autocorrelation times for all observables except magnetization m (due to lack of SSB in finite lattices).

$$\tau_{\text{int}} = \max\left\{\tau_{\text{int},|m|},\ \tau_{\text{int},m^2},\ \tau_{\text{int},m^4},\ \tau_{\text{int},E},\ \tau_{\text{int},E^2}\right\} \tag{2.3.3}$$

After running the preliminary MC run (Part 1) with measurements taken after every MC step to calculate the integrated autocorrelation time τ_{int}, we perform the main MC run (Part 3) with measurements taken after every $2\tau_{\text{int}}$ steps.

2.4 Jackknife 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

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

Below is a visual representation of the binning process

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

$$\Downarrow$$

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

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

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

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

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

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

$$\overline{\mathcal{O}} = \overline{\overline{\mathcal{O}}_j^{(B)}} = \frac{1}{N_B}\sum_{i=0}^{N_B-1}\overline{\mathcal{O}}_j^{(B)} \tag{2.4.1a}$$

$$\overline{\rho} = \overline{\rho_j^{(B)}} = \frac{1}{N_B}\sum_{i=0}^{N_B-1}\rho_j^{(B)} \tag{2.4.1b}$$

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

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

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

The error calculated using the Jackknife bins then comes out as

$$\delta\overline{\mathcal{O}} = \sqrt{\frac{N_B - 1}{N_B} \sum_{j=0}^{N_B-1} \left(\overline{\mathcal{O}}_j^{(J)} - \overline{\mathcal{O}}\right)^2} \tag{2.4.2a}$$

$$\delta\rho = \sqrt{\frac{N_B - 1}{N_B} \sum_{j=0}^{N_B-1} \left(\overline{\rho}_j^{(J)} - \overline{\rho}\right)^2} \tag{2.4.2b}$$

2.5 Simulation results

The following parameters were used in the 2D Ising model simulations for lattice sizes of $L = 8, 16, 32, 64, 128$.

For preliminary Monte Carlo (Part 1)

`no of equilibration sweeps = 1.0e3`

`no of sampling sweeps = 1.0e4`

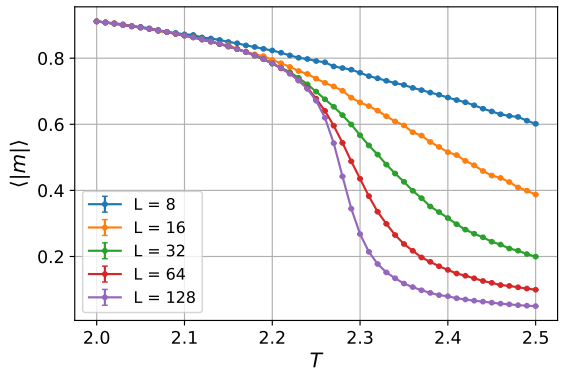
`sampling step size = 1.0e0`

For the main Monte Carlo run (Part 3)

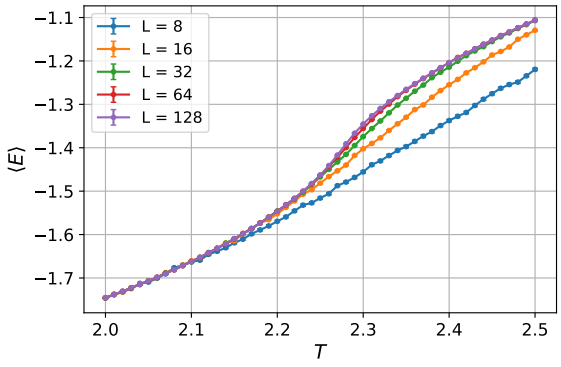
`no of equilibration sweeps = 1.0e3`

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

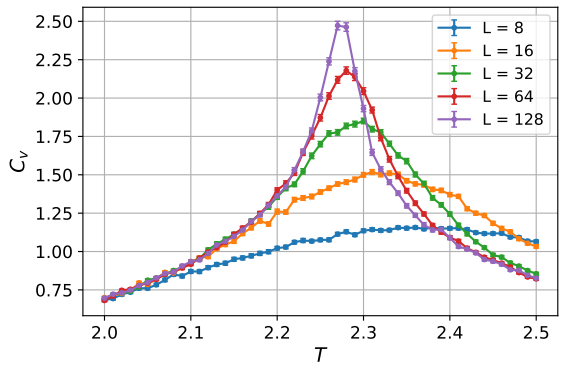
`sampling step size =` 2τ



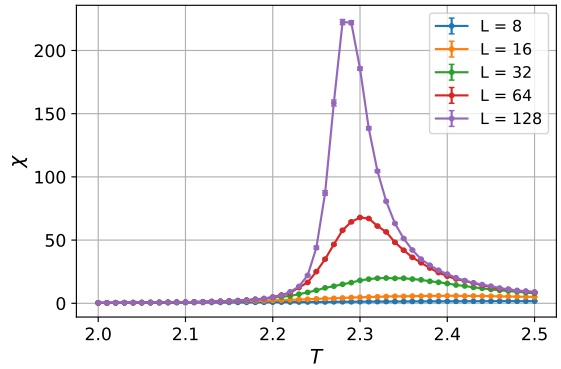
(a) $|M|$ vs T



(b) E vs T

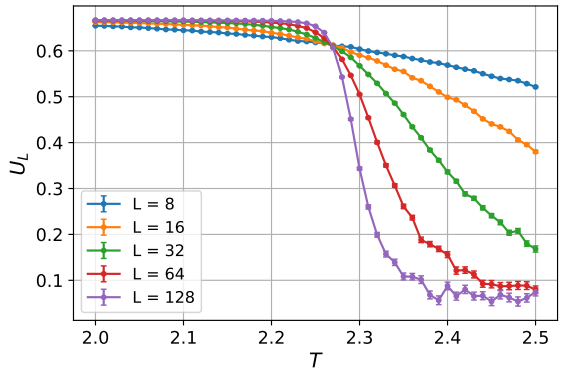


(c) C_V vs T

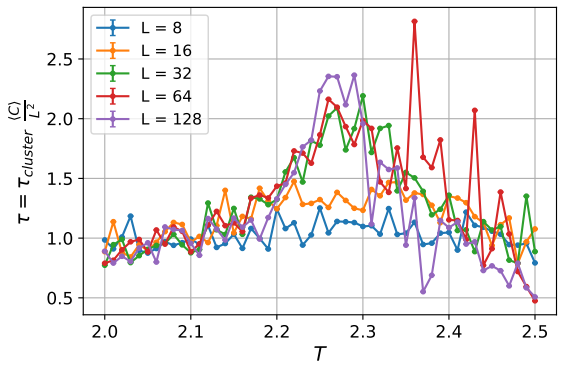


(d) χ vs T

Figure 2.2: Variation of physical quantities with T for different lattice sizes L.



(e) U_L vs T



(f) τ_{int} vs T

Figure 2.2: Variation of physical quantities with T for different lattice sizes L.

The Figure 2.2 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.

Furthermore, the critical temperature T_c is obtained by finding the point of intersection

of different U_L vs T curves, $\implies T_c \approx 2.269$.

2.6 Finite-size scaling

According to the FSS hypothesis [4], observables and derived physical quantities close to T_c are a power of L multiplied by a non divergent function of $tL^{1/\nu}$, i.e.

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

where $t = (T - T_c)/T_c$.

FS relations for 2D Ising model

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

We use the *data-collapse* method to estimate the critical exponents β, γ, and ν ($\alpha = 0$ for 2D Ising model, so FSS hypothesis doesn't hold) by collapsing the curves corresponding to different lattice sizes L on a single unknown master curve $g(x)$.

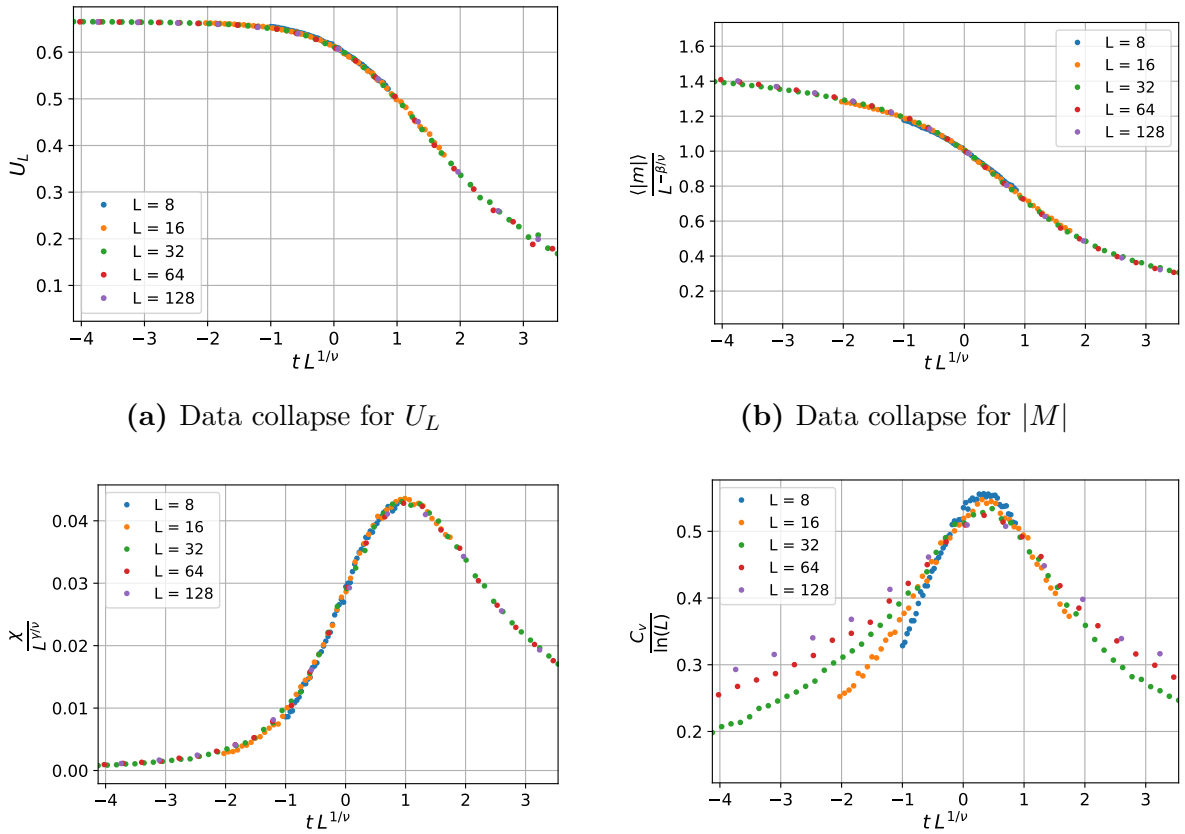
We use the following cost function P_b to as a measure of the collapse, which is essentially a normalized sum of residues, which we get by slightly modifying the cost function given in [1].

$$P_b = \frac{1}{N_{\text{over}}} \sum_p \sum_{j \neq p} \sum_i \frac{\left| L_j^{-\zeta/\nu} Q_{ij} - \varepsilon_p\left(L_j^{1/\nu} t_{ij}\right)\right|}{L_j^{-\zeta/\nu} Q_{ij} + \varepsilon_p\left(L_j^{1/\nu} t_{ij}\right)} \tag{2.6.1}$$

- 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.

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

- $Q = U_L$, minimize $P_b(0, \nu)$ with respect to ν. $\longrightarrow \nu_0$
- $Q = |m|$, 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. $\longrightarrow \gamma_0$



(a) Data collapse for U_L

(b) Data collapse for $|M|$

(c) Data collapse for χ

(d) Data collapse for C_V

Figure 2.3: Data-collapse to extract the critical exponents.

The critical exponents estimated from the data collapse method are

$$\begin{aligned} \nu &= 0.9765 \\ \beta &= 0.1204 \\ \gamma &= 1.7301 \end{aligned}$$

2.6.1 The quality of data collapse (theory)

The quality of the data collapse was defined as a reduced χ^2 statistic by [2], as follows

$$S = \frac{1}{\mathcal{N}} \sum_{i,j} \frac{(y_{ij} - Y_{ij})^2}{dy_{ij}^2 + dY_{ij}^2}, \tag{2.6.2}$$

where the values (y_{ij}, dy_{ij}) are the scaled observations and its standard errors at x_{ij}, and the values (Y_{ij}, dY_{ij}) are the estimated value of the master curve and its standard error at x_{ij}.

The quality S is the mean square of the weighted deviations from the master curve.

As we expect the individual deviations $y_{ij} - Y_{ij}$ to be of the order of the individual error $\sqrt{dy_{ij}^2 + dY_{ij}^2}$ for an optimal fit, the quality S should attain its minimum $S_{\min}$ at around 1 and be much larger otherwise.

Let i enumerate the system sizes $L_i, i = 1, \ldots, k$ and let j enumerate the parameters $t_j, j = 1, \ldots, n$ with $t_1 < t_2 < \ldots < t_n$. The scaled data are

$$\begin{aligned} y_{ij} &:= L_i^{-\zeta/\nu} Q_{L_i,t_j} \\ dy_{ij} &:= L_i^{-\zeta/\nu} dQ_{L_i,t_j} \\ x_{ij} &:= L_i^{1/\nu} t_j \end{aligned}$$

The sum in the quality function S only involves terms for which the estimated value Y_{ij} of the master curve at x_{ij} is defined. The number of such terms is $\mathcal{N}$.

The master curve itself depends on the scaled data. For a given i, L_i, we estimate the master curve at x_{ij} by the two respective data from all the other system sizes which respectively enclose x_{ij} : for each $i \neq i'$, let j' be such that $x_{i'j'} \leq x_{ij} \leq x_{i'(j'+1)}$, and select the points $(x_{i'j'}, y_{i'j'}, dy_{i'j'})$, $\left(x_{i'(j'+1)}, y_{i'(j'+1)}, dy_{i'(j'+1)}\right)$. Do not select points for some i', if there is no such j'. If there is no such j' for all i', the master curve remains undefined at x_{ij}.

Given the selected points (x_l, y_l, dy_l), the local approximation of the master curve is the linear fit

$$y = mx + b$$

with weighted least squares. The weights w_l are the reciprocal variances, $w_l := 1/dy_{ij}^2$. The estimates and (co)variances of the slope m and intercept b are

$$\begin{aligned} \hat{b} &= \frac{1}{\Delta}\left(K_{xx}K_y - K_x K_{xy}\right) \\ \hat{m} &= \frac{1}{\Delta}\left(K K_{xy} - K_x K_y\right) \\ \hat{\sigma}_b^2 = \frac{K_{xx}}{\Delta}, \hat{\sigma}_m^2 &= \frac{K}{\Delta}, \hat{\sigma}_{bm} = -\frac{K_x}{\Delta} \end{aligned}$$

with $K_{nm} := \sum w_l x_l^n y_l^m, K := K_{00}, K_x := K_{10}, K_y := K_{01}, K_{xx} := K_{20}$, $K_{xy} := K_{11}, \Delta := KK_{xx} - K_x^2$ Hence, the estimated value of the master curve at x_{ij} is

$$Y_{ij} = \hat{m} x_{ij} + \hat{b}$$

with error propagation

$$dY_{ij}^2 = \hat{\sigma}^2 x_{ij}^2 + 2\hat{\sigma}_{bm} x_{ij} + \hat{\sigma}_b^2.$$

2.6.2 The quality of data collapse (results)

Using the algorithm in the previous subsection, the quality of the different data collapse fits are as follows

Quality of U_L vs $tL^{1/\nu}$ collapse = 1.6933 (Fig. 2.3a)

Quality of $\langle|m|\rangle L^{\beta/\nu}$ vs $tL^{1/\nu}$ collapse = 22.4764 (Fig. 2.3b)

Quality of $\chi L^{-\gamma/\nu}$ vs $tL^{1/\nu}$ collapse = 0.7372 (Fig. 2.3c)

Quality of $C_V/\ln(L)$ vs $tL^{1/\nu}$ collapse = 6.9356 (Fig. 2.3d)

2.7 Problems that I'm facing

Following are some problems which I've not yet been able to resolve

1. **Interpreting the quality of data collapse as error** As discussed in our last meeting, you mentioned that the reduced χ^2 quality function S (Eq. (2.6.2)) itself is a measure of an error after looking at [2], so we can somehow normalize it to get the errors in physical quantities. However, even after reading the paper carefully, I still haven't understood how the authors calculated the errors, and to my knowledge, they haven't explicitly written down how they have calculated it.

Chapter 3

Singlet constraint

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

$$H = -\sum_p J_p B_p - \sum_l h_e X_e \tag{3.0.1}$$

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

$$H_{\text{dual}} = -\sum_p J_p \sigma_p^x - \sum_e h_e \sigma_{p_e}^z \sigma_{p'_e}^z \tag{3.0.2}$$

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**.

3.1 Quantum-to-Classical correspondance

To perform Monte Carlo on the singlet-Ising model,

$$H = -\sum_p J_p \sigma_p^x - \sum_e h_e \sigma_{p_e}^z \sigma_{p'_e}^z \ , \qquad \prod_p \sigma_p^x = \mathbb{I} \tag{3.1.1}$$

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

$$S_s[\{\sigma\}] = -\sum_l \sum_e (\Delta\tau) h_e \sigma_{p_e}(l)\sigma_{p'_e}(l) - \sum_l \ln\cosh\left[-\sum_p K_p \sigma_p(l+1)\sigma_p(l)\right] \tag{3.1.2}$$

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 $l \in \{1, \dots N_\tau\}$ 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

$$S[\{\sigma\}] = -\sum_{l=1}^{N_\tau} \sum_{i=1}^{N_x} (\Delta\tau) h_i \sigma_i(l)\sigma_{i+1}(l) - \sum_{l=1}^{N_\tau} \ln\cosh\left[-\sum_{i=1}^{N_x} K_i \sigma_i(l+1)\sigma_i(l)\right] \tag{3.1.3}$$

3.2 Alignment observable

Carefully analyzing the term $\sigma_i(l)\sigma_i(l+1)$ in Eqn. (3.1.3), 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** $A(l)$

$$A(l) \equiv \frac{1}{N_x} \sum_{i=1}^{N_x} \sigma_i(l)\sigma_i(l+1) = \frac{1}{N_x}(N_a(l) - N_o(l)) \quad \in [-1, 1] \tag{3.2.1}$$

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

$$S = -h\Delta\tau \sum_{i=1}^{N_x} \sum_{l=1}^{N_\tau} \sigma_i(l)\sigma_i(l+1) - \sum_{l=1}^{N_\tau} \ln\cosh\left[-KN_x A(l)\right]. \tag{3.2.2}$$

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. (3.2.2).

3.3 Subsystem symmetry breaking

The effective action in Eqn. (3.2.2) 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).

3.4 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 ε'_{l_0}, ε'_{l_0-1};
    define ΔS, ΔS_x, ΔS_t;

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

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

    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

3.5 Metropolis simulation results

3.5.1 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.

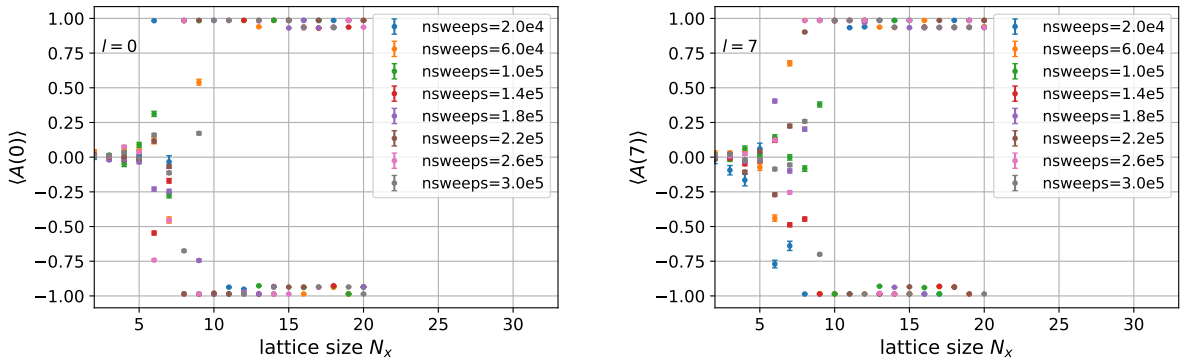
For the simulations, we use the following parameters

`spatial lattice size` $N_x \in \{2, 3, 4 \ldots, 20\}$

`no of sampling sweeps` $N_{\text{samp}} \in \{2.0e4, 6.0e4, 1.0e5, \ldots, 2.6e5, 3.0e5\}$

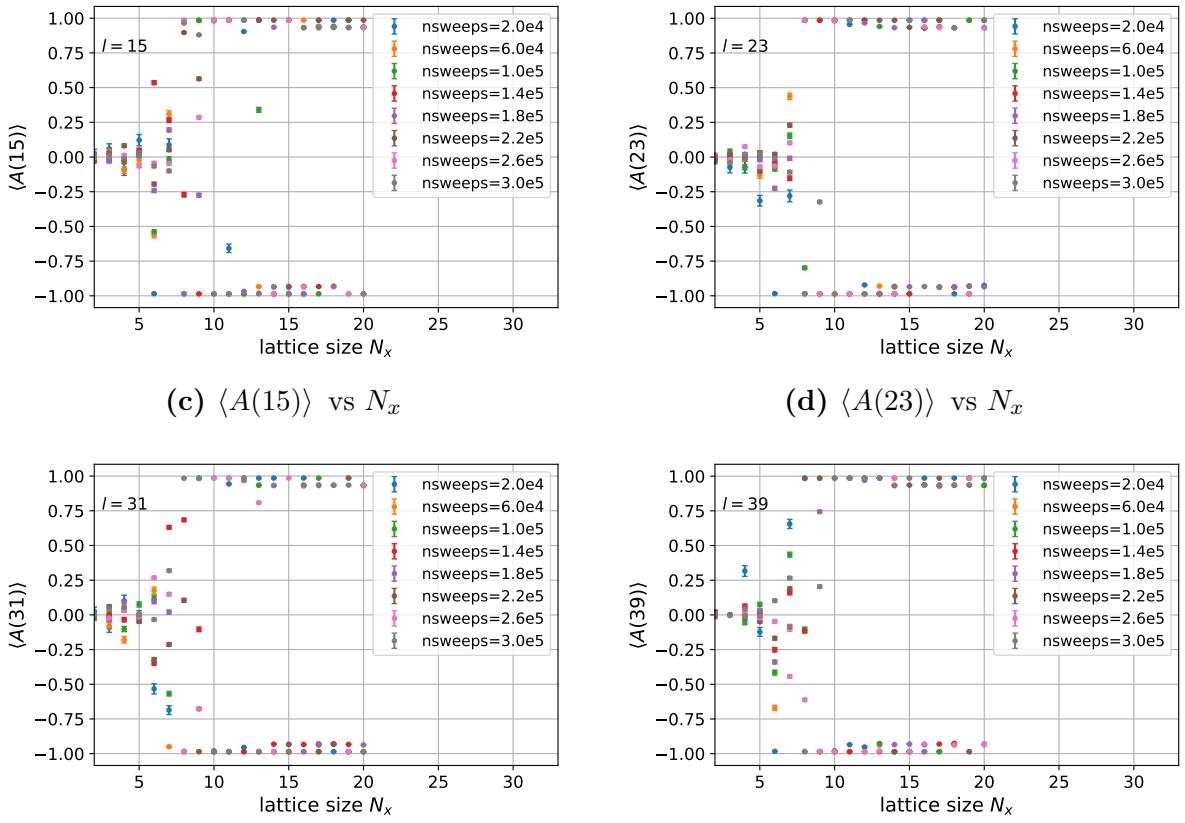
`imaginary time lattice size` $N_\tau = 40, \quad \Delta\tau = 1.0$

`coupling constants` $K, h = 1.0$



(a) $\langle A(0) \rangle$ vs N_x **(b)** $\langle A(7) \rangle$ vs N_x

Figure 3.1: $\langle A(l) \rangle$ 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 3.1: $\langle A(l)\rangle$ for different values of l calculated for different number of Monte Carlo sweeps.

As can be seen from Fig. 3.1, 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. 3.2 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. 3.1 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`.

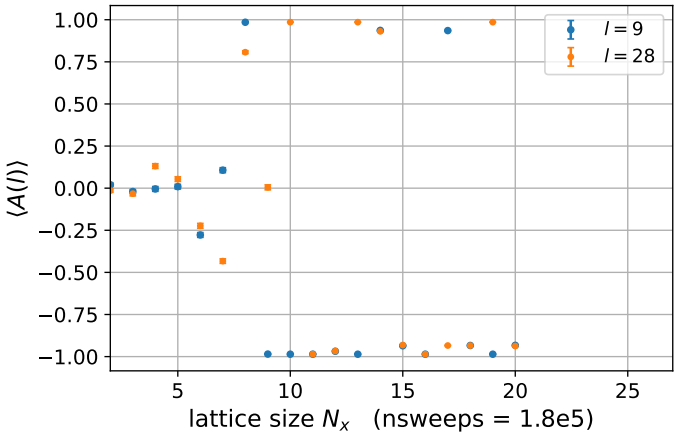


Figure 3.2: $\langle A(l)\rangle$ vs N_x for two different layers $l_1 = 9$, $l_2 = 28$ ran over `nsweeps = 1.8e5`.

3.5.2 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

$$\text{Autocorr}[A_l] = \frac{\langle A_l(k)A_l(k+T)\rangle}{\langle A_l(k)^2\rangle} = \frac{1}{N-T}\sum_{k=0}^{N-T-1}\frac{A_l(k)\cdot A_l(k+T)}{\langle A_l(k)^2\rangle} \tag{3.5.1}$$

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

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

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` $\}$.

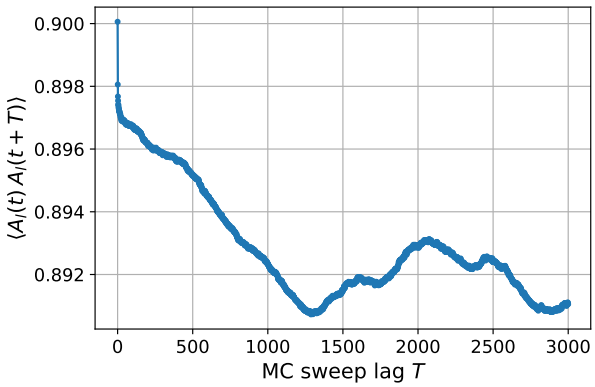


Figure 3.3: Autocorrelation function $\langle A_l(k)A_l(k+T)\rangle$.

As can be seen in Fig. 3.3, 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.

3.6 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.

3.6.1 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` $N_{\texttt{samp}} =$ `5.0e4`

`imaginary time lattice size` $N_\tau =$ `40`, $\Delta\tau =$ `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 $A(l)$.

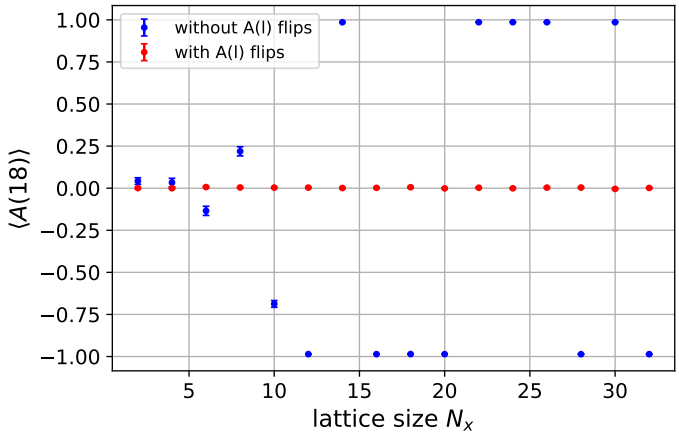
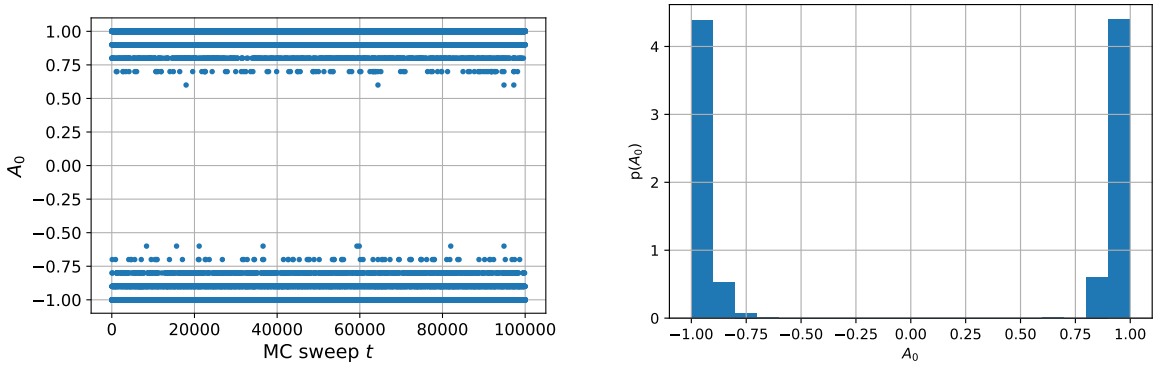


Figure 3.4: $\langle A(l = 18)\rangle$ both with and without alignment flips for different spatial lattice sizes.

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. 3.4. 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 3.5: Configuration space exploring both $A(l) > 0$ and $A(l) < 0$ sectors **with** alignment flips.

Fig. 3.5 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.

3.6.2 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

$$\text{Autocorr}[A_l] = \frac{1}{N-T} \sum_{k=0}^{N-T-1} \frac{(A_l(k) - \langle A_l \rangle) \cdot (A_l(k+T) - \langle A_l \rangle}{\langle A_l(k)^2 \rangle} \tag{3.6.1}$$

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

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

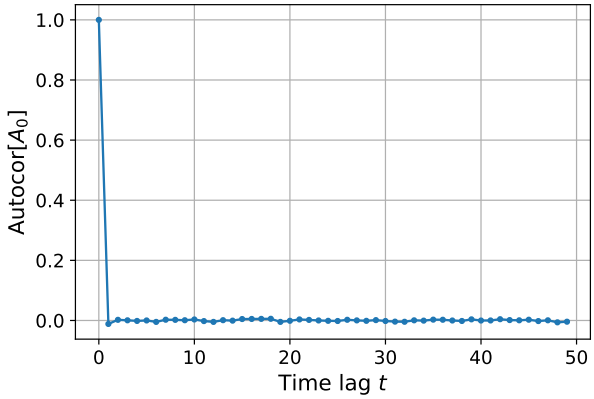


Figure 3.6: Autocorrelation function **with** alignment flips.

As is apparent from Fig. 3.6, 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.

3.7 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 $\hat{H}_{\text{TFIM}}$ (3.1.1) 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.

3.7.1 Types of operators in consideration

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

$$\hat{S}\hat{\mathcal{O}}\hat{S} = \hat{\mathcal{O}} \tag{3.7.1}$$

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 (3.7.1).

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

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

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

$$\mathcal{Z}\langle\hat{\mathcal{O}}\rangle = \left(\prod_{l=0}^{N_\tau-1}\sum_{\{\sigma_l\},\mu_l,\lambda_l}\right)\frac{1}{N_\tau}\sum_{l_0=0}^{N_\tau-1}\left[\langle\{\mu_{l_0+1}\sigma_{l_0+1}\}|e^{-\Delta\tau\hat{H}}\hat{\mathcal{O}}|\{\lambda_{l_0}\sigma_{l_0}\}\rangle\cdot\prod_{l\neq l_0}\langle\{\mu_{l+1}\sigma_{l+1}\}|e^{-\Delta\tau\hat{H}}|\{\lambda_l\sigma_l\}\rangle\right] \tag{3.7.2}$$

3.7.2 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. (3.7.2)

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

After manipulations on the $\prod_l$ product of terms, and performing a sum $\sum_{\mu_l,\lambda_l\in\{\pm1\}}$, we get the usual partition function with the classical basis states

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

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

$$\boxed{O_{Z_i Z_j} = \frac{1}{N_\tau} \sum_{l=0}^{N_\tau-1} \sigma_l^i \sigma_l^j} \tag{3.7.3}$$

3.7.3 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

$$\mathcal{Z}\langle \hat{X}_i \rangle$$

$$= \left(\prod_{l=0}^{N_\tau-1} \sum_{\{\sigma_l\},\mu_l,\lambda_l} \right) \frac{1}{N_\tau} \sum_{l_0=0}^{N_\tau-1} \left[\langle \{\mu_{l_0+1}\sigma_{l_0+1}\} | e^{-\Delta\tau \hat{H}} \hat{X}_i | \{\lambda_{l_0}\sigma_{l_0}\} \rangle \cdot \prod_{l\neq l_0} \langle \{\mu_{l+1}\sigma_{l+1}\} | e^{-\Delta\tau\hat{H}} | \{\lambda_l\sigma_l\} \rangle \right] \tag{3.7.4}$$

$$\tag{3.7.5}$$

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

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

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

$$\begin{aligned} &\langle \{\mu_{l_0+1}\sigma_{l_0+1}\} | e^{-\Delta\tau\hat{H}_T} \hat{X}_i | \{\lambda_{l_0}\sigma_{l_0}\} \rangle \\ &= \langle \{\mu_{l_0+1}\sigma_{l_0+1}\} | e^{\Delta\tau \sum_j \hat{X}_j} \hat{X}_i | \{\lambda_{l_0}\sigma_{l_0}\} \rangle \\ &= \big\langle \mu_{l_0+1}\sigma_{l_0+1}^i \big| e^{\Delta\tau\hat{X}_i} \hat{X}_i \big| \lambda_{l_0}\sigma_{l_0}^i \big\rangle \prod_{j\neq i} \big\langle \mu_{l_0+1}\sigma_{l_0+1}^j \big| e^{\Delta\tau\hat{X}_j} \big| \lambda_{l_0}\sigma_{l_0}^j \big\rangle \end{aligned} \tag{3.7.6}$$

Now, on simplifying the operator products above,

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

Therefore, when we evaluate the matrix products (3.7.6), we get

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

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. (3.7.6) becomes

$$\begin{aligned}
&\langle\{\mu_{l_0+1}\sigma_{l_0+1}\}|e^{-\Delta\tau\hat{H}_T}\hat{X}_i|\{\lambda_{l_0}\sigma_{l_0}\}\rangle \\
&= A^{N_x}e^{-K(\mu_{l_0+1}\lambda_{l_0})\sigma^i_{l_0+1}\sigma^i_{l_0}}\prod_{j\neq i}e^{+K(\mu_{l_0+1}\lambda_{l_0})\sigma^j_{l_0+1}\sigma^j_{l_0}} \\
&= A^{N_x}\exp\left(-2K(\mu_{l_0+1}\lambda_{l_0})\sigma^i_{l_0+1}\sigma^i_{l_0}\right)\exp\left(K(\mu_{l_0+1}\lambda_{l_0})\sum_{j=0}^{N_x-1}\sigma^j_{l_0+1}\sigma^j_{l_0}\right)
\end{aligned} \tag{3.7.7}$$

For all other layers $l \neq l_0$, the other matrix element (without $\hat{X}_i$) from (3.7.5) is similarly calculated as

$$\langle\{\mu_{l+1}\sigma_{l+1}\}|e^{-\Delta\tau\hat{H}_T}|\{\lambda_l\sigma_l\}\rangle = A^{N_x}\exp\left(K(\mu_{l+1}\lambda_l)\sum_{j=0}^{N_x-1}\sigma^j_{l+1}\sigma^j_l\right) \tag{3.7.8}$$

Combining the matrix elements of both (3.7.7) and (3.7.8), 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 $\in \{\pm 1\}$.

$$\sim \left(\prod_{l=0}^{N_\tau-1} \sum_{\{\sigma_l\}}\right) \frac{1}{N_\tau} \sum_{l_0=0}^{N_\tau-1} \left[\left(\sum_{\mu_{l_0}} e^{K\mu_{l_0+1} \sum_j \sigma^j_{l_0+1}\sigma^j_{l_0} - 2K\mu_{l_0+1}\sigma^i_{l_0}\sigma^i_{l_0+1}} \right) e^{h\Delta\tau \sum_j \sigma^j_{l_0}\sigma^{j+1}_{l_0}} \right.$$
$$\left. \prod_{l\neq l_0} \sum_{\mu_l} e^{K\mu_{l+1} \sum_j \sigma^j_{l+1}\sigma^j_l + h\Delta\tau \sum_j \sigma^j_l \sigma^{j+1}_l} \right]$$
$$\sim \left(\prod_{l=0}^{N_\tau-1} \sum_{\{\sigma_l\}}\right) \frac{1}{N_\tau} \sum_{l_0=0}^{N_\tau-1} \left[\cosh\left(K \sum_j \sigma^j_{l_0+1}\sigma^j_{l_0} - 2K\sigma^i_{l_0}\sigma^i_{l_0+1} \right) e^{h\Delta\tau \sum_j \sigma^j_{l_0}\sigma^{j+1}_{l_0}} \right.$$
$$\left. \prod_{l\neq l_0} \cosh\left(K \sum_j \sigma^j_{l+1}\sigma^j_l \right) e^{h\Delta\tau \sum_j \sigma^j_l \sigma^{j+1}_l} \right] \tag{3.7.9}$$

Finally, we can expand the $\cosh(x-y)$ term in Eqn. (3.7.9), 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

$$\mathcal{Z}\langle \hat{X}_i \rangle \sim \left(\prod_{l=0}^{N_\tau-1} \sum_{\{\sigma_l\}}\right) \frac{1}{N_\tau} \sum_{l_0=0}^{N_\tau-1} \left(\cosh\left(2K\sigma^i_{l_0}\sigma^i_{l_0+1}\right) - \tanh\left(K \sum_j \sigma^j_{l_0+1}\sigma^j_{l_0} \right) \sinh\left(2K\sigma^i_{l_0}\sigma^i_{l_0+1}\right) \right) e^{-S_{\text{eff}}} \tag{3.7.10}$$

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

$$\boxed{O_{X_i} = \frac{1}{N_\tau} \sum_{l=0}^{N_\tau-1} \left[\cosh\left(2K\sigma^i_l\sigma^i_{l+1}\right) - \tanh\left(K \sum_{j=0}^{N_x-1} \sigma^j_{l+1}\sigma^j_l \right) \sinh\left(2K\sigma^i_l\sigma^i_{l+1}\right) \right]} \tag{3.7.11}$$

One can also write it a little compactly as

$$\boxed{O_{X_i} = \frac{1}{N_\tau} \sum_{l=0}^{N_\tau-1} \left[\frac{\cosh\left(2K\sigma^i_l\sigma^i_{l+1} - K\sum_{j=0}^{N_x-1} \sigma^j_{l+1}\sigma^j_l\right)}{\cosh\left(K\sum_{j=0}^{N_x-1} \sigma^j_{l+1}\sigma^j_l\right)} \right]} \tag{3.7.12}$$

3.8 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.

3.8.1 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

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

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

$$
\begin{aligned}
\langle\hat{X}\rangle &= 2uv\left(\frac{e^{2\beta B} - e^{-2\beta B}}{e^{2\beta B} + e^{-2\beta B}}\right) = 2uv\tanh(2\beta B) \\
&\implies \boxed{\langle\hat{X}\rangle = \frac{J}{B}\tanh(2\beta B)} \qquad (3.8.1)
\end{aligned}
$$

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

$$\begin{aligned}
\langle \hat{Z}_0\hat{Z}_1\rangle &= (u^2 - v^2)\left(\frac{e^{2\beta B} - e^{-2\beta B}}{e^{2\beta B} + e^{-2\beta B}}\right) = (u^2 - v^2)\tanh(2\beta B) \\
\implies & \boxed{\langle \hat{Z}_0\hat{Z}_1\rangle = \frac{h}{B}\tanh(2\beta B)}
\end{aligned} \tag{3.8.2}$$

3.8.2 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.

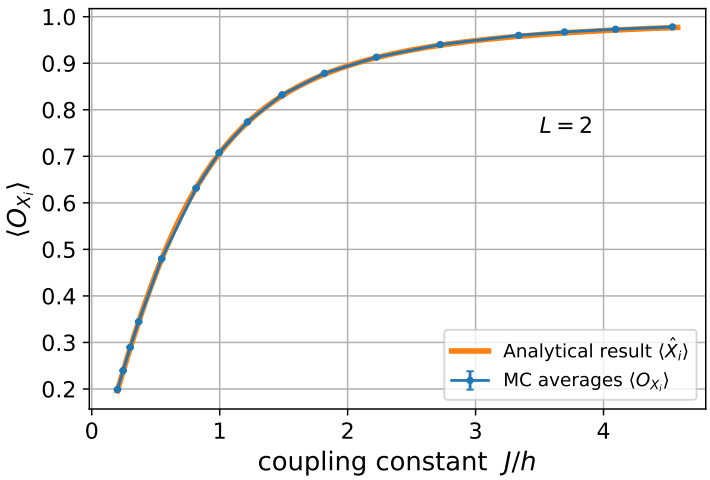
$$O_{Z_0Z_1} = \frac{1}{N_\tau}\sum_{l=0}^{N_\tau-1}\sigma_l^0\sigma_l^1$$

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

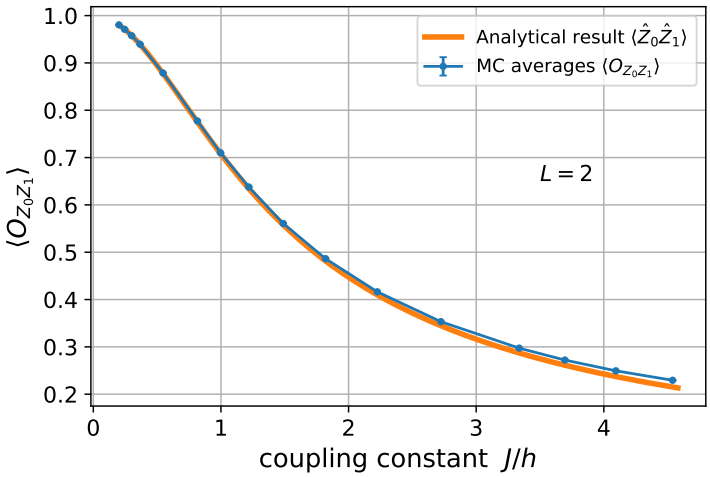
Therefore, we can now use Metropolis Monte Carlo with an acceptance probability of $\min(1, e^{-\Delta S})$ for spin flips, combined with alignment flips. Finally, we compare the MC expectation values of O_X and $O_{Z_0Z_1}$ with the analytical results for $\langle\hat{X}\rangle$ and $\langle\hat{Z}_0\hat{Z}_1\rangle$.

We perform the Monte Carlo simulation for $\beta = N_\tau = 50$, keeping $\Delta\tau = 1$ and alignment flip fraction $f = 1.0$. 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 $h = 0.05$ 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 3.7: 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. 3.7, 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.

3.9 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

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

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.

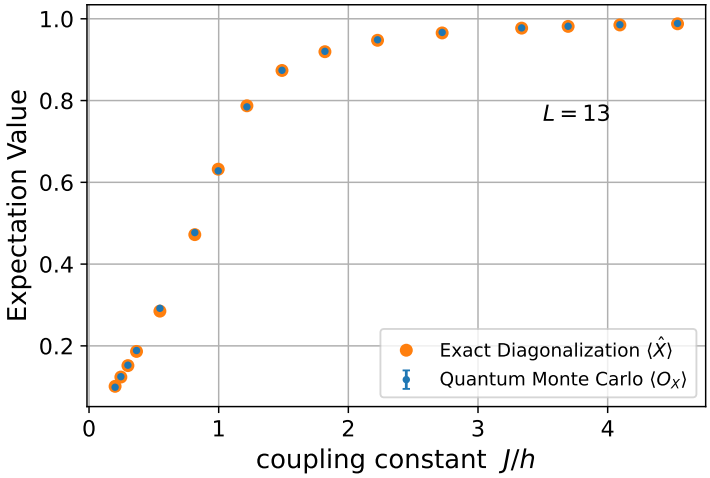
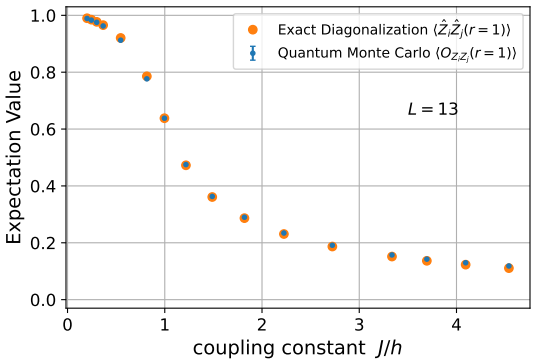
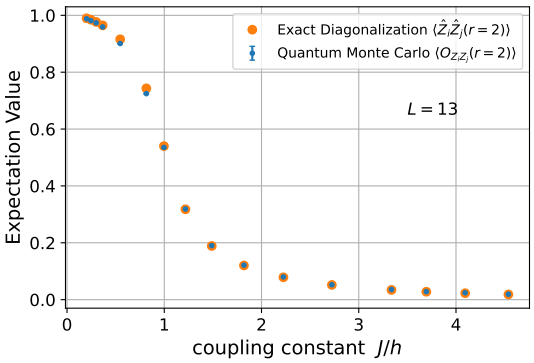


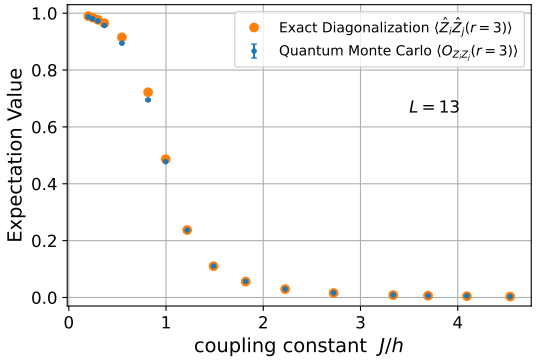
Figure 3.8: Comparing ED results for $\langle \hat{X} \rangle$ with MC average $\langle O_X \rangle$ vs. J/h.



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



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



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

Figure 3.9: 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.

3.10 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,

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

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.

[2] Jé rôme Houdayer and Alexander Hartmann. "Low-temperature behavior of two-dimensional Gaussian Ising spin glasses". In: *Physical Review B* 70.1 (July 2004). DOI: 10.1103/physrevb.70.014418. URL: https://doi.org/10.1103%2Fphysrevb.70.014418.

[3] Werner Krauth. *Cluster Monte Carlo algorithms*. 2003. DOI: 10.48550/ARXIV.COND-MAT/0311623. URL: https://arxiv.org/abs/cond-mat/0311623.

[4] Anders W. Sandvik, Adolfo Avella, and Ferdinando Mancini. "Computational Studies of Quantum Spin Systems". In: *AIP Conference Proceedings*. AIP, 2010. DOI: 10.1063/1.3518900. URL: https://doi.org/10.1063%2F1.3518900.

[5] A. Sokal. "Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms". In: *Functional Integration: Basics and Applications*. Ed. by Cecile DeWitt-Morette, Pierre Cartier, and Antoine Folacci. Boston, MA: Springer US, 1997, pp. 131–192. ISBN: 978-1-4899-0319-8. DOI: 10.1007/978-1-4899-0319-8_6. URL: https://doi.org/10.1007/978-1-4899-0319-8_6.