

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;

$$\langle Q \rangle = \int_0^L dx \rho(x) Q(x), \quad \int_0^L dx \rho(x) = 1, \quad (1.1.1)$$

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

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

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 $\{x_1, x_2, x_3, \dots, x_N\}$, then the integral in Eq. (1.1.1) can be estimated as

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

The error in a Monte Carlo estimate goes as $1/\sqrt{N}$ in any number of dimensions, and hence it is more efficient. 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 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

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

where $\sum {}^{(W)}$ denotes points being sampled from the distribution $W(x)$, and we write $L \sum W(x_i) \rightarrow \sum {}^{(W)}$. 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 $\rho(x)$

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

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 $\rho(\vec{x}, \vec{p}) = e^{-\beta H(\vec{x}, \vec{p})}$, 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 $C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow \dots \rightarrow C_N$ that evolves stochastically and satisfies the Markovian property, i.e., the probability of $C_{i-1} \rightarrow C_i$ transition is independent of its history. Put together, this implies the probability of obtaining the above sequence is

$$P(C_1 \rightarrow C_2 \rightarrow \dots \rightarrow C_N) = P(C_1) \cdot P(C_2|C_1) \cdot P(C_3|C_2) \dots P(C_N|C_{N-1}) \quad (1.1.6)$$

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 $\{X\} = \{X_1, X_2, \dots, X_n\}$ in the configuration space. Assume we start with some configuration $X_{i(0)}$ and stochastically generate a Markov chain $X_{i(1)}, X_{i(2)}, X_{i(3)}, \dots, X_{i(M)}$. 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

$$N_1(X_i) = N_0(X_i) + \sum_{j \neq i} [N_0(X_j) P(X_j \rightarrow X_i) - N_0(X_i) P(X_i \rightarrow X_j)] \quad (1.1.7)$$

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 $i = 1, 2, \dots, M$,

$$\sum_{j \neq i} [\rho(X_j) P(X_j \rightarrow X_i) - \rho(X_i) P(X_i \rightarrow X_j)] \stackrel{!}{=} 0 \quad (1.1.8)$$

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

$$\rho(X_j) P(X_j \rightarrow X_i) - \rho(X_i) P(X_i \rightarrow X_j) \stackrel{!}{=} 0 \quad (1.1.9)$$

which can be written as a ratio

$$\frac{P(X_j \rightarrow X_i)}{P(X_i \rightarrow X_j)} = \frac{\rho(X_i)}{\rho(X_j)}, \quad (1.1.10)$$

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 $P(X_i \rightarrow X_j)$ 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

$$\frac{P_{\text{accept}}(X_j \rightarrow X_i)}{P_{\text{accept}}(X_i \rightarrow X_j)} = \frac{\rho(X_i)}{\rho(X_j)} \quad (1.1.11)$$

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

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

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

$$P_{\text{accept}}(E_i \rightarrow E_j) = \min [e^{-\beta(E_j - E_i)}, 1]. \quad (1.1.13)$$

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

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

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

$$E = -J \sum_i \sigma_{i_x, i_y} [\sigma_{i_x, i_y+1} + \sigma_{i_x+1, i_y}] \quad (1.2.2)$$

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

$$\Delta E = 2J \sigma_{i_x, i_y} [\sigma_{i_x, i_y+1} + \sigma_{i_x+1, i_y} + \sigma_{i_x, i_y-1} + \sigma_{i_x-1, i_y}] \quad (1.2.3)$$

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

$$P_{\text{accept}}(\sigma_i \rightarrow -\sigma_i) = \min [e^{-\beta \Delta E}, 1] \quad (1.2.4)$$

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. In algorithmic language, a Monte Carlo sweep is defined as follows:

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

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

where we have suppressed the superscript (ρ) on the sum denoting sampling over the distribution $\rho(\{\sigma\}) = e^{-\beta E(\{\sigma\})}$.

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