# Monte Carlo Simulation of the 1D Ising Model and Thermal Behavior Analysis

Jimin BYUN

## 1. Introduction

The Ising model is used for the phase transition, particularly magnetic ordering in materials in statistical mechanics. This fundamental framework model, even though, discovers intrinsic features of real-world physical systems as collective behavior and temperature dependent dynamics. This project focus on the one-dimensional Ising model as a testbed for comprehending computationally. Analytical solutions become impractical due to upsurging the number of spins as exponential growth of the state space. Therefore, for sampling system configurations efficiently based on thermal probabilities, Metropolis Monte Carlo algorithm is used for estimating macroscopic quantities such as energy and magnetization. The primary objective of the project is to incarnate a C++ simulation of 1D Ising model, using an object-oriented approach. The aim is that to analyze average energy and magnetization for exploring system's thermodynamic behavior through repeatedly evolving spin configurations at various inverse temperature (β).

This report outlines the structure and design of the simulation code roughly, presents a series of test cases to validate functionality, and explains the simulation methodology minutely. Then analyze the results against a various β values and conclude with discussion on limitations and possible improvements.

## 2. Code Structure & Design

The project code consists of three main files: 'SpinSystem.h', 'SpinSystem.cpp', 'main.cpp'. This modular structure follows an object-oriented approach, separating the core function of the Ising model from the control flow of the simulation. The 'SpinSystem' class encapsulates all data and functions about the 1D Ising model. It maintains integer spins ( +1 or -1) and the private vector of the system size and provides the method of initialization, spin update and measurement.

| Function | Description |
|---|---|
| initializeRandom() | Randomly sets each spin to +1 or -1 |
| attemptSpinFlip(beta) | Performs a Metropolis step to possibly flip a random spin |
| calculateTotalEnergy() | Computes total energy of the current spin configuration |
| calculateMagnetization() | Calculates the net magnetization of the system |
| printSpins() | For debugging |

The 'main.cpp' is the simulation controller. This file sets up the system, implements the Monte Carlo steps, accomplishes loops over several configurations, and computes average quantities such as energy and magnetization. The code also supports the experiment that analyze thermal behavior across the range of β values. The development process was version-controlled by using Git, adding incrementally and tracking via meaningful commits. A 'gitignore' file was used to exclude build artifacts and output files.

## 3. Testing the Code

To ensure that the implementation operates as expected, tested the simulation on small, analytically tractable systems. The code computed expected energy and magnetization values manually, using minimal spins (e.g. 3 or 5) and compared it to the result of the simulation. For instance, consider a 3-spin system initialized to all spins aligned (+1, +1, +1). In this configuration, each spin pair contributes -1 to the energy. There are two such pairs

with a total energy of -2. The magnetization is simply the sum of spins: +3. With the identical system initialized randomly (e.g. +1, -1, +1), the expected energy and magnetization were computed manually and matched with the simulation output. This demonstrated that the function that computes the energy and magnetization operates correctly on known configurations. These cases for tests ensure the accuracy of the simulation logic in the energy and magnetization computations. It also raise confidence in the behavior of the Monte Carlo routine when applied to the larger systems.

For a system of 3 spins with configuration [+1, -1, +1], the calculated energy was 0, and the magnetization was +1, which matches the expected result based on manual calculation.

4. Simulation Procedure

The simulation begins by initializing a 1D array of spins, randomly set to either +1 or -1 respectively. Then, the spin configuration is updated repeatedly by applying the Metropolis Monte Carlo algorithm. At each step, a random spin is selected, and its flip is probabilistically accepted based on the energy change $\Delta E$ and inverse temperature $\beta$. For each configuration, a fixed number of Metropolis step is performed (e.g. 1000 steps), during the updates, the system evolves toward thermal equilibrium gradually. After the updates, the total energy and magnetization of system are recorded. To obtain credible average measurements, the simulation is repeated over multiple independent configurations (e.g. 100). Each implement uses a freshly initialized spin array, and the final results are averaged over all configurations. To investigate the thermal behavior of the system, reiterate the entire process across a range of $\beta$ values (e.g. 0.1, 0.5, 1.0, 1.5, 2.0, 3.0). Through this, we can observe how macroscopic quantities such as energy and magnetization react to changes in temperature.

5. Results & Analysis

To examine how the system reacts to temperature changes, 1D Ising model is simulated across a range of $\beta$ values from 0.1 to 3.0. For each $\beta$, across 100 independent configurations, the average energy and magnetization are computed. As shown in figure 1, the average energy of the system decreases with increasing $\beta$ (i.e. decreasing temperature). This is an expected result as the system tends to settle into lower energy states when thermal fluctuations are reduced. At the high temperature(low $\beta$), the spins are chaotic and maintain the energy relatively high, whereas at the low temperature(high $\beta$), the spins align more frequently, minimizing the interaction energy.

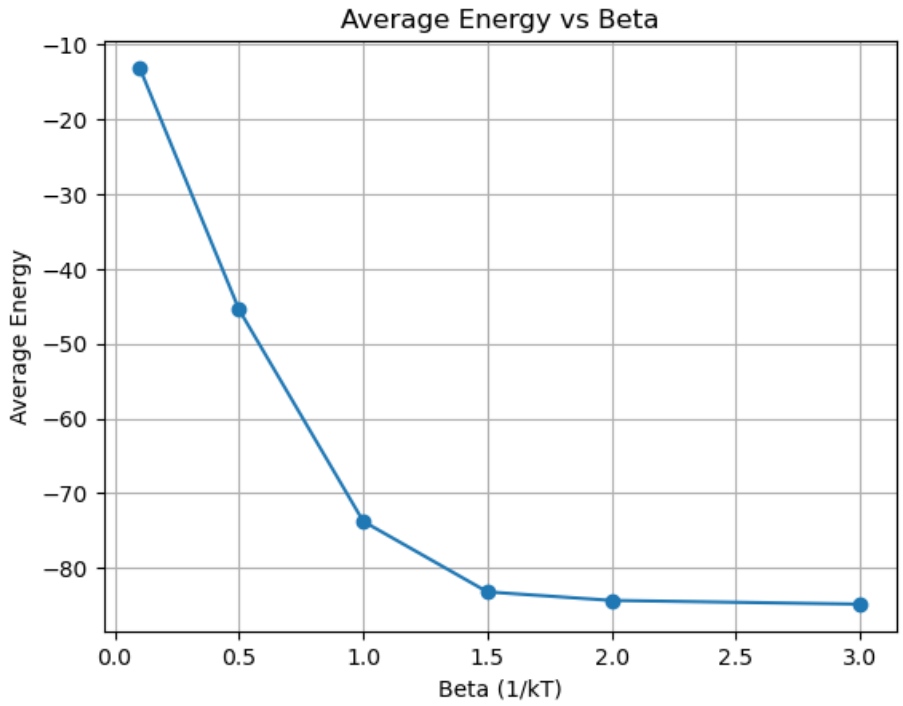


**Figure 1: Average Energy vs. β**

Figure 2 shows the behavior of the average magnetization as a function of β. For small β, the magnetization remains close to zero, indicating a disordered phase. As β increases, magnetization begins to increase, reflecting the increasing of alignment of spins. Although the true spontaneous magnetization does not occur in a 1D Ising model with finite size, the simulation captures transition-lookalike trend due to finite-size effects.

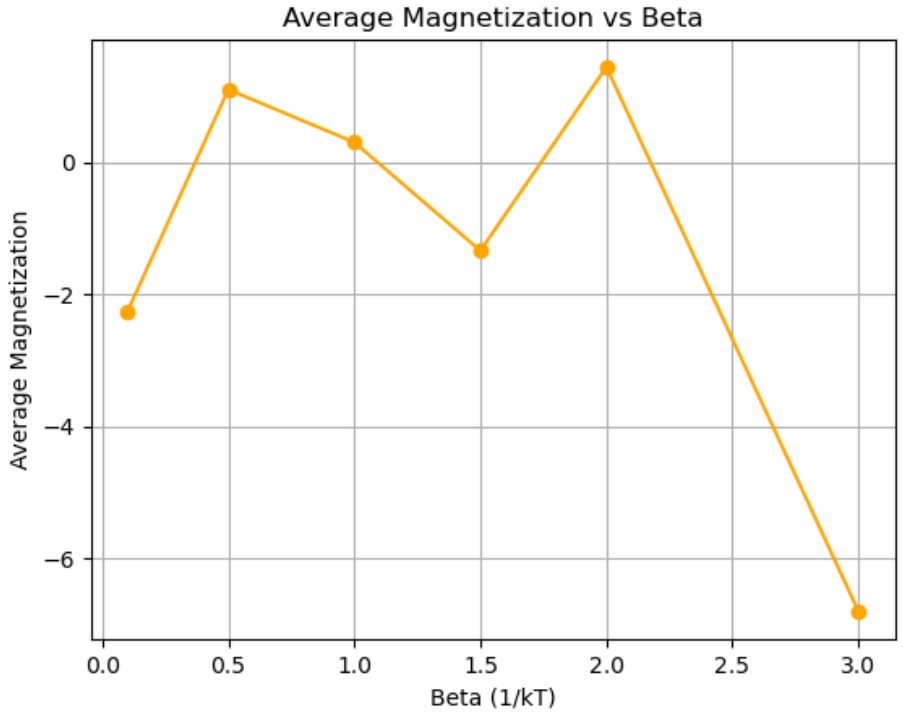


**Figure 2: Average Magnetization vs. β**

Overall, the simulation results are consistent with theoretical expectations. The effectiveness of Metropolis algorithm is validated in sampling the equilibrium behavior by smooth increase in observed magnetization and decrease in energy.

## 6. Limitations & Improvements

One of the main limitations is that the project is using one-dimensional Ising model. Unlike its two-dimensional Ising model, the 1D Ising model does not indicate a true phase transition at finite temperature. Consequently, the behavior like the sharply transition that is observed in magnetization is an artifact of the system's finite size.

The simulations, additionally, were restricted to relatively small system sizes (N = 100) and finite number of Metropolis steps. Bigger systems and longer equilibration times would estimate thermodynamic quantities more accurately, however, it gives increased computational time.

To improve the performance of computational perspectives, reusing random number generators or parallelizing independent configurations could optimize the model. Furthermore, incorporating of periodic boundaries would make the present implementation that uses fixed boundary conditions more closely to physical systems.

If the model is extended to two dimensions, the true phase transition can be observed. Additional features such as data logging, adaptive β scheduling could also raise the educational and analytical value of the simulation.

## 7. Conclusion

In the project, using an object-oriented approach, I implemented C++ simulation of one-dimensional Ising model. The simulation updated the spin configurations by utilizing the Metropolis Monte Carlo algorithm and compute key physical quantities such as energy and magnetization.

Through organized experiments across miscellaneous values of inverse temperature β, expected thermal behavior was observed: decreasing energy and increasing magnetization at lower temperatures. These results, even though it is restricted by features of 1D nature of the model, demonstrated the ability of the simulation that captures collective behavior in spin systems.

The project provided valuable insight into both statistical mechanics and computational modeling. It reinforced the importance of equilibration, averaging and the size of system in numerical experiments.

Overall, the project achieved its objectives successfully, the process in the project deepened my understandings of Monte Carlo method, thermodynamic behavior, the benefits of object-oriented program design in scientific computing.

## 8. Appendix

Commit history

- Step 1: Create SpinSystem class and random initialization
- Step 2: Implement Metropolis algorithm for spin flipping
- Step 3: Add energy and magnetization calculation functions
- Step 4: Loop over configurations to compute averages
- Step 5: Sweep over beta values and analyze results
- Step 6: Output results to CSV and generate plots
- Final: Clean .gitignore and generate submission bundle