Simulation of Ising model

Rithik Rai
3rd Semester MSc Physics
Registration Number: 31822026

Presented to,
Dr. Sasidevan V
Assistant Professor,
Department of Physics
Cochin University of Science and Technology
November 25, 2023.

1 Objective

To simulate the 2D ferromagnetic Ising model on a square lattice and investigate:

- Phase transition point
- Average magnetization as a function of temperature
- Magnetic susceptibility as a function of temperature
- Specific heat as a function of temperature

2 Theoretical Background

Understanding the Ising Model

The Ising model is a mathematical representation designed to explore phase transitions in magnetic systems, particularly in ferromagnetic materials. It operates on a lattice, a finite set of points in dimension D. Each lattice site can assume one of two 'spin' states: +1 or -1. The former corresponds to spin-up, and the latter to spin-down. Denoting N lattice sites as S_i, where $i = 1, 2, 3, \ldots, N$, the spins only interact with their nearest neighbors. Utilizing periodic boundary conditions ensures each spin has an equal number of nearby spins, even at the lattice edges.

The Hamiltonian describing the system's energy with a spin configuration is given by:

$$H = -\sum_{<ij>} J_{ij}\sigma_i\sigma_j - B\sum_i \sigma_i$$

where J_{ij} represents the exchange energy between spins σ_i and σ_j, and B is the applied magnetic field. In the absence of an external magnetic field, the Hamiltonian reduces to:

$$H = -\sum_{<ij>} J_{ij}\sigma_i\sigma_j$$

For ferromagnetic materials, J takes a positive value.

3 Phase Transitions in Ferromagnetic Systems

The Ising model sheds light on phase transitions in ferromagnetic systems. At low temperatures, thermal energy diminishes, making spin interaction energy more dominant. Spins tend to align parallelly, forming a ferromagnetic phase to minimize energy. Conversely, at high temperatures, thermal fluctuations disrupt spin alignment, resulting in a disordered paramagnetic phase with randomly oriented spins. The transition between these states is the Curie point or critical point, marked by the Curie temperature (T_c). Below T_c, magnetization (M) exhibits a non-zero value, dropping abruptly to zero at T_c. Quantities like magnetic susceptibility and specific heat describe this phase change near T_c.

4 Observable Quantities from Simulation

The simulation aims to extract observables: average energy (E), average magnetization (M), specific heat (C_v), and magnetic susceptibility (χ), given by:

$$\langle E \rangle = \frac{1}{2}\left\langle \sum_{i,j} H_{ij} \right\rangle$$

$$\langle M \rangle = \frac{1}{N^2}\sum_{i,j}\sigma_{i,j}$$

$$C_v = \frac{\beta}{T}\left(\langle E^2 \rangle - \langle E \rangle^2\right)$$

$$\chi = \beta\left(\langle M^2 \rangle - \langle M \rangle^2\right)$$

where $\beta = \frac{1}{k_b T}$, with k_b as the Boltzmann constant. A sudden drop of magnetization to zero at T_c signifies a phase transition.

5 Simulation Procedure

We employ the Metropolis Algorithm, a standard method for generating sample configurations of the Ising model in thermal equilibrium at a specific temperature.

- A 2D NxN matrix is created, where each lattice site can be +1 or -1.
- For periodic boundary conditions, we use the modulo operator, taking neighboring sites as $((i+1) \mod (N), j)$ instead of $((i+1), j)$.
- A lattice site is randomly selected, and $\triangle E = (E_{\text{final}} - E_{\text{initial}})$ is calculated for a fixed temperature.
- If $\triangle E \leq 0$, the spin flips because the change is favorable (i.e., the final state has lower energy).
- If $\triangle E > 0$, a random number x is generated on the interval [0, 1]. The spin flips only if $e^{-\triangle E/k_b T} > x$ (i.e., the particle has enough energy to flip the spin), otherwise, the spin change is rejected.
- This process is repeated, and the modified state of the system at a particular temperature T, represented by a new matrix, is returned.
- The procedure is repeated at different temperatures, noting the energy and magnetization for each step.

Onsager's solution for the critical temperature for a square lattice is:

$$T_c = \frac{2}{\log(1+\sqrt{2})} = 2.269 J/k_B$$

where J = 1.

6 Results

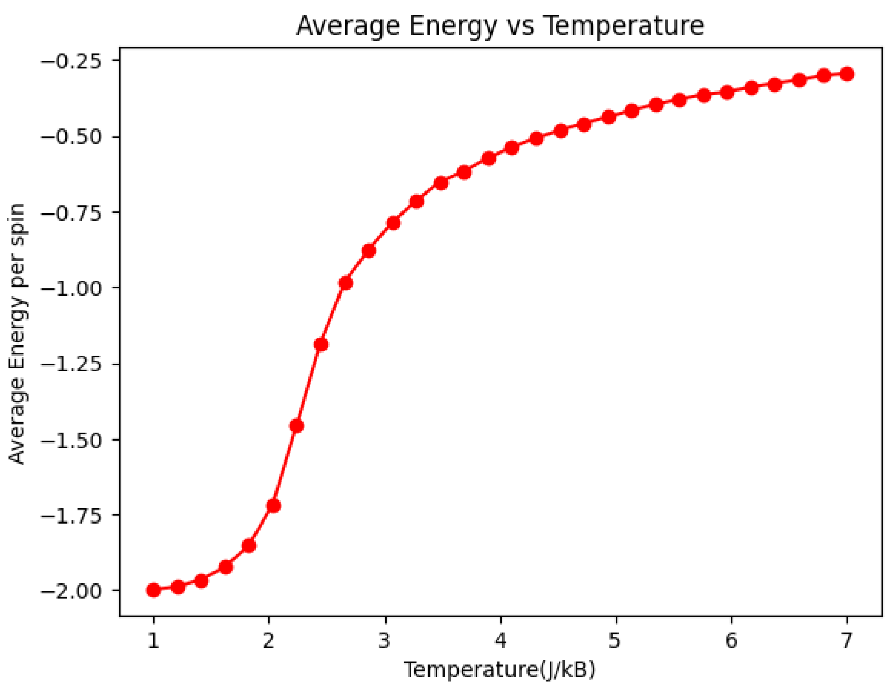

Figure 1: **Average energy as a function of temperature**

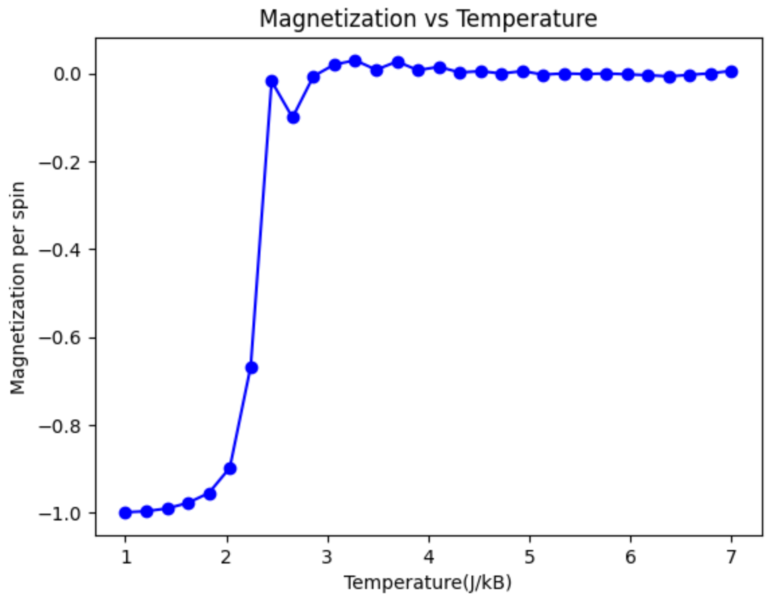

Figure 2: **Average magnetization as a function of temperature**
The magnetization tends to drop to zero between the temperatures $2 < T < 3$.

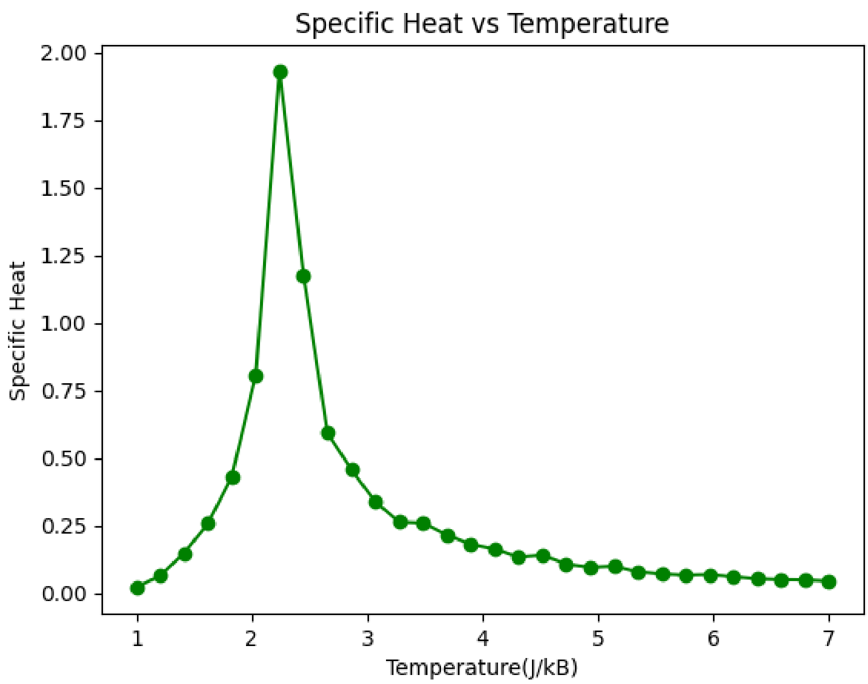

Figure 3: **Specific heat capacity as a function of temperature**
A peak is observed at temperature $2 < T < 3$, and the value of specific heat approaches zero above and below these temperatures.

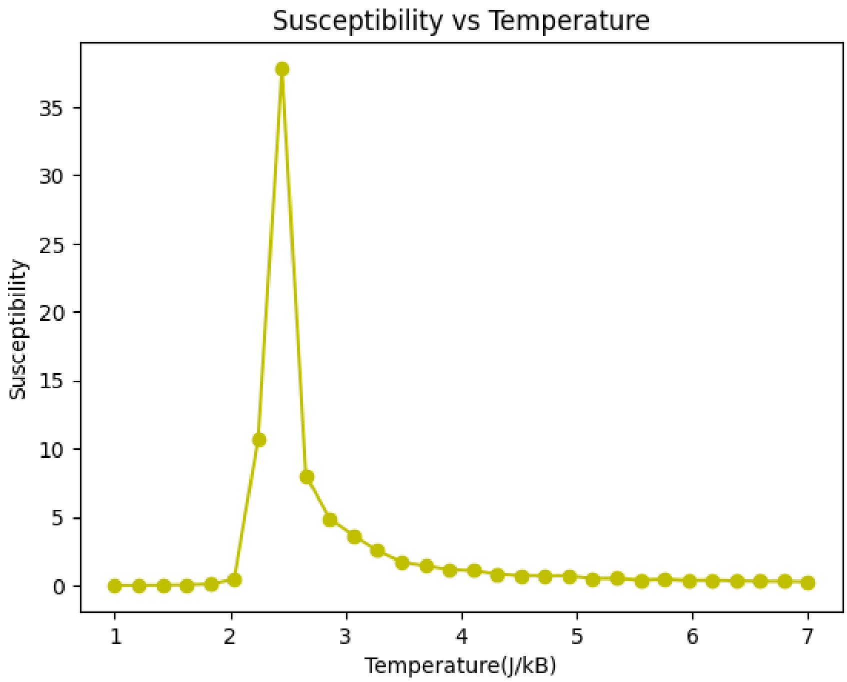

Figure 4: **Magnetic susceptibility as a function of temperature**
A peak is observed at temperature $2 < T < 3$, and the value of susceptibility approaches zero above and below these temperatures.

```python
import numpy as np
import math
import matplotlib.pyplot as plt

def init(N):
    lattice1 = np.random.random(size=(N, N))
    lattice = np.zeros((N,N))
    lattice[lattice1>=0.75] = 1
    lattice[lattice1<0.75] = -1
    return lattice

def energy(lattice, N):
    total_energy = 0
    for i in range(len(lattice)):
        for j in range(len(lattice)):
            S = lattice[i, j]
            nb = lattice[(i + 1) % N, j] + lattice[i, (j + 1) % N] + lattice[(i - 1) % N, j] + lattice[i, (j - 1) % N]
            total_energy += -nb * S
    return total_energy / 2

def magnetization(lattice):
    mag = np.sum(lattice)
    return mag

def mc_step(lattice, temp):
    N = len(lattice)
    beta = 1 / temp
    for i in range(N):
        for j in range(N):
            a = np.random.randint(0, N)
            b = np.random.randint(0, N)
            sigma = lattice[a, b]
            neighbors = lattice[(a + 1) % N, b] + lattice[a, (b + 1) % N] + lattice[(a - 1) % N, b] + lattice[a, (b - 1) % N]
            del_E = 2 * sigma * neighbors
            if del_E < 0:
                sigma *= -1
            elif np.random.rand() < np.exp(-del_E * beta):
                sigma *= -1
            lattice[a, b] = sigma
    return lattice
def cal_E_M_C_X(lattice, N, step, eqstep):
    nt = 30
    T_c = 2 / math.log(1 + math.sqrt(2))
    T = np.linspace(1.0, 7.0, nt)
    energies = []
    magnetizations = []
    specificheats = []
    susceptibilities = []

    for t in range(nt):
        temp = T[t]
        E = 0
        M = 0
        E_sqrd = 0
        M_sqrd = 0

        for i in range(eqstep):
            mc_step(lattice, temp)
        for i in range(step):
            mc_step(lattice, temp)
            e = energy(lattice, N)
            m = magnetization(lattice)
            E += e
            M += m
            E_sqrd += e**2
            M_sqrd += m**2

        E_mean = E / step
        M_mean = M / step
        E_sqrd_mean = E_sqrd / step
        M_sqrd_mean = M_sqrd / step

        Energy = E_mean / (N**2)
        Magnetization = M_mean / (N**2)
        SpecificHeat = (E_sqrd_mean - E_mean**2) / (N**2 * temp**2)
        Susceptibility = (M_sqrd_mean - M_mean**2) / (N**2 * temp)

        energies.append(Energy)
        magnetizations.append(Magnetization)
        specificheats.append(SpecificHeat)
        susceptibilities.append(Susceptibility)


    plt.plot(T, magnetizations, 'bo-')
    plt.xlabel('Temperature')
    plt.ylabel('Magnetization')
    plt.title('Magnetization vs Temperature')
    plt.show()


    plt.plot(T, energies, 'ro-')
    plt.xlabel('Temperature')
    plt.ylabel('Average Energy')
    plt.title('Average Energy vs Temperature')
    plt.show()


    plt.plot(T, specificheats, 'go-')
    plt.xlabel('Temperature')
    plt.ylabel('Specific Heat')
    plt.title('Specific Heat vs Temperature')
    plt.show()


    plt.plot(T, susceptibilities, 'yo-')
    plt.xlabel('Temperature')
    plt.ylabel('Susceptibility')
    plt.title('Susceptibility vs Temperature')
    plt.show()


N = 20
step = 1000
eqstep = 100
lattice = init(N)
cal_E_M_C_X(lattice, N, step, eqstep)
```

Figure 5: **simulation code**