Ising Model and Monte Carlo Simulation

Mina Shiri (40011023)

1 Problem Statement

The magnetic properties of materials involve an N-dimensional array of sites, which, for example, represent molecules. The magnetic moment (spin) of each site is assumed to be only in the up or down direction; no other orientation is allowed. Therefore, each spin in the system can have one of two possible values, which for convenience we take as $s_i = \pm 1$. The total energy of the system is derived from the interaction of spin pairs and is given as follows:

$$E = -J_1 \sum_{<ij>} s_i s_j - J_2 \sum_{<<ij>>} s_i s_j - J_3 \sum_{<<<ij>>>} s_i s_j \tag{1}$$

Where, according to Figure 1, $< ij >$ is the sum over the nearest neighbors, $<< ij >>$ is the sum over the second nearest neighbors, and $<<< ij >>>$ is the sum over the third nearest neighbors of spin i. J_1, J_2, and J_3 are known as the spin interaction constants, which we assume to be positive, and we have:

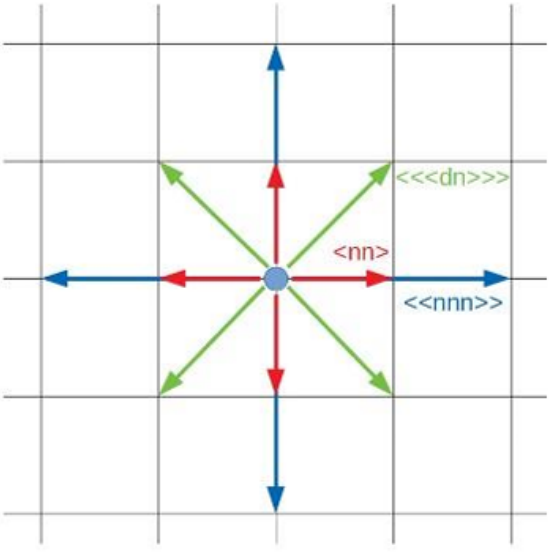


Figure 1: Defined neighbor interactions in the system. Labels: <nn>, «nnn», «<nnn»>

$$\frac{J_1}{k} = 1; \quad \frac{J_2}{k} = 0.5; \quad \frac{J_3}{k} = 0.2 \tag{2}$$

Also, note that the negative sign on the right side makes the state of parallel spins more stable than that of anti-parallel spins.

1.0 Problem Requirements

1. Write a program based on Monte Carlo simulation to solve the 2D Ising model with periodic boundary conditions.
2. Obtain the magnetization and total energy for the system at different temperatures.
3. Investigate time and ensemble averaging in the calculation of magnetization and energy.
4. Determine the role of the system size (number of spins) in the magnetization and energy.
5. By changing the interaction constants, calculate the role of considering the second and third neighbors in these simulations.

In the following sections, we will address the solution to this problem in detail.

2 Analytical Solution

The availability of an analytical solution for the Ising model depends on the problem's conditions:

- The one-dimensional model has an exact analytical solution.
- The two-dimensional nearest-neighbor model has an exact analytical solution, obtained by Lars Onsager in 1944.
- The two-dimensional model with second and third neighbors is analytically intractable due to increased complexity.

Therefore, there is no exact analytical solution for the 2D Ising model with second and third nearest-neighbor interactions. Given that our problem includes these interactions on a 2D lattice, it lacks an analytical solution and requires numerical simulation for its resolution.

3 Numerical Solution

To solve this problem, we write a numerical program. First, we define the problem parameters.

```
import numpy as np
import matplotlib.pyplot as plt
from tqdm import tqdm # import for progress bar

L = 10 # Lattice size
T = 2.5 # Temperature
k = 1.38e-23 # Boltzmann constant
J1, J2, J3 = 1.0, 0.5, 0.2 # Interaction constants
steps = 10 # Number of Monte Carlo steps
```

Listing 1: Defining the initial problem parameters

As you can see, our initial system is 10x10, and the number of time steps is also set to 10. Next, we define our random system.

```
def initialize_lattice(L):
    return np.random.choice([-1, 1], size=(L, L))

lattice = initialize_lattice(L)
```

Listing 2: Function to create a random spin lattice

Each part of the lattice is assigned a spin of up (1) or down (-1). We also plot the resulting values.

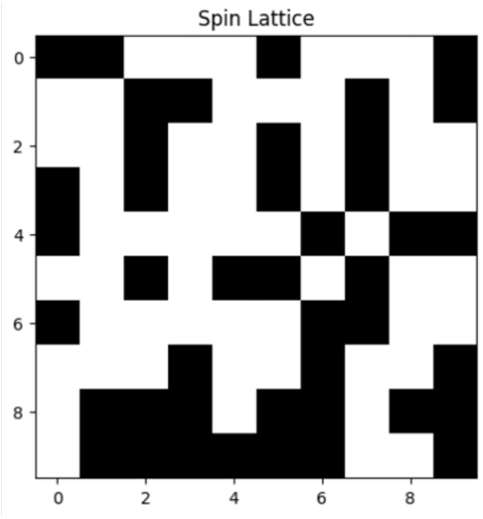


Figure 2: A graphical representation of a random 10x10 spin lattice. Black and white colors represent spins -1 and 1, respectively.

Now, we need to write a function to calculate the energy of the system.

```
def calculate_energy(lattice, J1, J2, J3):
    L = lattice.shape[0]
    energy = 0
    for i in range(L):
        for j in range(L):
            S = lattice[i, j]
            # Nearest neighbors
            nb = lattice[(i+1)%L, j] + lattice[i, (j+1)%L] + \
                 lattice[(i-1)%L, j] + lattice[i, (j-1)%L]
            # Second nearest neighbors
            snb = lattice[(i+1)%L, (j+1)%L] + lattice[(i-1)%L, (j-1)%L] + \
                  lattice[(i+1)%L, (j-1)%L] + lattice[(i-1)%L, (j+1)%L]
            # Third nearest neighbors
            tnb = lattice[(i+2)%L, j] + lattice[i, (j+2)%L] + \
                  lattice[(i-2)%L, j] + lattice[i, (j-2)%L]

            energy += -S * (J1*nb + J2*snb + J3*tnb)
    return energy / 2.0 # Divide by 2 to avoid double counting
```

Listing 3: Function to calculate the total energy of the system

3.0 Monte Carlo Algorithm

Now we write the Monte Carlo algorithm. In this algorithm, we have two conditions:

- If the energy change after a spin flip (ΔE) is negative, the spin always flips.
- If the energy change (ΔE) is positive, the spin flips with the Boltzmann probability ($e^{-\Delta E/kT}$). This is done by comparing the Boltzmann factor with a random number.

```
def monte_carlo_step(lattice, T, J1, J2, J3, k=1):
    L = lattice.shape[0]
    i, j = np.random.randint(0, L), np.random.randint(0, L)
    S = lattice[i, j]

    # Calculate energy change
    nb = lattice[(i+1)%L, j] + lattice[i, (j+1)%L] + \
         lattice[(i-1)%L, j] + lattice[i, (j-1)%L]
    snb = lattice[(i+1)%L, (j+1)%L] + lattice[(i-1)%L, (j-1)%L] + \
          lattice[(i+1)%L, (j-1)%L] + lattice[(i-1)%L, (j+1)%L]
    tnb = lattice[(i+2)%L, j] + lattice[i, (j+2)%L] + \
          lattice[(i-2)%L, j] + lattice[i, (j-2)%L]

    dE = 2 * S * (J1*nb + J2*snb + J3*tnb)

    # Flip condition
    if dE < 0 or np.random.rand() < np.exp(-dE / (k * T)):
        lattice[i, j] = -S

    return lattice
```

Listing 4: Implementation of the Monte Carlo algorithm

4 Answering the Questions

4.1 Time and Ensemble Averaging

To examine the averages, we first plot magnetization and energy versus temperature. This graph represents the time average at each temperature.

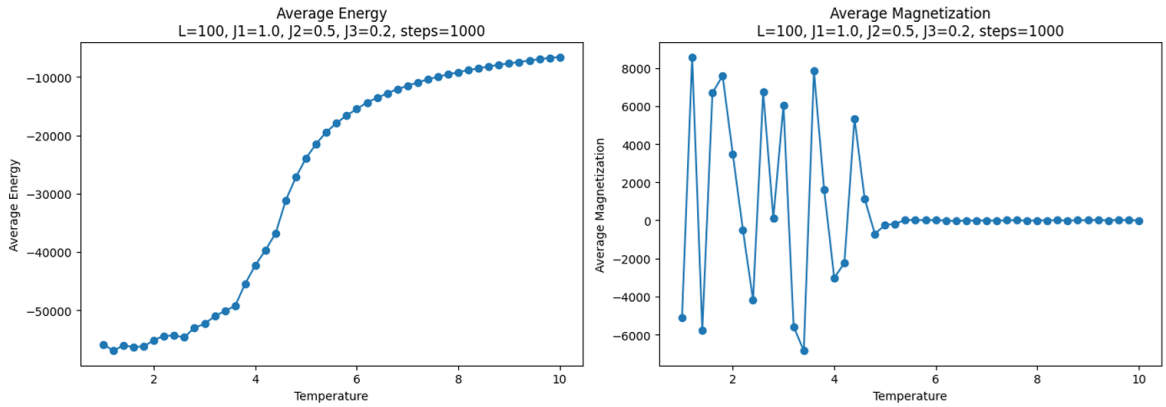


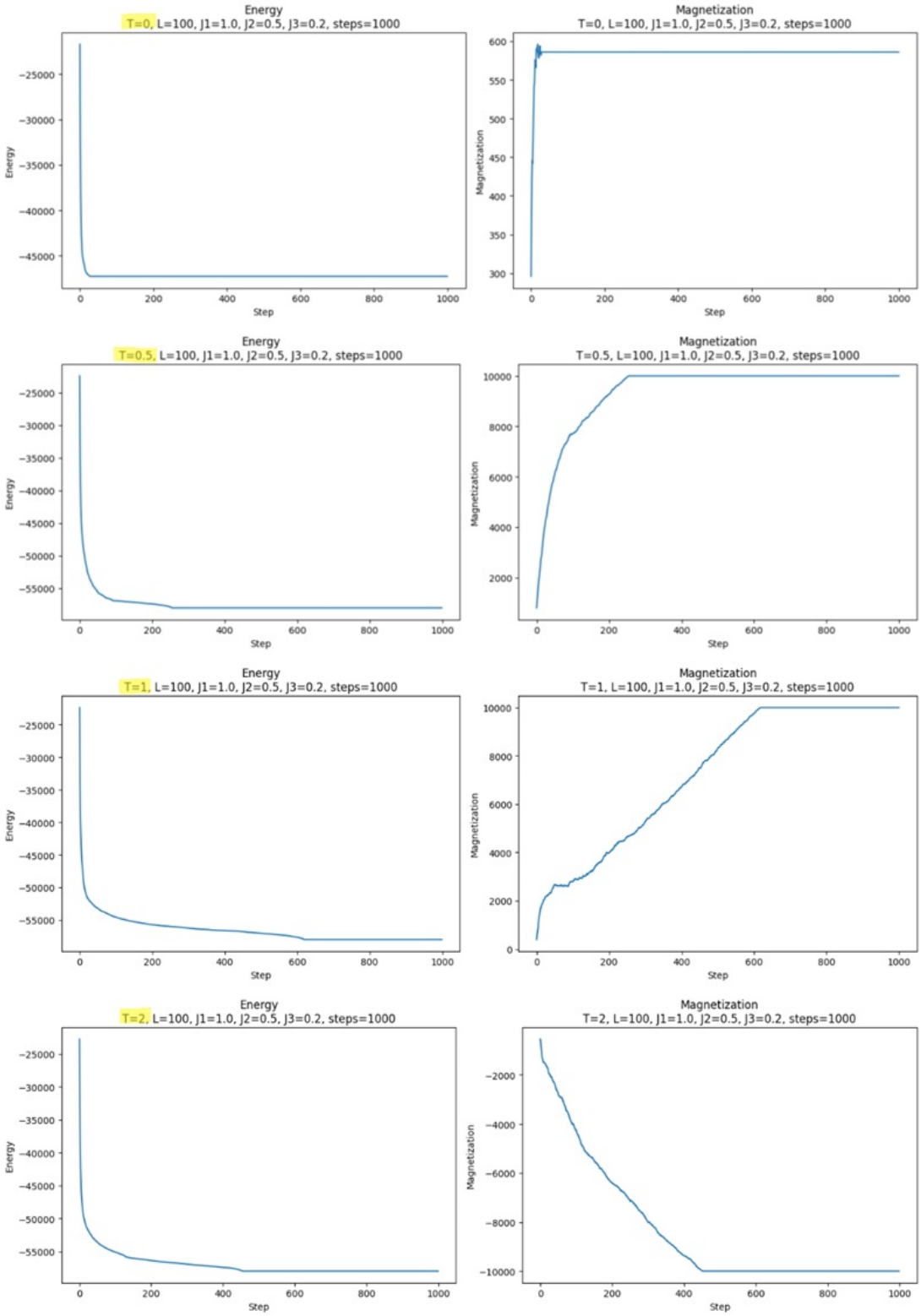
Figure 3: Plot of average energy and average magnetization versus temperature for a 100x100 lattice and 1000 Monte Carlo steps.

Analysis of Figure 3

Average Energy (E) vs. Temperature (T): The average energy of the system is given by $E(T) = \langle H \rangle$. As the temperature increases, thermal fluctuations disrupt the order, and the energy increases. At low temperatures, spins tend to align, minimizing the energy.

Average Magnetization (M) vs. Temperature (T): Magnetization is calculated as $M(T) = \langle \sum_i S_i \rangle$. At low temperatures, the system exhibits ferromagnetic behavior with high magnetization. As the temperature increases beyond the Curie temperature (T_c), the system transitions to a paramagnetic phase, and the magnetization approaches zero. Based on the results, this transition occurs at a temperature between 5 and 6.

For ensemble averaging, we plot the energy and magnetization versus the number of time steps for different temperatures.



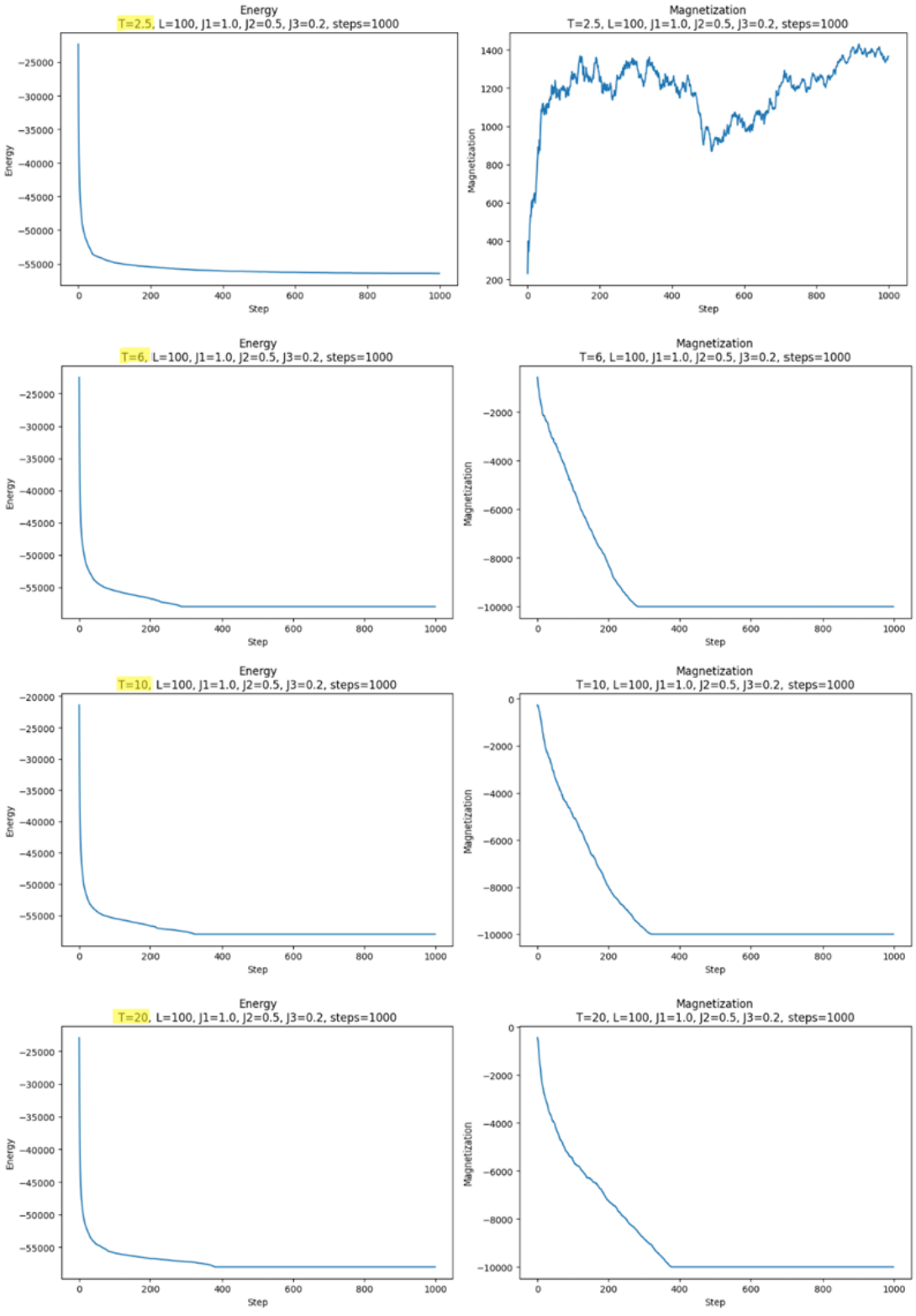


Figure 4: The effect of temperature changes on the convergence of energy and magnetization during the simulation (ensemble averaging).

4.2 Role of System Size

This time, we simulate the system at a constant temperature of $T = 2.5$ for different dimensions.

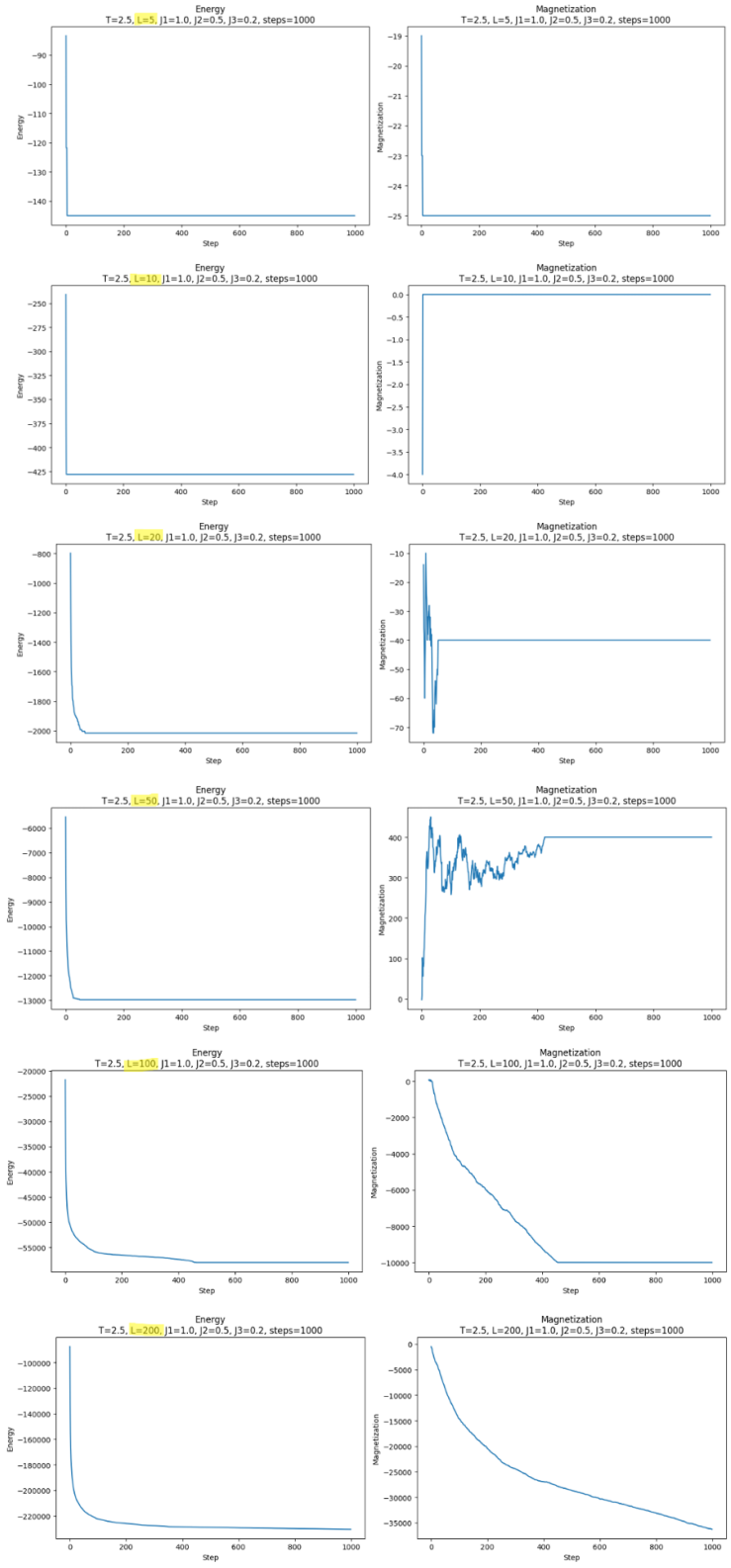


Figure 5: Plot of energy and magnetization versus time for different system sizes.

4.3 Role of Interaction Constants

To investigate the role of the second and third neighbors, we perform the simulation for three different cases:

1. **First neighbor only:** $J_1 = 1.0, J_2 = 0.0, J_3 = 0.0$
2. **First and second neighbors:** $J_1 = 1.0, J_2 = 0.5, J_3 = 0.0$
3. **First, second, and third neighbors:** $J_1 = 1.0, J_2 = 0.5, J_3 = 0.2$

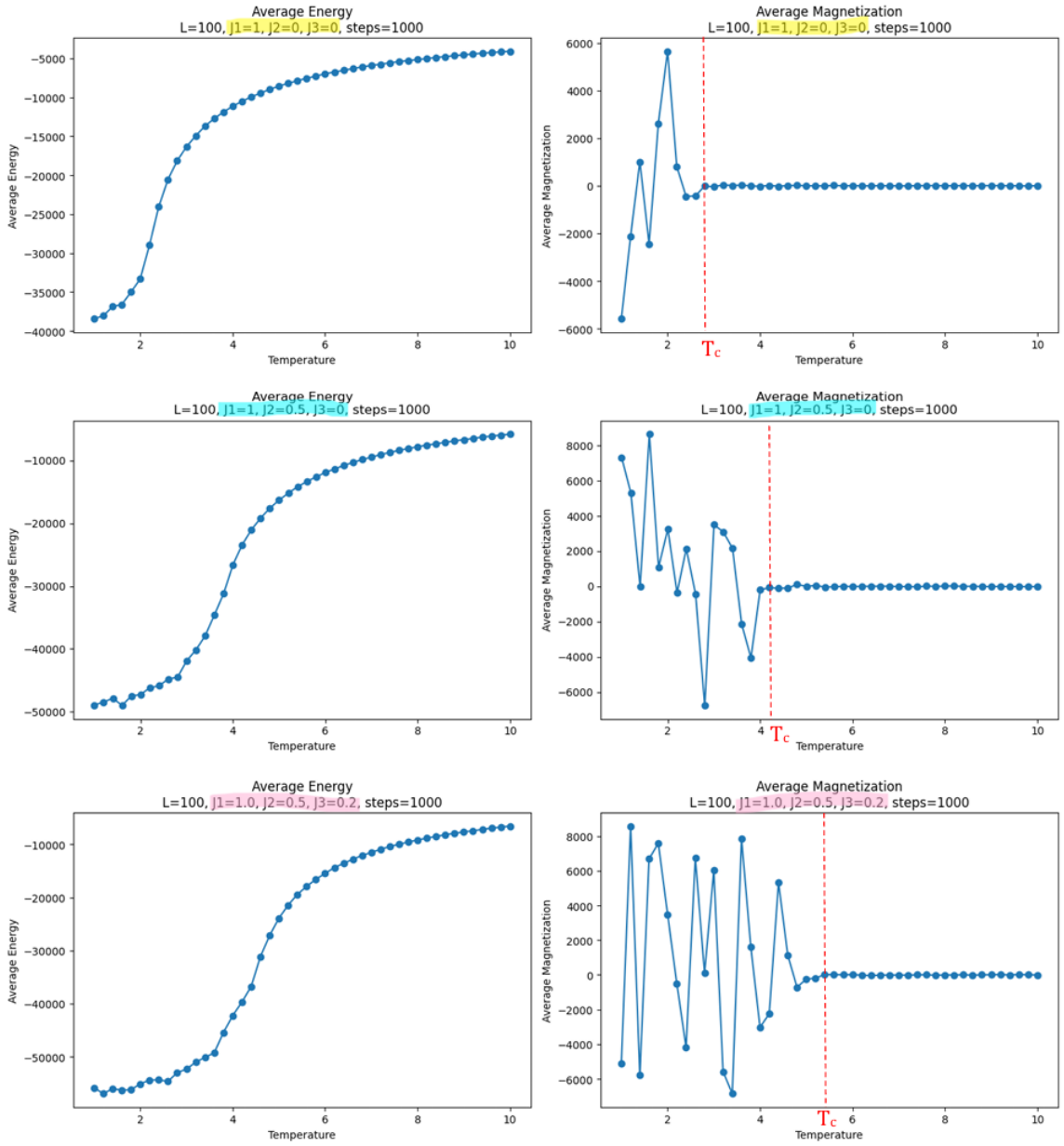


Figure 6: System results for different values of neighbor interaction coefficients (J_i).

Analysis of Figure 6

Average Energy: In all three cases, the average energy increases with temperature (indicating the system's tendency towards lower energy states at low temperatures). Adding the second and third neighbor interaction coefficients causes an overall decrease in the system's energy and changes the shape of the graph.

Average Magnetization: In general, considering higher-order neighbor coefficients (J_2 and J_3) leads to a lower average energy and also more initial fluctuation in the behavior of the average magnetization. Additionally, the Curie temperature (the phase transition point) shifts to higher temperatures.

5 Conclusion

In this project, a 2D Ising model simulation was performed using the Monte Carlo method, considering interactions up to the third nearest neighbor. The results showed that as temperature increases, the system's energy increases and its magnetization decreases, which is consistent with phase transition theory. It was also observed that the interactions of the second and third neighbors contribute to the model's complexity and accuracy, clearly illustrating the behavior of ferromagnetic and paramagnetic phases.